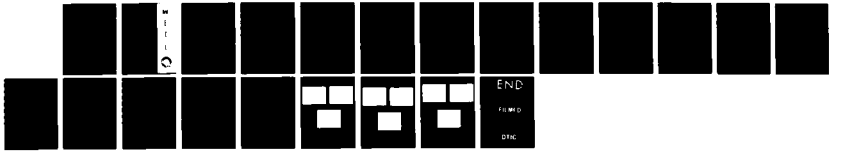
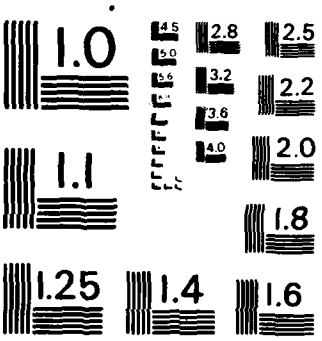
AD-A160 946 ROAD BOUNDARY DETECTION FOR AUTONOMOUS VEHICLE NAVIGATION(U) MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION RESEARCH L S DAVIS ET AL. SEP 85 CAR-TR-140 1/1

UNCLASSIFIED ETL-0407 DACA76-84-C-0004 F/G 17/7 NL





ETL-0407

Road boundary detection for autonomous vehicle navigation

AD-A160 946

Larry S. Davis
Todd Kushner

Center for Automation Research
University of Maryland
College Park, Maryland 20742

September 1985

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

85

E T L

U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
ETL

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0407	2. GOVT ACCESSION NO. AD-A160 994	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ROAD BOUNDARY DETECTION FOR AUTONOMOUS VEHICLE NAVIGATION		5. TYPE OF REPORT & PERIOD COVERED Technical Report July 1984 - July 1985
		6. PERFORMING ORG. REPORT NUMBER CAR-TR-140, CS-TR-1538
7. AUTHOR(s) Larry S. Davis Todd Kushner		8. CONTRACT OR GRANT NUMBER(s) DACA76-84-C-0004
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Automation Research University of Maryland College Park, Maryland 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060-5546		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 18
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image processing
Computer vision
Road following
Autonomous navigation
Sobel edge detector

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Computer Vision Laboratory at the University of Maryland has been developing a computer vision system for autonomous ground navigation of roads and road networks. This report describes the image processing component of that vision system and its implementation on a VICOM image processing system.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document was prepared under contract number DACA76–84–C–0004 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia by the Center for Automation Research, University of Maryland, College Park, Maryland. The Contracting Officer's Representative was Ms. Rosalene M. Holecheck.

A-1

CONTENTS

1. Introduction

The Computer Vision Laboratory at the University of Maryland has been participating in DARPA's Strategic Computing Program for the past year. Specifically, we have been developing a computer vision system for autonomous ground navigation of roads and road networks. Figure 1 contains a block diagram of the system as it is currently configured. The complete system runs on a VAX 11/785, but certain parts of the system have been reimplemented on a VICOM image processing system for experimentation on an autonomous vehicle built for the Martin Marietta Corp., Aerospace Division in Denver, Colorado.

An early description of the system was presented in [1]; our current version is described in [2,3] We give a brief overview here of the principal software components of the system, and then describe the VICOM implementation in detail in the body of this paper.

The Vision Executive (see Figure 1) is responsible for the overall coordination of the system. It represents a centralized source of control that is responsible for scheduling the activities of all of the vision and reasoning processes in the system. It is currently implemented in C on the VAX, but is being redesigned and reimplemented in FLAVORS[4] to run on a SYMBOLICS LISP machine.

The Image Processing module transforms an input image into a symbolic representation of the boundaries of the roads in the field of view. It runs in one of two modes – a bootstrap mode and a feed-forward mode. The bootstrap mode is used to develop an initial representation of the road on which the vehicle is to

travel. Since we assume, at this point, that aside from map information the vehicle has no preconceptions about where the road will be in its field of view or what the detailed structure of that road is (e.g., single lane, with or without shoulders, lane markings, etc.), the bootstrap image processing performs a global analysis of the image to identify significant global linear features. These linear features are grouped into elements called "pencils" (convergent lines in the image plane) which are the units reasoned about by the knowledge base and the geometry module.

During continuous operation, of course, the system has fairly specific expectations concerning the position and appearance of the road. These expectations are generated by the Prediction Module which, based on a three dimensional model of the road constructed by the Geometry and Knowledge based *Modules* and an estimate of the travel between consecutive frames obtained from an inertial navigation system (INS), generates a prediction of where the boundaries of the road will appear near the bottom of the current frame. This prediction is used to constrain the analysis of the image processing operators in the so-called "feed-forward" mode of operation. Here, based on the prediction of where the road boundaries will appear, the Vision Executive identifies small windows in the image that will contain pieces of the left and right road boundary and using a tightly constrained analysis (since both the geometric and photometric properties of large pieces of the road can be carried forward from the analysis of previous frames) identifies the projections of the road boundaries through those windows. Based on the computed locations of the road boundaries through those windows,

subsequent windows are placed and the road is tracked through the image.

It is this set of algorithms that constitute the feed-forward image processing that have been reimplemented on the VICOM image processor. The VICOM implementation allowed us to take advantage of some of the special purpose hardware that the VICOM provides for image processing, but at the same time forced us to seriously consider the various time/space trade-offs that would be solved in one way on a machine such as the VAX (with a moderately fast instruction cycle and built-in floating point operations), but in quite a different way on a Motorola 68000.

Section 2 of this paper contains a brief description of the VICOM image processor as it is configured in our laboratory and at the Martin Marietta test site. In Section 3 we present a detailed description of the feed-forward algorithms as they are implemented on the 68000. Section 4 contains the results of applying those algorithms to several images taken from the test vehicle at the test track in Denver.

2. Vicom Image-Processor Configuration

The University of Maryland and Martin Marietta test site VICOM image processors are standard VICOM VDP configurations. Each can be functionally separated into a standard Motorola 68000 microcomputer system and a special-purpose image processor. The microcomputer contains up to 1.5 megabytes (Mbytes) of main memory, with a combination of 332-Mbyte Winchester-style disk and/or 25- or 16-Mbyte diskette systems. The image processor contains a

three-channel color (RGB) analog Video Input Digitizer (VID), twelve 512-by-512 16-bit-pixel image memories, a three-channel color (RGB) display system, and dedicated image point, ensemble, spatial, and morphological processors. The dedicated processors can perform 12-bit-in/16-bit-out single-point lookup operations, 12-bit ensemble arithmetic (addition, subtraction, multiplication, or logical), image-pair or image/constant combination, 12-bit 3-by-3 pixel spatial convolution, or morphological (binary-image) operations in approximately 0.333 seconds per frame. VICOM memory pixels are directly accessible to the 68000 microprocessor, at a slight cost over the local processor-memory cycle time. The laboratory (non-vehicle) systems are supplied with a VAX host interface and a trackball and mouse.

Software for the VICOM was written using VERSADOS Pascal and Motorola 68000 Assembler, interfacing to the VICOM VDP Image Processor Applications Library and Hardware Driver packages. An Advanced Information and Decision Systems package was used for transferring information between the VAX host and the laboratory VICOM.

3. Extracting Linear Features

The VAX 11/785 implementation of the feed-forward image processing steps are described in [3]. The VICOM implementation exploits special-purpose image processing hardware, involves specific hardware-feature accommodation, and varies from the VAX implementation steps where time/space tradeoffs warrant.

The vehicle camera acquires a 512-by-512, 8-bit/pixel, three-channel color (RGB) image via the VICOM VID. Only a single color band (ordinarily red) is analyzed by the feed-forward algorithm. The input image is converted to VICOM's 16-bit two's-complement pixel format using a high-speed (one-frame-time) pipeline operation. The image is then subsampled from 512-by-512 to 256-by-256 spatial resolution for further processing (as this is a slow microprocessor-performed step, it can be integrated into later steps to save time).

Image edges are then extracted over the entire image using the Sobel edge detector. The x- and y-derivative images are each obtained in one pipeline-convolution operation frame-time. To save processing time, an image whose pixels contain the concatenation of the six most-significant bits of the x- and y-derivative image is produced. This image can be converted to the gradient magnitude and direction images in two pipelined steps using specially-constructed lookup tables. The concatenated image is produced in three pipelined operations by aligning the six most significant bits of the x- and y-gradient images (using two specially-constructed table lookup operations) and one logical **OR** operation to align the images. The precision loss, in this step, is five bits from each derivative image. Finally, the gradient magnitude and direction are each computed using a single table lookup.

The initial windows covering segments of the left and right boundaries of the road are chosen based on projecting the current 3-D road model onto the image plane and determining where the road boundaries enter the image. For these windows we must estimate both the orientation θ and position ρ of the (assumed

locally straight) projection of the road edges. Since θ is constrained somewhat by the prediction, we can ignore any edge points in the window whose directions differ significantly from the predicted value of θ. In addition to "thresholding" the edge points in a window based on gradient direction, we also apply a conservative threshold on the gradient magnitudes. A Hough transform is then computed using the remaining edge points to estimate both θ and ρ. The Hough transform would ordinarily be computed using the following simple algorithm

for each edge point (x, y)

For $\theta = \theta_{min}, \theta_{max}, \Delta\theta$

$p = \lceil X\cos\theta + y\sin\theta \rceil (*)$

$H(p,\theta) = H(p,\theta) + 1$

where H is the Hough transform array. On the VICOM, however, step $(*)$ is very expensive even if the values of the cosine and sine are precomputed and all arithmetic is performed using fixed-point operations. Therefore, we replaced these arithmetic operations by a microprocessor-performed table lookup on the VICOM, assuming a fixed maximum window size (64 by 64) and $\Delta\theta = 3°$. This table is 240K bytes and can reside in either the VICOM program or image memory. The coordinates of the element in H having maximal value determine the projection of the road edge through the window. In subsequent windows, the Hough transform is simplified by constraining the lines in these windows to connect to the lines in the immediately preceding windows (the *road continuity* assumption) – e.g., an endpoint of the line in the previous window becomes the

pivot, or intercept, of the line in the current window. Furthermore, we constrain the orientation of the line in the current window to be in a small interval, $[\theta_m, \theta_M]$, centered about the orientation of the line in the previous window. Since the pivot point is fixed, we need to estimate only one parameter – the direction, θ, of the line through the pivot point. Given the pivot point (x_p, y_p) and any edge point point (x_v, y_v) in the window, the Hough parameter θ is simply

$$\theta = \tan^{-1}((y_v - y_p) / (x_v - x_p)) .$$

The values of θ can be stored in a lookup table. The two lookup parameters Δy and Δx, given by

$$\Delta y = (y_v - y_p)$$
$$\Delta x = (x_v - x_p)$$

are in the range $[-W, +W]$ where W is the maximum length of a window side. For a 64-by-64 maximum window size, this results in a 16K-byte table.

Notice that if the interval $[\theta_m, \theta_M]$ is small, then most points in the square window cannot possibly lie on the line being sought, and it would be wasteful to consider those points at all. We can efficiently enumerate the points in the convex region corresponding to the intersection of the square window and the "infinite" cone centered at (x_p, y_p) and bounded by lines l_1 and l_2 at orientations θ_m and θ_M respectively through (x_p, y_p) using a Discrete Differential Analyzer (DDA) [5] to enumerate the grid points on l_1 and l_2 starting from (x_p, y_p). The DDA algorithm takes a point (x_p, y_p) and an angle θ (rather than two line endpoints as in the standard DDA) and generates points on the line having unit

spacing in either the horizontal or vertical directions. For horizontal spacing, the relationship between the ith and $(i-i)$st points generated are:

$$x_i = x_{i-1} + 1$$

$$y'_i = y'_{i-1} + \tan\theta$$

$$y_i = \left\lceil y'_i \right\rceil$$

For vertical spacing, we have:

$$y_i = y_{i-1} + 1$$

$$x'_i = x'_{i-1} + \cot\theta$$

$$x_i = \left\lceil x'_i \right\rceil$$

The choice of horizontal or vertical spacing is determined by the octant in which the line lies.

Finally, a fixed number of Hough accumulator peak lines are selected, and the peak line closest to the direction of the line in the previous window is chosen as the peak line. To handle slightly curving roads, the line was cut back a fixed fraction above the pivot point (typically, by one-half).

Window placement terminates when a window reaches a fixed image-height fraction (typically, 80%), when a window leaves the image, or when the left and right window sequences cross (e.g., at a horizon vanishing point).

Using 48-by-32 first windows and 32-by-32 subsequent windows on Denver test-track vehicle images, processing typically takes 6-7 seconds of microprocessor and pipelined-image-processor time, including approximately 2.5 seconds for the first two and 1.5 seconds for the subsequent windows' processing time.

4. Results

Figure 2 illustrates the results of applying these algorithms to several images taken from the vehicle at the Denver test track. In Figure 2a we show, for each window, the edge points (thresholded on the basis of both direction and magnitude) that fall within the "cone" generated by the DDA algorithm. Figure 2b shows the superposition of the detected road edges on the images in Figure 2a. Finally, Figure 2c contains the original image, along with the windows and located road boundaries.

5. Summary

This paper has described the image processing component of a computer vision system for autonomous ground navigation of simple roads that identifies road boundaries in small image windows whose positions are predicted from a three-dimensional model of the road and the estimated distance traveled during continuous operation. Based on the computed locations of the road boundaries through those windows, subsequent windows are placed in the image and the road tracked through the image. This set of algorithms has been implemented on the VICOM image processor to take advantage of some special purpose hardware for image processing, but resulted in some trade-offs (heavy reliance on table

lookup operations) due to the nature of the host machine.

Acknowledgements

The authors thank Advanced Information and Decision Systems for its VAX/VICOM interface software, and give particular thanks to John Canning for implementing and enhancing their system on the Computer Vision Laboratory VAX. We would also like to thank Allen M. Waxman and Jacqueline Le Moigne for their advice concerning the conversion of their VAX 11/785 system to the VICOM.

References

[1] A. M. Waxman, J. Le Moigne, and B. Srinivasan, "Visual Navigation of Roadways," *1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985.

[2] A.M. Waxman, J. Le Moigne, L. Davis, E. Liang, and T.K. Siddalingaiah, "A Visual Navigation System for Autonomous Land Vehicles," University of Maryland Center for Automation Research Technical Report 139, July 1985.

[3] J. Le Moigne, A.M. Waxman, B. Srinivasan, and M. Pietikainen, "Image Processing for Visual Navigation of Roadways," University of Maryland Center for Automation Research Technical Report 138, July 1985.

[4] D. Weinreb and D. Moon, *Flavors: Message passing in the LISP machine*, Memo 602, Massachusetts Institute of Technology Artificial Intelligence Laboratory, MIT, November 1980.

[5] J.D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, MA, 1982, pp. 432-436.

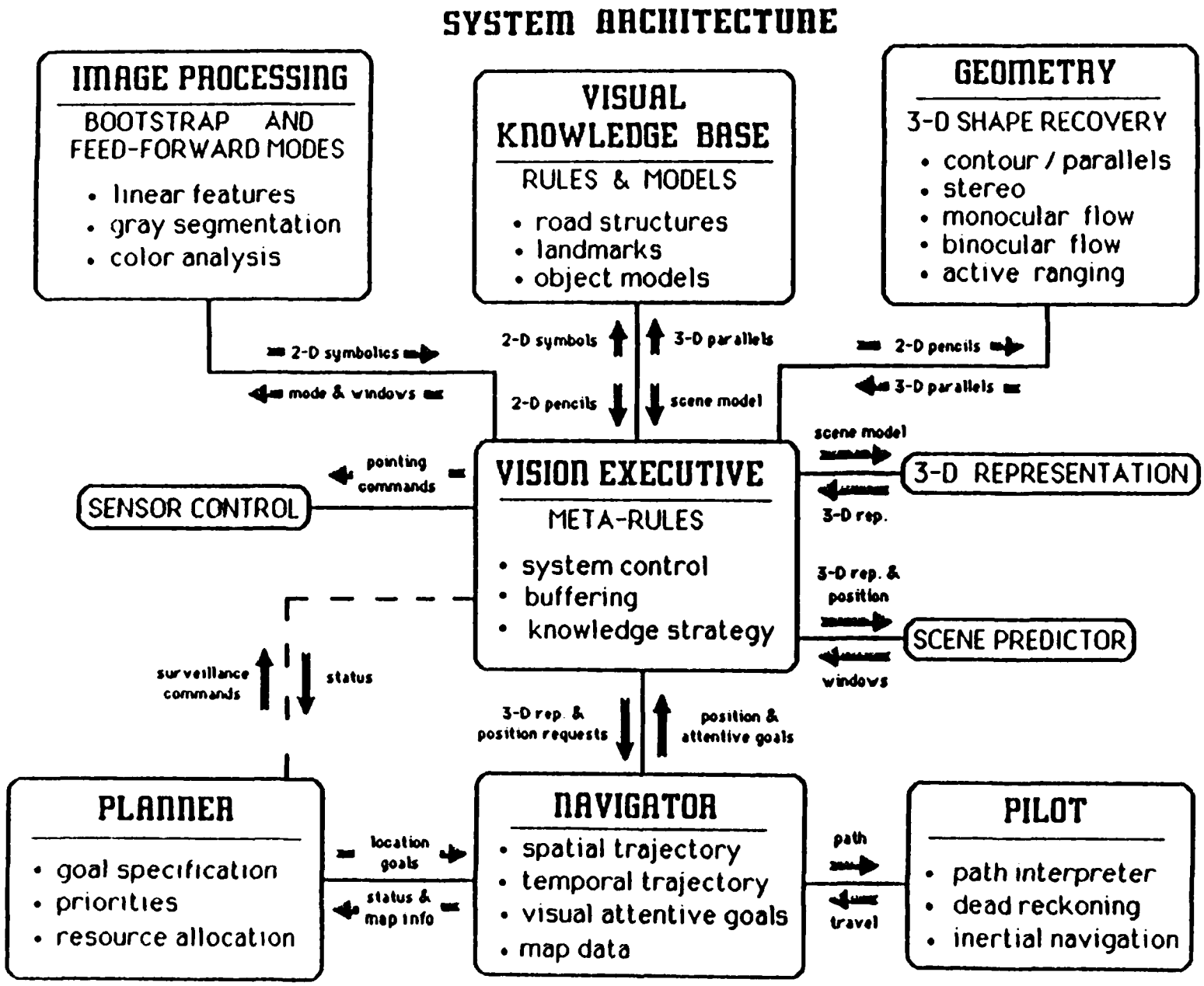
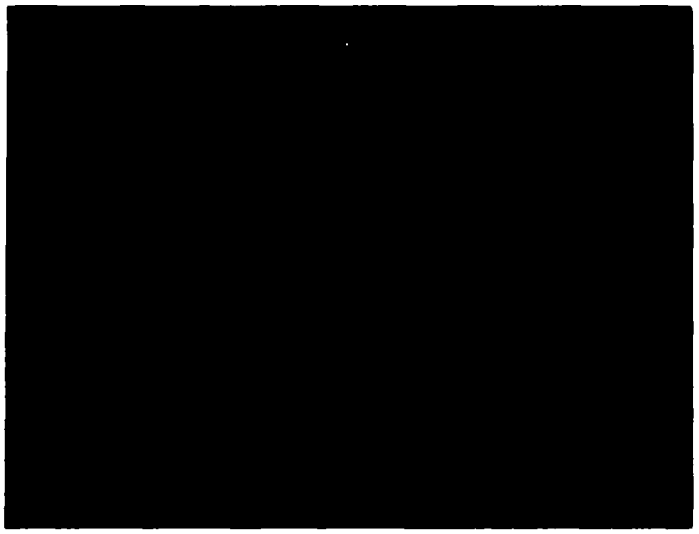
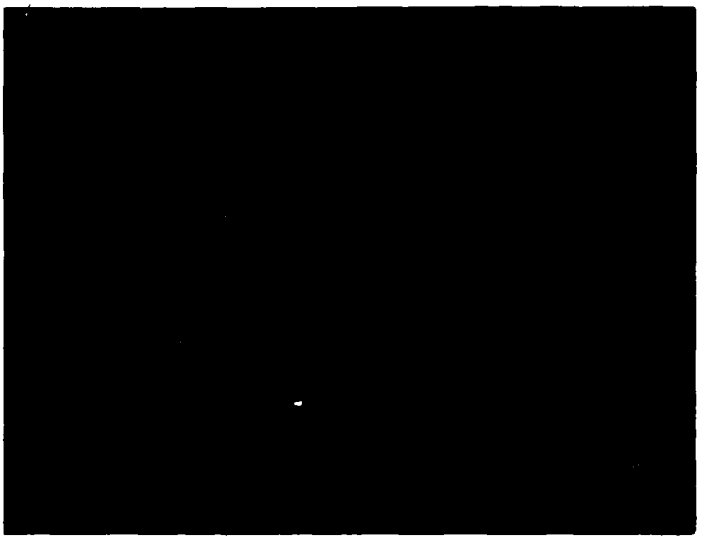


Figure 1. A block diagram of the system





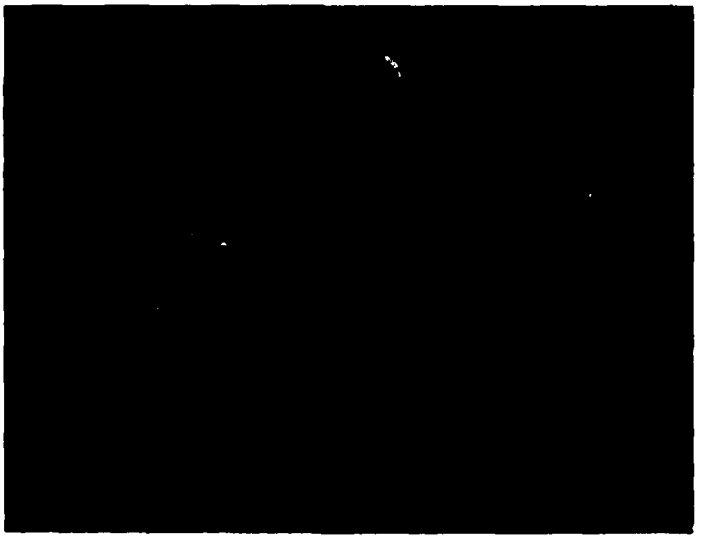


Figure 2. These are the results of applying these algorithms to several images taken from the vehicle at the Denver test track.

Figure 2a. The edge points (thresholded on the basis of both direction and magnitude) that fall within the "cone" generated by the DDA algorithm

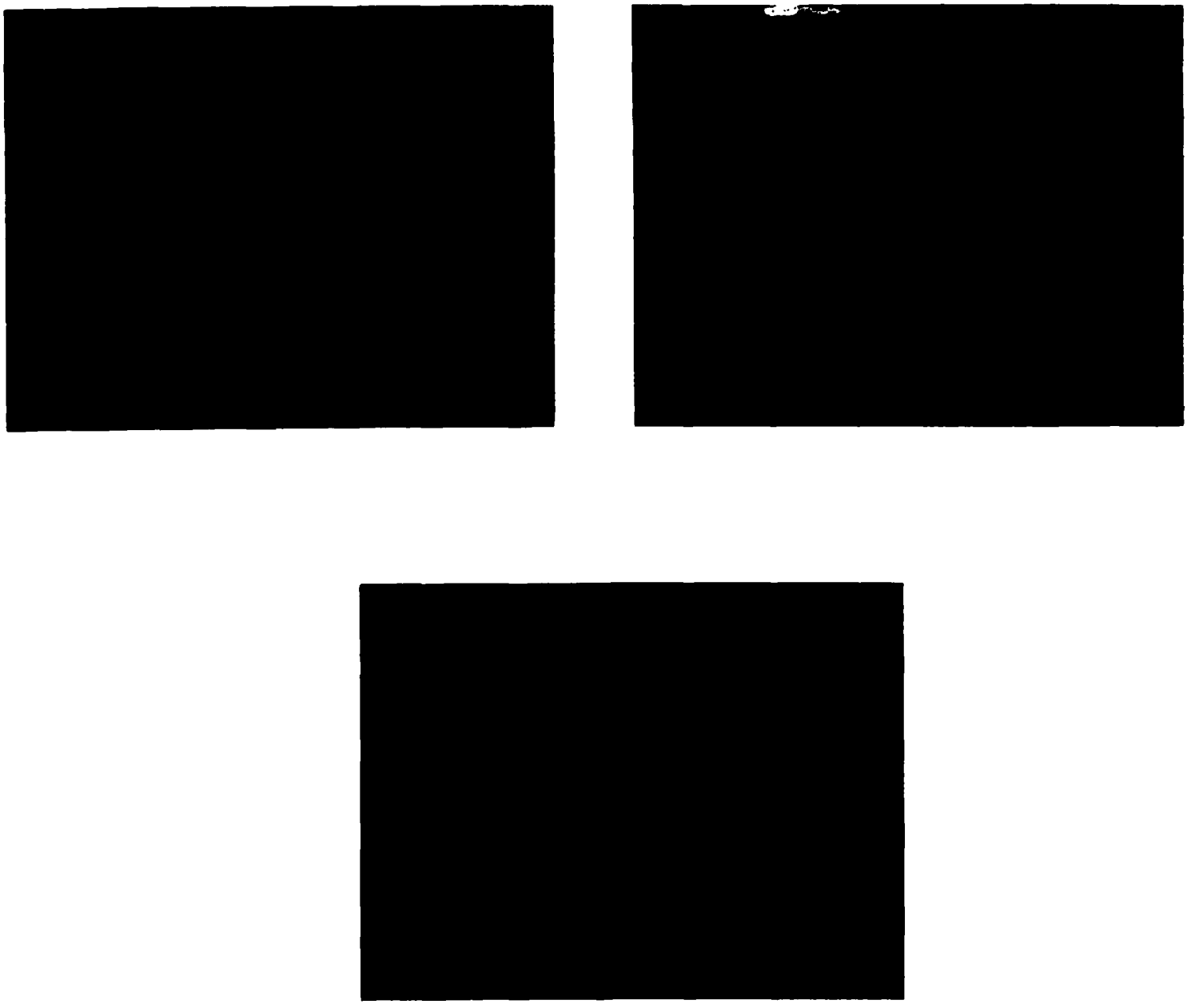


Figure 2b. Superposition of the detected road edges on the images in Figure 2a

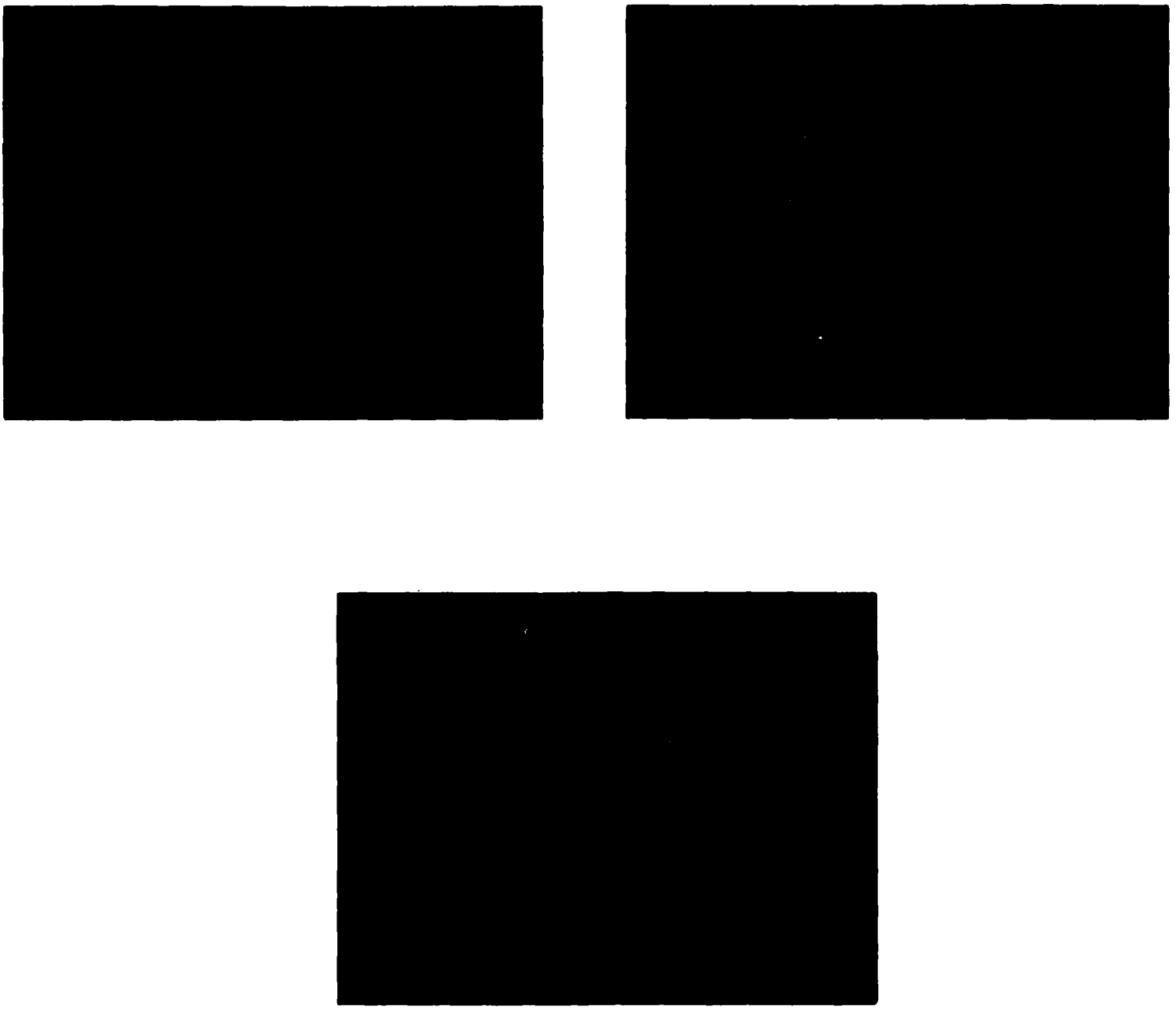


Figure 2c. The original image, along with the windows and located road boundaries

END

FILMED

12-85

DTIC