

AUTOMATIC ANALYSIS OF HIGHWAY FEATURES FROM DIGITAL STEREO-IMAGES

Guangping He, Kurt Novak
Department of Geodetic Science and Surveying, Center for Mapping
The Ohio State University
Commission III

ABSTRACT

A mobile mapping system was developed by the Center for Mapping of the Ohio State University to capture digital stereo-image sequences from a moving van. Each stereo-pair is oriented and located in a global coordinate system by GPS and an inertial system. Therefore, any point which appears in an image-pair can be located in a global coordinate system. Techniques were developed to automatically recognize objects in each image pair. Specifically, we concentrated on finding traffic signs and road edges.

The edges of a highway can be easily extracted, if they are marked by bright lines. In this case an horizontal edge operator is applied, and the bright markers are automatically detected and approximated by straight lines in each image. An intersection transfers the road edges to object space. If a starting point is defined, road edge can be automatically traced until it vanishes. We analyzed milemarkers along railroad tracks and were able to automatically read the log mile number.

Keywords: Digital Systems, Feature Extraction, Image Analysis, 3-D

1. BACKGROUND

The creation of geographic information systems (GIS) requires enormous amounts of digital information. To date most land related data-bases still rely on existing line maps which are manually digitized. In order to collect digital data faster and more accurately the combination of new mapping sensors is necessary. An integrated system can produce digital maps on-line from a moving vehicle, therefore, we talk about "Real Time Mapping".

At the Center for Mapping of the Ohio State University a number of integrated, mobile mapping systems have been designed, assembled, and demonstrated. The most successful system to date is the so-called *GPS-Van* (Bossler, et. al. 1991). Its development was initiated by the US Federal Highway Administration, 38 state transportation agencies, as well as private companies. In principal, it consists of three components: an absolute positioning system, a relative positioning device, and tools for gathering attribute data for the GIS (figure 1).



Figure 1: The GPS-Van integrates a digital stereo-vision system with absolute positioning sensors.

The *absolute positioning sensor* is a combined GPS/inertial surveying unit. Using differential GPS processing the road-alignment can be mapped with an accuracy of 1-3 meters in a global coordinate frame. The inertial system, which consists of two gyros and a wheel counter, takes over when satellite-lock is lost. For *relative positioning* a stereo-vision system was mounted on the GPS-Van. It yields 3-dimensional coordinates relative to the vehicle; they can be transformed into a global system by information collected by the absolute positioning sensors. Finally, an analog video-camera and a touchscreen are applied to *collect attributes*. All data captured by the GPS-Van is immediately stored in a relational data-base that was enhanced by an image management and analysis system.

The GPS-Van's stereo-vision system permits the user to locate any object in the field of view of both cameras in a global coordinate system. The theory behind the transformations involved is discussed in another paper (He, et. al., 1992).

Here we concentrate on the automatic analysis of the digital road images. The extraction and identification of objects in one of the most challenging topics of our research activities. It is absolutely necessary for a practical utilization of the stereo-images, which represent such a large amount of data that a human operator would not be able to extract all information in a reasonable time. The major advantage of detecting features in road images is that we can well predict what kind of objects will appear in the images-pairs, we can even predict their locations.

So far we have been successful in extracting road-edges and center-lines, in matching almost any point in the stereo-pairs, and in reading text and numbers in the digital images. The named tasks are explained in this paper. The following chapter gives a short description of the hardware components used in the GPS-Van. In this conclusions we talk about ongoing developments related to the image-analysis software, and its full integration in a GIS.

2. HARDWARE COMPONENTS

Absolute positioning of the GPS-Van is achieved by two surveying type GPS-recievers (Trimble 4000ST). One serves as a base-station at a known location, the other one is a rover station mounted on the van. When satellite signals are blocked a dead-reckoning system takes over. It consists of a directional and a vertical gyro, and a wheel counter. The directional gyro measures horizontal angular changes (directions), the vertical gyro determines two angles (pitch and roll) that measure the van's tilts relative to the vertical. Additionally, a magnetic proximity sensor counts wheel-rotations at the disk brakes of the two front wheels. Together these sensors generate the absolute positions of the GPS-Van and its orientation (attitude) at any time.

For relative positioning the stereo-vision system was installed. It consists of two fully digital CCD cameras (Cohu 4110) with a resolution of 732 x 484 pixels. They are mounted on a rack on top of the vehicle. We assume tht they are rigidly attached to the van and do not change their attitudes during operations. The two cameras directly interface to a real-time imaging system (Trapix Plus from Recognition Concepts Inc. (RCI)), where the images are temporarily stored in a frame buffer. They can also be processed on-line using a digital signal processor, or they can be sent to the Data Store real-time disk, which has a data transfer rate of 4 MBytes per second and holds 2 GBytes of digital data. It is interfaced to an Exabyte digital tape drive through a SCSI connector.

Finally, a touchscreen is used to control operations of the data-collection procedure and to key in a number of pre-defined attributes as the GPS-Van passes an object of interest. A color-video camera is applied for photo-logging of the road environment; the video scenes are also related to the GPS-positions. All sensors of the mobile mapping system are controlled by a PC. The system configuration is shown in figure 2.

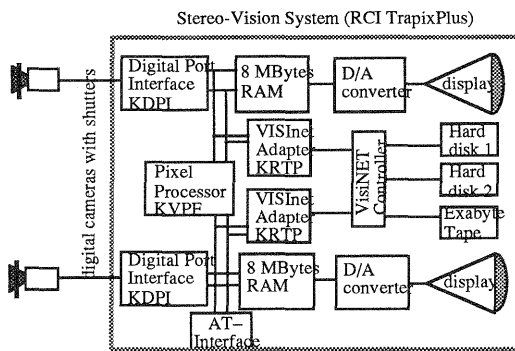


Figure 2: Hardware components of the GPS-Van.

3. EXTRACTION OF ROAD EDGES AND CENTER-LINES

One of the most important features we want to extract from highway image-pairs are edges and centerlines of roads. Two different approaches were developed: the first one relies on the definition of the edge or centerline by a certain pattern, e.g. by a bright line. Well defined patterns can be found fully automatically by the computer; once detected, they are approximated by analytical functions. The second approach permits the user to define one point of an edge on the screen. A line following algorithm traces this edge from the bottom to the top of the image until the edge disappears. In both cases we work in both images of the stereo-pair, which means that the edges can be directly related to a global object coordinate system.

3.1 Automatic Detection of Bright Line Markers

There are many edge detection algorithms available. Most of them require considerable computer-power such as the LoG operator. For our special application we prefer to use a very simple technique, which should work in a horizontal scan line or in a narrow horizontal window of the digital image. In most cases we know where the center-line or the edge of the highway appears in the image-pair. Therefore, we can limit the search area and accelerate the procedure.

Our algorithm is based on the assumption that the edges of the road are marked by solid bright lines and the center-lines or lane dividers are marked by bright dashed lines. We perform the following procedure to directly detect these bright lines in the highway images:

- We calculate the *average and standard deviation of the gray values* (\bar{g}_a, σ_g) of each scan-line,
- A *median filter* is applied to the scan-line to eliminate noise. Typically, a window of five pixels is used, the median gray value is assigned to the center pixel if the difference between the median value and the average is larger than 1.5 times the standard deviation the value is replaced by 1 (white). Otherwise, the pixel is set to 0 (black). This means that we detected a pixel corresponding to a bright line image. This procedure is performed for every pixel of a scan-line.
- If a number of bright pixels appear side by side, their *center point* is used to represent the edge. We also make sure that there are no discontinued line, in the images. This method detects only very bright linear features, which run from top to bottom. The result is a binary image. The extracted edges are represented as a raster and need to be vectorized for further 3-dimensional representation.

3.2 Thinning of Binary Images

Before the detected features can be vectorized their width must be reduced to one pixel. This means that the center pixels of the detected features have to be used to represent the lines. The principle of this procedure is to keep the central point at its correct location while iteratively replacing the boundary of the white (1) areas by 0 values, which corresponds to shrinking the width of the line.

3.3 Line Following

In the next step we convert from raster data (digital images) to vector data (lines). This procedure starts at the corner of the image and looks for the first white pixel (value 1), which represents a pixel of the line. The algorithm tries to follow the line continuously replacing each processed pixel by 0, until the end of the line is reached. Then we procede to the next pixel and apply the same procedure. This method is *chain coding*. It enables us to define a line through the digital image without specifically writing any coordinates, but by expressing dependencies between consecutive pixels.

After following a line its length is computed and checked. All lines which are shorter than a threshold (in our case shorter than 10 pixels) are eliminated. If point i as well as $(i - 1)$ and $(i + 1)$, appear to be on the same linear edge, point i is not registered. Thus we can reduce the amount of data.

3.4 Lane Detection

With this procedure we refer to the approximation of the detected lines by analytical functions. These curves define lanes of the highway. Once the vectorization procedure has been completed a large number of polylines exists in the images. A few of them correspond to the edges of the road, most represent edges of other objects, such as cars, traffic signs, light poles, and trees. To detect the lanes of a highway, which are defined by the center and the edge lines, we have to eliminate all polylines which do not correspond to the real road edges. We implemented a number of rules which form a knowledge base that allows the computer to decide on the most likely candidates for the road edges. The major rules are given below:

- (a) We only want to keep straight lines as road edges. Therefore, we approximate each segment by a straight line (1).

$$x = k * y + b \quad (1)$$

If the standard deviation between the straight line and the approximation is smaller than a threshold, this edge line is kept as a candidate.

- (b) As the edge and center lines of the road are parallel in object space they must appear as intersecting straight lines in the image (a central perspective projection). The intersection point is also called *vanishing point*. It is the location where parallel object lines intersect in infinity. The vanishing point is close to the center of the image; as the cameras of our stereo-vision system are tilted downward by about 8° the vanishing point appears in the upper half of the image. In order to keep a line as a candidate it has to satisfy condition (2), which says that it must pass through an area around the approximate vanishing point.

$$pixel/3 < x = k*y + b < pixels * 2/3$$

$$for\ y = scan/3 \quad (2)$$

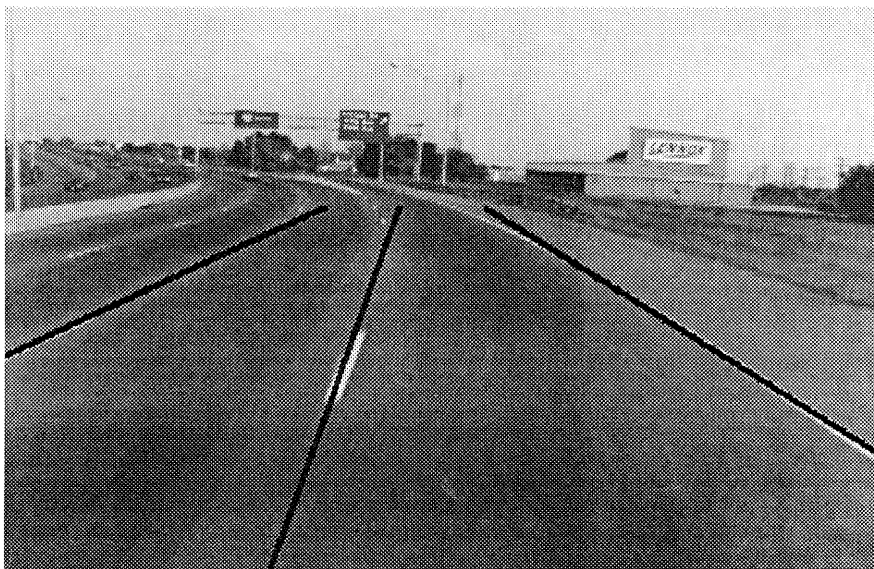
- (c) Next, we collect line segments of similar slope and offset of the origin (k and b) into groups. Additionally, we include information about the lengths of these lines. Segments collected into one group are defined by similar parameters of k and b. Each group of lines is defined by only one straight line, which is the best approximation of all segments contained in this group. As the edges and center lines of the road are much longer than all the other lines, we can simply distinguish by keeping only significant groups as edge candidates. A criterion for the selection is the number of individual points which is represented by a specific straight line.
- (d) Finally, we have to approximate the edges and center-lines of the road in each of the selected groups by analytical function. A straight line is defined through the pixels of each group. This straight line really represents the edge of the road in a certain distance in front of the van. Of course, if the road is bending these approximations will no longer be true. However, in a more sophisticated procedure the straight lines can be approximated or replaced by analytical curves.

3.5 Intersection of Linear Features

The procedure discussed in the previous chapter results in a number of straight lines in each of the stereo-images. As we are mainly interested in the object coordinates of these features they must be projected into object space. For this purpose we have to find two corresponding points for each line in the two images.

For any point selected in the left image we can find the conjugate point in the right image by intersecting the epipolar line with the road edge. If we do this for at least two points we can easily compute the object coordinates by the collinearity equations. These two points are connected by a straight line which represents the road edge in local coordinates relative to the GPS-Van. By applying information about the position of the van and its orientation the global coordinates of these edges can be computed.

The procedure described here can be performed for each stereo-pair of a sequence. The parallel lines which are created for each image-pair can be plotted in the same world coordinate system and be displayed as a road map (figure 3).



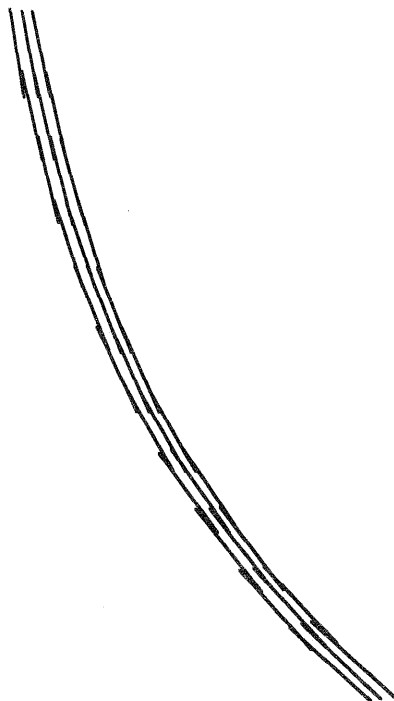


Figure 3: The road edges of sequential stereo-pairs were automatically extracted by the described algorithm.

4. FOLLOWING OF ROAD EDGES

This procedure is used to trace almost vertical lines in digital images. We assume that a starting point is available; it can be found by manual selection on the computer monitor, or by edge extraction as described previously. The objective of this procedure is to follow a line, such as the edge of the road, in an image-pair from the bottom to the top or along a sequence of images at the same horizontal line. In our extended version we can get 3-dimensional coordinates of the edge points in object space. This algorithm could be directly implemented in the GPS-Van to automatically trace the edges through the image sequence while the vehicle is moving.

As the lines to be followed in the road images are almost vertical, we only need to check the horizontal off-set of the feature when we move from one scan-line to the next. For this purpose the least squares image matching technique was modified. We have to solve for only one horizontal shift parameter per line. Equation (3) gives the gray-value equation which can be written for each pixel of the matching window. This window should be rectangular with a larger horizontal extension; we could limit this rectangle to only one line of pixels.

$$f(x, y_i) = g(x^o, y_k) + g_x(x^o, y_k) dx \quad (3)$$

$f(x, y_i)$ template window at line i ,
 $g(x^o, y_k)$ search window at an approximate horizontal location (x^o) at line k ,
 $g_x(x^o, y_k)$ gray-value derivative in the x -direction,
 dx unknown shift parameter.

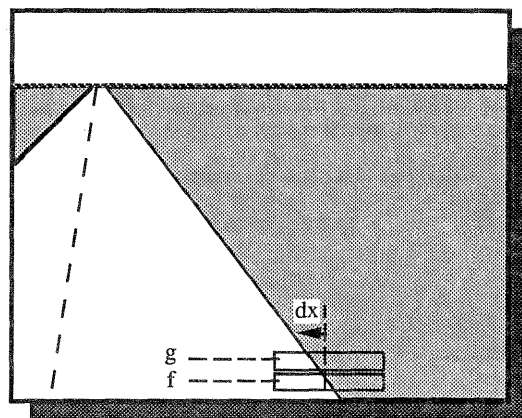


Figure 4: Following the edge of a road in one image.

Figure 4 shows the basic procedure of tracking a road edge starting at the lowest line of the image. First, we define a template at the position of the known edge (x, y_1) (f - window) and match it to another window (g), which is a few pixels above the original (scan-line k). As a result, we get the offset (dx) of the center of the template in the search area. This gives us our next template position. Basically, the displayed pair of windows (f and g) is just shifted towards the top by a few lines. At this location the matching is repeated. As we solve for only one parameter (dx) the solution of the normal equations is done by a simple division. If the correlation coefficient between template and matching window is too small after modeling, line following is terminated this means that the two windows show different gray-value patterns, or if the computed precision of the horizontal shift parameter is too large, which indicates an unreliable match. This happens at the upper half of the image, where the contrast of the road edge is fading.

Edge following can be extended to the stereo-pair as well. As our road images are very close to normal case, we basically have to match the same lines of the two images and directly compute an intersection to get the position on the ground.

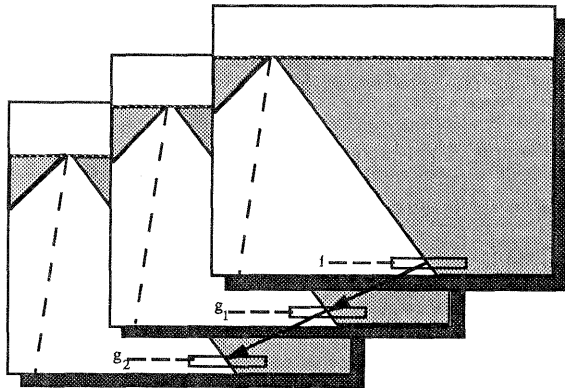


Figure 5: Tracking edges through a sequence of images.

A variation of this procedure is to track edges through a sequence of images (figure 5). In this case, we select a horizontal line in the first image and compare it to the same position in the next image. This means that the y-coordinates (horizontal scan lines) of the two points are identical. One can assume that the x-coordinate does not change very much, especially, if the GPS-Van is moving on relatively straight highways and does not change lanes.

A merit of the least squares procedure is the capability of adding constraints to make it more robust. In our case, we can predict the position of the edge by the knowledge of previously matched points. We assume that all edge points are on a smooth line, which can be represented by a polynomial function in object space. This helps to bridge over areas of low contrast or other disturbances, such as shadows or different layers of asphalt on the road surfaces. These constraints could be defined in image space or in object space. Figure 6 shows the results of tracing the road edges our image sequence. One can see that the line exactly corresponds to the marked edge-line on the asphalt, and that the procedure stops when the road begins to bend.

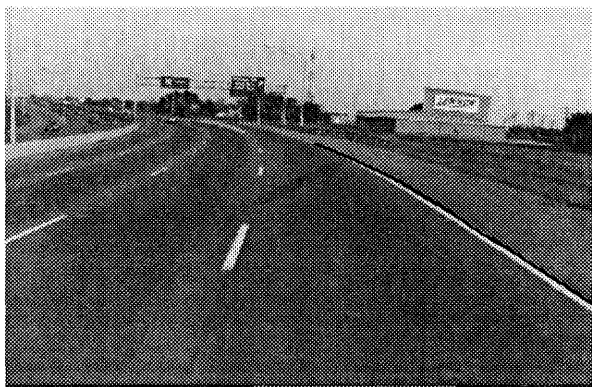


Figure 6: Road edges detected by line following in the digital images.

5. DETECTION OF MILEMARKERS IN DIGITAL IMAGES

The algorithms described in this chapter were specifically developed for detecting vertical, linear bright objects in digital imagery. They evolved from techniques originally implemented to detect road edges (see chapter 3). Their major application is the automatic recognition and identification of milemarkers from image-pairs of railroad tracks. This procedure consists of three steps: the detection of mile-marker candidates by a horizontal edge-detector, the tracing of the vertical edges of the mile-marker candidates to identify their outlines, and the extraction of the log-mile number.

5.1 Finding Mile-Marker Candidates

We assume that the image is displayed upright on the computer screen. For each line of the digital image a one-dimensional (horizontal) edge extraction is performed, which detects close to vertical contrasts of dark and bright pixels. The mile-marker typically is brighter than the background; as this is only true in the lower part of the images, where the background would be grass or soil, the bottom of the mile-markers can be detected in a reliable way. In the upper part of the images the background is mostly sky, which is sometimes brighter than the mile-marker itself. Here the second step of the procedure takes over.

We use a gradient based operator, which computes the averages and standard deviations of short horizontal windows. If the standard deviation becomes larger and the average varies significantly, the image contrast changes from dark to bright, or the other way around. This is the type of edge we want to detect (bright object, dark background). With this technique we can segment each scan-line in background (dark) and object (bright) areas.

As a result we obtain white patches in the image, which are the mile-marker candidates, however, many of them are of completely different form than the real monument. Now the small patches (defined by the number of pixels) are eliminated, and only those are kept that have a long vertical and short horizontal dimension. These candidates are used as approximations for the next step.

5.2 Edge Following

Beginning at the edges detected in 5.1 a Sobel operator is applied to a small area around the approximate edge in the gray scale image. By this function the edges are located more accurately. Two edge points are found in each scan-line. The distance between these edges determines the width of the mile-marker in the digital image. The left and right gray values at each edge are checked; inner and outer gray values must be consistent, which means that a gray value inside the object must be brighter than the one on the outside of the detected edge (figure 7). Starting at this line, the edges are traced upwards, always checking the width of the mile-marker and its color (average gray-value), in order to obtain a consistent edge up to the top. This edge tracing technique yields very reliable results, even at the top of the mile-markers, where the contrast against the sky is usually weak.

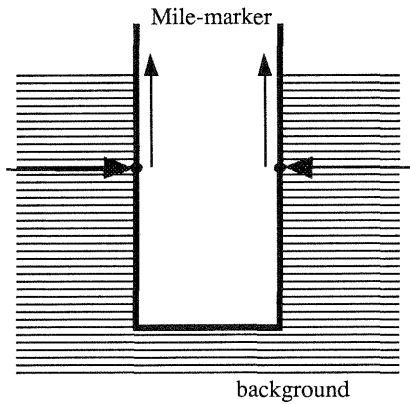


Figure 7: Tracing the edges of the mile-marker from the bottom to the top.

5.3 Extraction of the Log-Mile Number

While the edges are traced from the bottom of the mile-marker to the top, the average and the standard deviation of the pixels between the detected edge points are computed. As we reach the number at the top of the monument, the gray-value average decreases and the standard deviation gets larger for a number of lines (figure 8). These lines of lower gray values, but higher standard deviation define the area in which the log-mile number is written; they are chosen to identify the position of this number. It can be read by the OCR techniques described below.

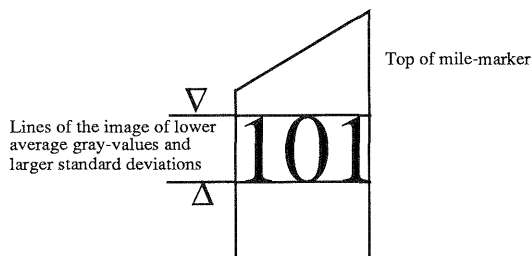


Figure 8: Locating the log-mile number at the top of a mile-marker by computing average and standard deviation of the gray-values between the edges.

It is based on a simple algorithm which allows to identify characters from vector graphics. For this technique the numbers must appear upright in the images, which is the case in our mile-marker photos. A prerequisite is the vectorization of the number, e.g. by thresholding the image and thinning the black (binarized) digits to a width of only one pixel. By chain-coding a polyline is created which forms the detected number. It is smoothed, in order to eliminate noise from binarization.

Each polyline (= number) is intersected by some horizontal and vertical lines; the number of intersections in x and y directions is counted (figure 9). Each digit is uniquely identified by the number of intersections, which are stored for comparison purposes in the program. The major advantage of this technique is its computational speed.

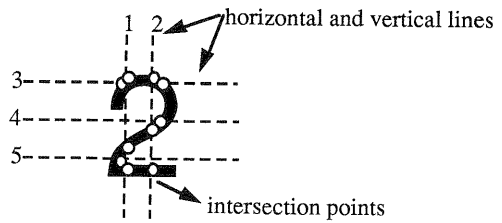


Figure 9: A character can be identified in a unique way by counting intersections of horizontal and vertical lines, e.g. number of intersections per line:

line 1 : 3
line 2 : 3
line 3 : 2
line 4 : 1
line 5 : 1

6. CONCLUSIONS

In this article we tried to show the big potential of using digital image-pairs for automatic mapping of features from land-based vehicles. Although we are currently limiting ourselves to a few well-defined objects, this system could be applied to create a complete highway or railroad inventory, containing traffic signs, mile markers, and road signals. Due to the known position and attitude of the vehicle, any object identified in an image-pair is immediately available in a global coordinate system.

As both the locations of image-pairs, and the detected objects are given in geographic coordinates, it makes sense to integrate these functions in a GIS. We are currently working on this problem, combining an existing vector GIS with the stereo-image data-base. The feature extraction functions serve as tools for collecting attribute information from the imagery. In order to allow the operator to better check the results a stereo-monitor will be used to display image-pairs and to select objects directly in 3-dimensions.

Once a set of image analysis functions is available we envision to install image processing capabilities directly in the vehicle. Together with real-time, differential GPS we will be able to extract road features while driving along highways. Then we can truly talk about real-time mapping.

7. REFERENCES

- Bossler J., Goad C., Johnson P., Novak K., 1991. "GPS and GIS Map the Nation's Highways." *GeoInfo Systems Magazine*, March issue, pp. 26-37.
- Goad C., 1991. "The Ohio State University Highway Mapping System: The Positioning Component." *Proceedings of the Institute of Navigation Conference*, Williamsburg, VA, pp. 117-120.
- He G., Novak, K., 1992. "Automatic Analysis of Highway Features from Digital Stereo-Images." *International Archives of Photogrammetry and Remote Sensing*, Vol. , Commission III.
- Novak K., 1991. "The Ohio State University Highway Mapping System: The Stereo Vision System Component." *Proceedings of the Institute of Navigation Conference*, Williamsburg, VA, pp. 121-124.