Image-Based Geo-Specific Road Database Creation for Driving Simulation

Dahai Guo, Arthur R. Weeks, Harold I. Klee

Department of Electrical and Computer Engineering

University of Central Florida

4000 University Boulevard, Orlando, FL, 32828

Tel: 407-823-5810, Email: dgu@bruce.engr.ucf.edu

Abstract

Geo-specific road databases are sometimes necessary for driving simulation studies. However, manually creating them is very time consuming. One of the reasons is that a real road can have non-uniform cross sections due to irregular shaped road boundaries. Additionally, in the UCF driving simulator system, the road database format requires identification of repeatable cross sections, which is also a labor intensive process. While some commercial software applications, such as the MultiGen Road Tool have road modeling functions, roads with non-uniform cross-sectional profiles are still hard to model.

In this research, an image understanding based method is proposed for geo-specific road database development. Digital Line Graph (DLG) data, issued by the United States Geographical Survey (USGS) is used to limit the search space for road segmentation. The mean-shift clustering method is chosen to be the solution for separating pavement and non-pavement areas within road areas. The Linde-Buzo-Gray (LBG) vector quantization method is used to identify repeatable cross sections. The proposed method results in an average accuracy of road segmentation above 85%. The average error of cross-sectional profile is no more than 0.36 feet. The proposed system can significantly reduce the efforts in creating geo-specific road databases. The utilization of the two types of geographical information, high-resolution aerial photos and USGS DLG data, play the most important role in the proposed system.

MATCHING USGS DLG DATA WITH HIGH RESOLUTION AERIAL PHOTOS

Since the aerial images in this research and the digital road information (DLG from USGS) [1] are from different sources, they need to be matched accurately before the recognition process begins. Two types of distortions between an image and the USGS data are considered in this research. One is linear and the other is non-linear. Figure 1 shows the two kinds of distortions. The methods for identifying these two types of distortion are explained in [2].

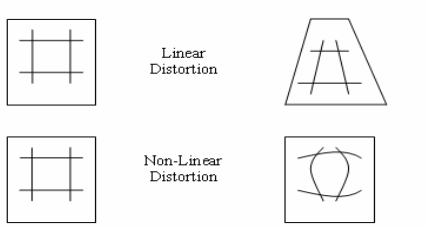

FIGURE 1 Linear and Non-Linear Distortions

The most significant linear distortions discovered were 2D translation and scaling. No significant non-linear distortion was found. Matching USGS DLG data and high resolution aerial photos was relatively straightforward and simple to implement. Figure 2 shows one of the results.

FIGURE 2 Matching of the USGS DLG and UCF Aerial Photos

The DLGs, represented by the red lines in Figure 2, dramatically limited the search space for road boundaries. The unwanted image contents, such as ponds and forests, were removed from consideration.

ROAD SEGMENTATION

THE MEAN-SHIFT CLUSTERING METHOD

There are many image segmentation methods. Each method uses cues to segment images. Those cues can be RGB (Red, Green, Blue), or HSI (Hue, Saturation, Intensity). No matter which cues are selected, a method that groups similar pixels is needed. The mean-

shift clustering method [3] is adopted as the grouping method in this research. It is a nonparametric method for searching the real estimation within a search window with radius r. The value of radius r decides the resolution of segmentation. Larger values of r result in lower segmentation resolution. In this paper, r is chosen to be the square root of the trace of the global covariance matrix in the HSI fields. The segmentation algorithm is defined as follows:

1. Given an aerial image, convert the RGB image to an HSI image. The process of converting RDB to HIS is explained in [4].
2. A 3D histogram of the HSI image is obtained.
3. m points in the HSI image are chosen randomly.
4. The point which corresponds to the largest value in the HSI histogram is identified.
5. Starting from the point, $\bar{x} = \{h, s, i\}$, in the HIS histogram, find the shift vector using the formula

$$\Delta x = \frac{r^2}{5} \frac{\nabla p(\bar{x})}{p(\bar{x})} \tag{1}$$

   where $p(\bar{x})$ is the value in the HSI histogram at $\bar{x}$.
6. Update $\bar{x} = \bar{x} + \Delta\bar{x}$. If x converges, go to step 7, otherwise go to 5.
7. Centering at $\bar{x}$, all points in the window with radius r in the HSI histogram will be treated as a class and taken off the image.
8. Repeat steps 3, 4, 5 and 6 until all the points in the HSI image have been classified.
9. After step 7, several classes are formed, each of which is an estimation vector $\bar{x}$ of hue, saturation and intensity. Each class also has a number of points that are classified to it. From the formed classes, drop the classes whose number of pixels is less than 5% of the total number of pixels. For each point in the dropped classes, reclassify it to one of the remaining classes which has the shortest Euclidean distance to this point.

In step 5, the formula calculates the shift vector $\Delta\bar{x}$ which points to the real estimation within the window which has radius r and centers at $\bar{x}$. When the shift vector $\Delta\bar{x}$ is calculated, $\bar{x}$ is updated. This process repeats until $\bar{x}$ converges. The resulting $\bar{x}$ is the real estimation within the window, and all the points within the result window that centers at $\bar{x}$ will be treated as one class. The classified point's corresponding pixels in the image will be removed from the image. This process is repeated until all the points in the image are classified.

POST-PROCESSING THE RESULTS BY THE MEAN-SHIFT CLUSTERING METHOD

Pavements in aerial photos generally have lower saturation, compared with other areas. In other words, the areas with a high average saturation are more likely to be non-pavement. It is possible to use saturation values to segment aerial images. The difficulty is that a saturation image can easily become noisy. Sometimes a saturation image is so noisy that the road boundary is not recognizable. In this case, the classification results via the mean-shift algorithm can be used. The mean-shift algorithm classifies an image into several classes. The average saturation value of each class separates pavements and non-pavements. Mathematically, the recognition process is as follows:

1. Sort the classes based on the average saturation in a descending order;
2. For class i, where i is from 1 to n, where n is the number of the classes, classify it to be non-pavement if $i \le t$, otherwise classify it to be pavement. Index t must satisfy $\frac{\bar{s}_t - \bar{s}_{t+1}}{\bar{s}_t} > 0.4$, where s stands for the average saturation with in a class.

The threshold value of 0.4 is used to detect the first significant drop in the average saturations.

IDENTIFYING REPEATABLE CROSS SECTIONS

STS ROAD DATABASE FORMAT

The UCF driving Simulator is manufactured by Simulation Technology Solution (STS), which has a unique way of describing road data. Instead of using polygonal road descriptions, a combination of road center line points and cross sections is adopted to describe roads in an STS driving simulator system because the cross section description could be reused at different points along the road. This scheme, similar to the method used in the MultiGen Creator Road Tool [5], is particularly efficient when roads are primarily composed of uniform cross sections.

Each uniform section can be defined by a cross section profile and a set of center line points. Curve road boundaries are modeled in the STS RDB system using distinct center line points and corresponding cross sections. Identical cross section profiles can occur many times in a road network. Figure 3 illustrates how the STS's RDB system handles this situation in an efficient manner by employing a pool of cross sections. Each road segment is spatially sampled along its center line. For each center line point, an entry in the cross section pool will be used to define the cross sectional view at this point. Entries in the pool could be used more than one time as in the case of the cross section designated as Cross Section Number 1 (see Figure 3).

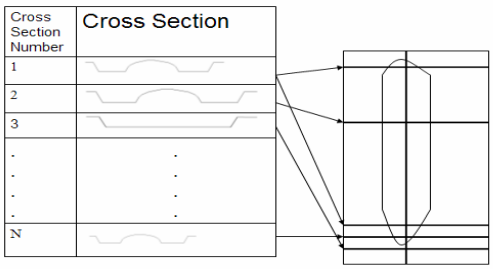

FIGURE. 3 STS Road Databases Requirements

The process of forming a pool of cross sections is actually encoding them. In other words, similar cross sections should be identified and appear in road databases as a distinct integer.

LBG ALGORITHM

In order to form a pool of cross sections, center lines are sampled. The sampling rate can vary. For each sampled center line point, a line is drawn perpendicular to its center line. This line will intersect with boundaries of roads and medians, and comprise the top view of the cross section at this point. The true cross-sectional profile can be derived from the top view using basic geometric design parameters for existing roads and medians as shown in Figure 4.

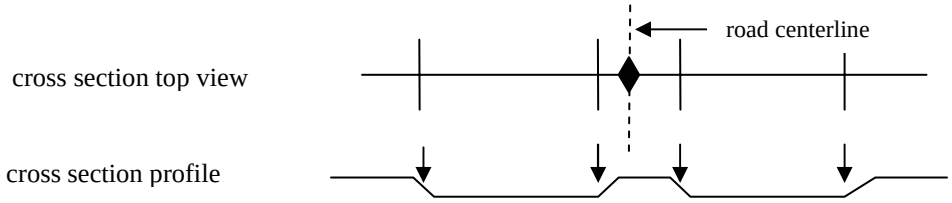

FIGURE. 4 Transforming Cross Sections Top Views in to True Profiles

In Figure 4, the cross section top view line intersects with four vertical lines. The middle two vertical lines are median boundaries and the other two vertical lines are road boundaries. The four intersection points uniquely define a top view of the cross section associated with the diamond shape road centerline point. Assuming the height of curbs conforms to a six inch standard enables construction of the true cross section profile at this point.

Top views of cross sections are available at each sampled center line point with the possibility that many are very similar, if not identical. The question is how to identify similar cross sections. In this research, the Linde-Buzo-Gray (LBG) vector quantization algorithm [6] is used to identify similar top views automatically.

For example, in Figure 4 the distances from the center line point to the four intersection points on the cross section top view are -48.0, -6.0, 7.5, and, 56.7 feet. A negative value indicates that the intersection point is located on the left side of the center line point. Consequently, the cross section vector of this center line point is (-48.0, -6.0, 7.5, 56.7). The input to the LBG algorithm includes a set of cross section vectors.

The LBG algorithm automates the process of forming the pool of cross sections. It encodes cross section databases, identifies and removes redundant cross sections. The LBG algorithm can quantitatively evaluate the encoding error. Below is the description of the LBG algorithm training process.

1. Start with an initial set of reconstruction values $\{Y_i^{(0)}\}_{i=1}^{N}$ and a set of training vectors $\{X_i^{(0)}\}_{i=1}^{M}$. Set $k = 0$, $D^{(0)} = 0$. Select threshold $\in$.
2. The quantization regions $\{V_i^{(k)}\}_{i=1}^{M}$ are given by $V_i^{(k)} = \{X_n : d(X_n, Y_i) < d(X_n, Y_j) \forall j \neq i\}, i = 1,2,....,N$, where $d(X_n, Y_j)$ is the Euclidean distance between vectors X_n and Y_j
3. Compute the average distortion $D^{(k)}$ between the training vectors and the representative reconstruction value.

4. If $\frac{D^{(k)} - D^{(k-1)}}{D^{(k)}} < \in$ stop; otherwise continue.
5. $k = k + 1$. Find new reconstruction values $\{Y_i^{(k)}\}_{i=1}^{N}$ that are the average value of the elements of each of the quantization regions $V_i^{(k-1)}$. Go to Step 2.

The input to the LBG algorithm is a set of M cross section vectors. The desired number of cross sections is N and $\{Y_i^{(k)}\}_{i=1}^{N}$ will be the cross sections in the cross section pool when training stops. $\{X_i^{(0)}\}_{i=1}^{M}$ are the M cross sections before quantization and each one will point to the closest member of $\{Y_i^{(k)}\}_{i=1}^{N}$ after quantization.

Initially there is no prior knowledge of how many cross sections are needed for describing the original cross section database. An acceptable approach is to start with a small value of N, e.g. 2 and then calculate the average error among all the cross sections. If this error is not acceptable, the intermediate N vectors will be perturbed to generate another group of N vectors for retraining. The perturbing method adopted in this research is to shift each original entry by a small vector.

The relationship between average errors and numbers of vectors in the codebook is shown in Table 1. The LBG algorithm should work towards two pools of cross sections corresponding to roads with and without medians. The top views of road cross sections with medians will have four points, while those without medians will have only two points. Therefore, the LBG algorithm is identifying similar 2D vectors and 4D vectors. The data in Table 1 shows that the average error decreases when the pool size increases.

Two-Point Cross Section		Four-Point Cross Section	
Pool Size	Average Error (ft)	Pool Size	Average Error (ft)
16	2.03	16	1.88
32	1.38	32	1.39
64	0.82	64	1.00
128	0.40	128	0.74
256	0.17	256	0.53

TABLE 1 Encoding Errors

RESULTS

ROAD SEGMENTATION RESULTS

The proposed method for road segmentation results in binary segmentations, which separates pavement and non-pavement areas. The following picture demonstrates one of the segmentation results in this research. Within Figure 5, the gray areas are removed from being considered by the segmentation algorithm, because they are far from roads.

FIGURE 5 Road Segmentation Result

Table 2 lists the accuracies of road segmentation.

Image Size	Accuracy
600X190	86.82%
600X190	86.87%
600X190	93.65%
600X190	93.78%
600X190	87.15%
600X190	88.05%
600X190	95.75%
2542X1475	91.78%
1652X806	90.56%

TABLE 2 The Accuracies of the Segmentation Results

In Table 2, all the accuracies are higher than 85%. The first seven images are small compared with the last two images. However, the accuracies are not affected by the image size.

Results of the LBG Algorithm

Three values summarize how well the LBG algorithm works. The first one is the average error between a cross section and its corresponding code entry. It is simply the Euclidean distance between a data entry and a code entry. The second and the third ones are the number of data entries before being quantized and the number of code entries after. Below is a table that shows these values.

Number of Cross Section Points	Average Error (pixel)	Number of Data Entries	Number of Code Entries
2	0.25	2615	275
4	0.36	6253	600

TABLE 3 Results of the LBG Algorithm

The differences between numbers of data entries and code entries are significant and the average errors are acceptable. This information can be easily converted into RDB and below is a snapshot of a road in the STS Scenario Editor.

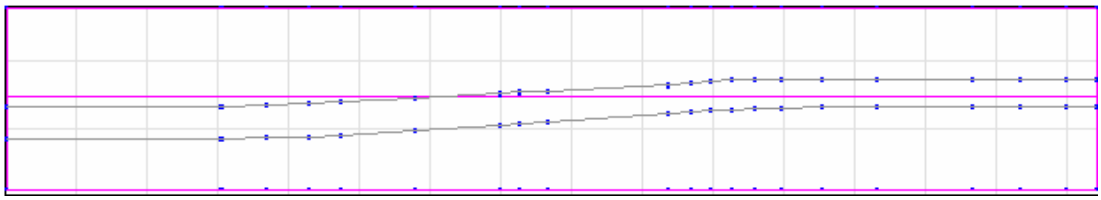

FIGURE 6 A Snapshot of STS Road Database

After determining repeatable cross sections, the next step is to create 3D road models with textures. During this step, some geometric design knowledge should be used, which include lane width and marker width. Below are two snapshots of 3D road models with textures.

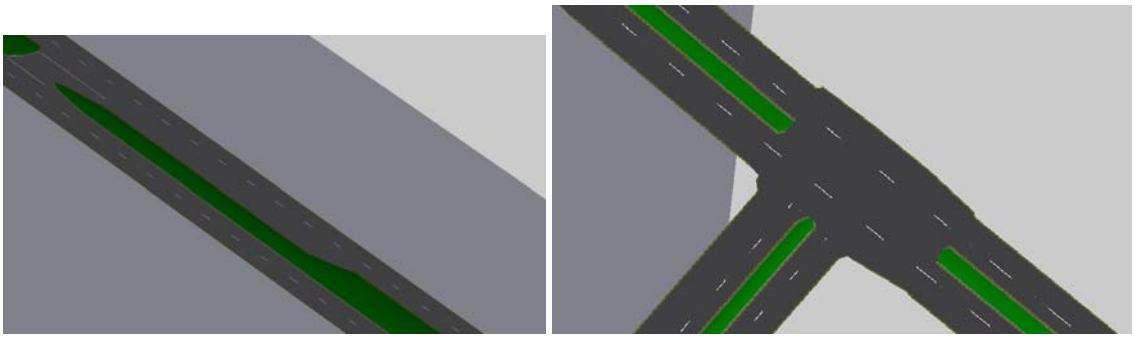

FIGURE 7 Snapshots of 3D Road Models with Textures

CONCLUSIONS

In this research, the problem of how to quickly develop 3D road databases for driving simulation is addressed. A method based on image understanding techniques is proposed. This method does not extract road center line information from images, instead it matches aerial images with the USGS DLG data, which is a collection of road center line point coordinates. Doing this excludes most unwanted contents from consideration. Before using USGS DLG data, it has to be matched with aerial photos in order to detect any distortion.

The chosen method for road segmentation is based on the mean-shift clustering algorithm. This method works within a feature space, where local modes are searched. Segmented classes in this step are categorized into two kinds, pavement and non-pavement by comparing their average saturation values. The proposed road segmentation method achieves high accuracies that are at least 85%. In the phase of creating 3D road models, repeatable cross sections are identified by using the LBG vector quantization algorithm. Afterwards, RDBs can be developed automatically.

An important aspect of this research is the use of USGS DLG information to remove a great portion of noise, irrelevant to road segmentation. This step enables the proposed method to avoid the problem of searching for roads in an image. Most road extraction

methods [7] start with the whole aerial image. Another unique feature of this research is automatically identifying repeatable cross sections using a vector quantization algorithm. This method can automatically reduce redundant information in creating road databases for an STS driving simulator system and quantitatively calculate average error. Additionally, this research uses geometric design standards when automatically mapping textures for road models. This proposed system greatly benefits from existing geographical data and knowledge. In the future, more existing geographical data should be studied.

REFERENCES:

[1] http://edc.usgs.gov/products/map/dlg.html#description, accessed on 11/15/2004
[2] D. Guo, H. I. Klee and A. R. Weeks "Segmentations of Road Area in High Resolution Images" IEEE Geoscience and Remote Sensing Symposium, Anchorage, AK, 20-24, September, 2004
[3] D. Comaniciu and P. Meer "Robust analysis of feature spaces: Color image segmentation" In International Conference on Computer Vision and Pattern Recognition, Pages 750-755. San Juan, Puerto Rico,1997
[4] A. R. Weeks "Fundamentals of Electronic Image Processing" SPIE/IEEE Series on Imaging Science & Engineering, 1996
[5] Multi-Gen Creator Pro Options Guide, MultiGen-Paradigm, Inc., 1999
[6] K. Sayood 1996 "Introduction to Data Compression", Morgan Kaufmann Publisher, Inc., San Francisco, CA
[7] D. Guo Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Central Florida, Summer, 2005