VEHICLE EXTRACTION USING HISTOGRAM AND GENETIC ALGORITHM BASED FUZZY IMAGE SEGMENTATION FROM HIGH RESOLUTION UAV AERIAL IMAGERY

LI Yu

Department of Geography, University of Waterloo, 602 White Cedar Ave., Waterloo, Ontario,Canada, N2V 2W2 - y62li@fes.uwaterloo.ca

Commission III, WG III/5

KEY WORDS: Vehicle Extraction, Histogram, Genetic Algorithm, Image Segmentation, Unmanned Aerial Vehicle (UAV)

ABSTRACT:

In this paper, an approach for extracting vehicles from UAV aerial imagery is given. The approach is based on a fuzzy segmentation algorithm, which combine fuzzy c-partition and genetic algorithm, and the geometric feature of vehicles for vehicle extraction. In this research, UAV colour imagery is examined experimentally. The results obtained demonstrate that most extracted vehicles match well with the original ones. From the analysis of results, it can be concluded that the proposed method is effective in both visual effect and positional accuracy.

1. INTRODUCTION

The effectiveness and usefulness of traffic management systems is dependent on accurately estimating the traffic flow conditions on traffic networks being monitored and managed. The ground based data are commonly available for such estimation. However, a limitation of the ground based methods for traffic data collection is that they rely on techniques that are strictly and spatially local in nature. For example, the cameras at fixed locations for traffic monitoring cannot easily observe the spatial progression and movement of traffic beyond the field of view since these camera platforms are not mobile.

Airborne based data have the potential to significantly enhance the quality of traffic condition estimations due to their spatial scale and connectivity (Grejner-Brzezinska *et a*l., 2007). To date there have been growing research activities in using of aerial imagery for transportation management, examples include the excellent summary in Kumar *et al*. (2001) and Angel *et al*. (2003). Most of researches examined the use of conventional aircrafts such as helicopter and manned light plane to collect aerial imagery. Unfortunately, they have a limit capability to acquire low cost and timely imagery.

The use of high resolution aerial imagery acquired from unmanned aerial vehicle (UAV) would be attractive for studying, modelling, and monitoring traffic flow, since the UAV platform allow for the easy, cost effective and timely acquisition of high resolution and wide spatial coverage imagery unobtainable from ground based sensors. To gain acceptance for using UAV aerial imagery in traffic applications, it is necessary to demonstrate that vehicles can indeed be identified, extracted and classified accurately from the imagery and to establish a framework for data fusion with the ground based traffic information.

In this light, the objective of this research is to develop vehicle extraction approach to demonstrate the ability to identify and extract vehicles from high resolution UAV imagery. This approach is following two stages for vehicle extraction. Firstly, the colour UVA aerial imagery is segmented by a proposed segmentation algorithm. The designed segmentation algorithm is based on fuzzy c-partition, in which the colour histogram is used for colour space analysis to obtain a proper initial estimate of centre positions, then genetic algorithm is employed to optimally cluster the colour space data point projected from an input colour imagery, hence optimal colour segmentation can be achieved. Secondly, the post-processing procedure is carried out on segmented image by use binary mathematical morphology operators to extract the outlines of vehicles.

The paper is organized as follows. In the next section, the basic concepts from fuzzy c-partition colour histogram, genetic algorithm and colour histogram are introduced. In the third section, the proposed approach to vehicle extraction is described. The approach is applied to UAV aerial colour imagery in the fourth section. Conclusion about the approach is given in the fifth section.

2. BACKGROUND

2.1 Fuzzy C-Partition

According to the fuzzy paradigm (George and Bo, 1995), fuzzy clustering is seen as partitioning a data space into a number of fuzzy sets and assigning each data point a membership to each cluster. Consider a vector set V formed by n vectors in L dimensional real number space R^L, i.e., $V = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$, $\boldsymbol{v}_j = (v_{j1}, \dots, v_{jL}) \in R^L$ and $j = 1, 2, \dots, n$, a fuzzy c-partition on V is represented by the fuzzy partition matrix $P = [p_{ij}]$, $i = 1, \dots, c$, where c is the positive integer to indicate the number of the clusters in the partition, and $p_{ij} \in [0, 1]$ is the fuzzy membership value of $\boldsymbol{v}_j$ belonging to ith cluster and satisfy,

$$\sum_{i=1}^{c} p_{ij} = 1, \text{ for all } j = 1,\dots,n \tag{1}$$

$$0 < \sum_{j=1}^{n} p_{ij} \le n, \text{ for all } i = 1,..,c \tag{2}$$

Before using fuzzy c-partition to design a clustering algorithm, the following two issues should be solved. First one is how to determinate the number of clusters for a clustering. Another issue is how to calculate the fuzzy c-partition matrix. In this paper the number of cluster is determined by user a prior. The second issue is solved by calculating colour similarity measurement as follows.

For a given vector set $V = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$, the fuzzy c-partition matrix can be calculated as follows.

$$p_{ij} = \frac{\mu(\boldsymbol{u}_i, \boldsymbol{v}_j)^{\frac{1}{m-1}}}{\sum_{k=1}^{c} \mu(\boldsymbol{u}_k, \boldsymbol{v}_j)^{\frac{1}{m-1}}} \qquad (3)$$

where $m \in (1, \infty)$ is the weighting exponent on each fuzzy membership. The larger m is the fuzzier the partition is, $\boldsymbol{u}_i$ is central vector for cluster i, and $\mu(\boldsymbol{u}_i, \boldsymbol{v}_j)$ is a similarity measure between vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ and can be calculated by

$$\mu(\boldsymbol{u}_i, \boldsymbol{v}_j) = \exp(-k_1 d(\boldsymbol{u}_i, \boldsymbol{v}_j)) \cos(k_2 \theta(\boldsymbol{u}_i, \boldsymbol{v}_j)) \qquad (4)$$

where k_1 and k_2 are parameters and $d(\boldsymbol{u}_i, \boldsymbol{v}_j)$ and $\theta(\boldsymbol{u}_i, \boldsymbol{v}_j)$ are the distance and the angle between $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ as follows,

$$d(\boldsymbol{u}_i, \boldsymbol{v}_j) = \left(\sum_{l=1}^{L} | u_{il} - v_{jl} |^2 \right)^{1/2} \qquad (5)$$

$$\theta(\boldsymbol{u}_i, \boldsymbol{v}_j) = \arccos \left(\frac{\sum_{l=1}^{L} u_{il} v_{jl}}{\sqrt{\sum_{l=1}^{L} u_{il} \sum_{l=1}^{L} v_{jl}}^{2}} \right) \qquad (6).$$

2.2 Genetic Algorithm

Genetic Algorithms (GA) are computer procedures that employ the mechanics of natural selection and natural genetics to evolve solutions to problems (Goldberg, 1989). Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem encoded in some fashion, and a metric called a fitness function that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random. In a pool of randomly generated candidates, these promising candidates are kept and allowed to reproduce by evaluating each candidate according to the fitness function using a series of genetic operations: selection, crossover and mutation.

There are many different techniques by which a genetic algorithm can be used to select the individuals to be copied over into the next generation. Roulette-wheel selection is one of the most common methods. It is a form of fitness proportionate selection in which the chance of an individual's being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness. Conceptually, this can be represented as a game of roulette, each individual gets a slice of the wheel, but more fit ones get larger slices than less fit ones. The wheel is then spun, and whichever individual owns the section on which it lands each time is chosen. Once selection has chosen fit individuals, they must be randomly altered in hopes of improving their fitness for the next generation. There are two basic strategies to accomplish this. The first and simplest is called mutation. Just as mutation in living things changes one gene to another, so mutation in a genetic algorithm causes small alterations at single points in an individual's code. The second method is called crossover, and entails choosing two individuals to swap segments of their code, producing artificial offspring that are combinations of their parents. This process is intended to simulate the analogous process of recombination that occurs to chromosomes during sexual reproduction. Common forms of crossover include single-point crossover, in which a point of exchange is set at a random location in the two individuals' genomes, and one individual contributes all its code from before that point and the other contributes all its code from after that point to produce an offspring.

2.3 Colour Histogram

Colour histogram is an important technique in colour image analysis, because of its efficiency, effectiveness and triviality in computation (Pratt, 1991). Generally speaking, a colour histogram represents the statistical distribution of the colours in a colour image on all colours in a colour space.

Given a colour space divided into I colour bins, the colour histogram of the colour image with n pixels is represented as a vector $H = [h_0, \dots, h_{I-1}]$, in which each entry h_i indicates the statistical figures of the colours in the colour image which belong to the ith bin, i.e.,

$$h_i = \frac{n_i}{n}, \ \mathrm{i} = 0, 1, , \mathrm{I} - 1 \qquad (7)$$

where n_i is the number of pixels with colours in the ith colour bin.

Let RGB colour space be discretized along the R, G, and B axes by the numbers N_R, N_G, and N_B, respectively. Then the total N ($= N_R \times N_G \times N_B$) bins are available. These bins are coded in such a sequence from R to G and then from G to B. According to the specified discretizing and coding scheme the index of each bin can be represented as

$$i = R + N_G \times G + N_B{}^2 \times B \qquad (8)$$

where $R = 0, 1, \dots, N_R-1$, $G = 0, 1, \dots, N_G-1$, and $B = 0, 1, \dots, N_B-1$.

Then the pixel (r_p, g_p, b_p) will be in the bin with the index i_p,

$$i_p = \left\lfloor \frac{r_p N_R}{256} \right\rfloor + N_G \times \left\lfloor \frac{g_p N_G}{256} \right\rfloor + N_B{}^2 \times \left\lfloor \frac{b_p N_B}{256} \right\rfloor \qquad (9)$$

where $\lfloor \ \rfloor$ is an integral operator.

3. DESCRIPTION OF PROPOSED APPROACH

3.1 Fuzzy Segmentation

Based on the concept of the fuzzy c-partition above mentioned, the colour segmentation approach is designed. The approach consists of three steps: (1) Pre-clustering. This process includes finding an initial centre vector set U_0 and indicating the ranges in which the centre vectors are chosen in the following optimal procedure. This procedure is finished by using a histogram-based technique, (2) Searching the best fuzzy c-partition. It is realized by genetic algorithm to find a optimal fuzzy c-partition matrix, (3) Defuzzifing procedure. It converts the fuzzy c-partition matrix to the crisp c-partition matrix.

3.1.1 Pre-clustering: For the given colour image *C*, the colour histogram *H*(*C*) can be obtained by method described in 2.3 Section. It is obvious that if an image is composed of distinct objects with different colours, its colour histogram usually shows peaks. Each peak corresponds to one object and adjacent peaks are likely to be separated by a valley. The height of a peak implies the number of the pixels falling in the bin corresponding to the location of the peak.

After the number of the clusters in the fuzzy c-partition c is chosen by user, the *c* highest peaks in the colour histogram can be detected and the bins corresponding to the detected peaks determine the ranges in which the centre vectors are investigated for the purpose of the optimization. The initial centre vectors are randomly selected in each of *c* bins.

3.1.2 Optimizing: To search the optimal resolution of the fuzzy c-partition, a genetic algorithm is utilized and designed as follows.

Chromosome Representation

For a given m-dimensional vector set, the chromosome in a population is represented by a string consisting of c m-dimensional real vectors which encode the center vectors corresponding to c clusters in a c-partition. Figure 1 gives an example of such string where *U* donates the centre vector set corresponding to a string with *c* centre vectors.

u_1	u_2	……	u_j	……	u_c

Figure 1 String consisting of *c* vectors

Population Initialization

In the initial population, the string vectors are the colour vectors randomly selecting from each bin of *c* bins mentioned in 2.3 Section The number of strings in the population *N* is given by users**.**

Fitness Computation

In order to use the genetic algorithm, it is necessary to define an objective function. In this paper, the aggregation similarity for a set of centre vectors to all vectors in the considered vector set is used to be the function

$$\frac{1}{c}\sum_{i=1}^{c}\sum_{j=1}^{n} p_{ij} \tag{10}$$

The fitness of each chromosome in the population is evaluated with the objection function. For each chromosome, the center vectors encoded in it are first evaluated, and then the fuzzy c-partition matrix corresponding to the chromosome is calculated by using the Equation (3).

Genetic Operations

Three kinds of genetic operators are used in this genetic algorithm, selection, crossover and Mutation. The conventional roulette wheel method is used for selection. Then one-point crossover is applied to selected two strings to generate two offspring. The crossover operation is applied stochastically with probability p_c. After crossover, the offspring is considered for mutation. In this algorithm, the mutation is carried out by replacing strings in current generation to new strings selected from the same bins in which the original strings are. The mutation is performed with a fixed probability p_m.

Stopping Criterion

There are two stopping criterion for the GA. Firstly, after some number of generations without improvement in the solution pool, the algorithm terminates. Secondly, setting a maximum iteration, after the iteration the algorithm terminates.

3.1.3 Defuzzifing

In order to obtain the segmented image, it is necessary to transform the fuzzy c-partition matrix to the crisp partition matrix. In this study, the following defuzzification scheme is used.

Let $P = [p_{ij}]$ $i = 1, …, c$ and $j = 1, …, n$ be the fuzzy c-partition matrix, it is well known that p_{ij} presents the membership grade for pixel *j* belonging to cluster *i*. A percent partition matrix, P_p, is defined as

$$p_{pij} = \frac{p_{ij}}{\sum_{j=1}^{n} p_{ij}} \tag{11}$$

In terms of the percent partition, the crisp partition matrix, $P_c = [p_{cij}]$, is defined as

$$p_{cij} = \begin{cases} 1, & p_{pij} = \max_{i=1}^{c}(p_{pij}) \\ 0, & otherwise \end{cases} \tag{12}$$

It is clear that in the crisp-partition matrix each pixel belongs to a certain cluster.

3.2 Object Extractions

Once the segmented images are obtained by the above segmentation algorithm, the binary object image can be extracted by selecting the pseudo-colour corresponding to the object regions. In general, the objects in the binary image are corrupted by noise objects, which have the similar colour to objects. In order to make the object regions clear, it is necessary to filter the corrupted object image. To this end, binary morphological operations are used. For example, depending on the shapes of noise objects, the appropriate combinations of binary dilation, erosion, opening, and closing should be chosen.

3.3 Delineation of Vehicle Outline

To extract the building regions according to the colour features of the buildings and uses an edge extraction algorithm to detect the skeletons of the detected buildings. To this end, a boundary extractor is designed and described in this section.

Following the definition of 8-neighborhood shown in Figure 2, the boundary pixel for building is determined if it is a contour pixel and satisfies the following condition, $0 < N(p) < 8$, where $N(p)$ is the number of nonzero neighbors of pixel p, i.e.,

$$N(p) = \sum_{i=0}^{7} p_i \tag{13}$$

p_7	p_0	p_1
p_6	p	p_2
p_5	p_4	p_3

Figure 2. Neighbourhoods arrangement

4. EXPERIMENTS AND RESULTS

The proposed road extraction algorithm has been tested on UAV aerial imagery, Figure 3 shows two testing images.

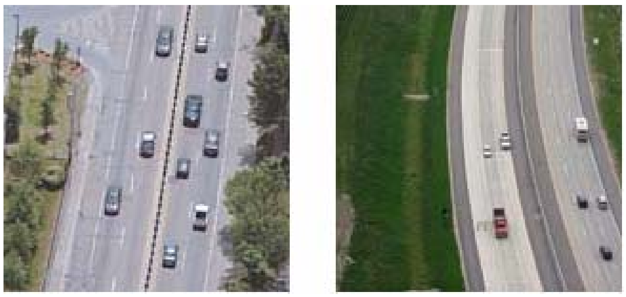

Figure 3. Testing Imagery.

In colour segmentation operation, the parameters k_1 and k_2 in Equation (5) for calculation of the colour similarities are chosen to be 0.0001 and 0.2, respectively. The number of the clusters is set as 4 and the maximum iterates is 50. The crossover and mutation probability are 20%. Figure 4 gives the results of the colour segmentation.

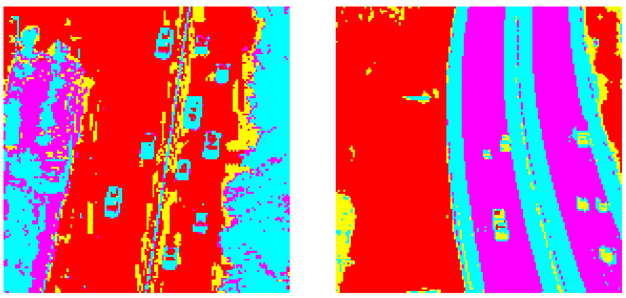

Figure 4. Segmented Imagery

In the designed vehicle extraction approach, we try to use the uniform of radiations from road. In this case, the road cluster is selected and the vehicle in the segmented imagery can be seen as "noise". Figure 5 shows the binary road imagery.

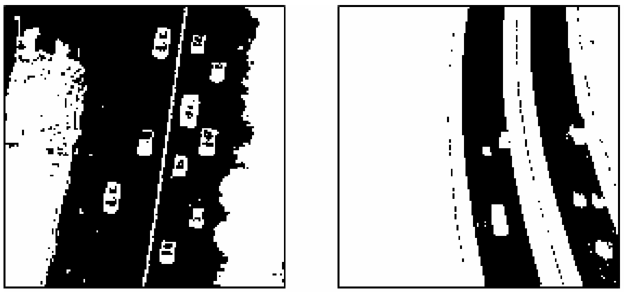

Figure 5. Binary Road Imagery

Above road imagery is filtered by binary morphological open operation to obtain solid road. In this experiment, the structuring element is set as 5. The result of the morphological operation is in Figure 6.

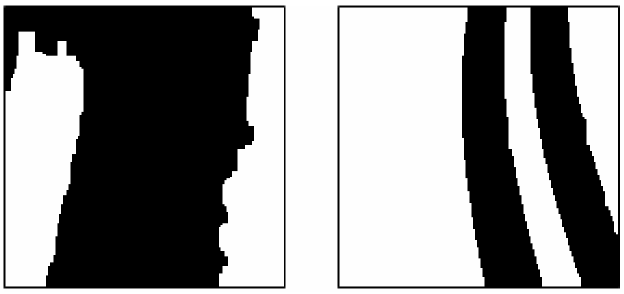

Figure 6. Filtering Binary Road Imagery

In order to extract the vehicles from imagery in Figure 5, we subtract the imagery in Figure 5 from the imagery in Figure 6. Figure 7 demonstrates the vehicles extracted by the subtraction operation.

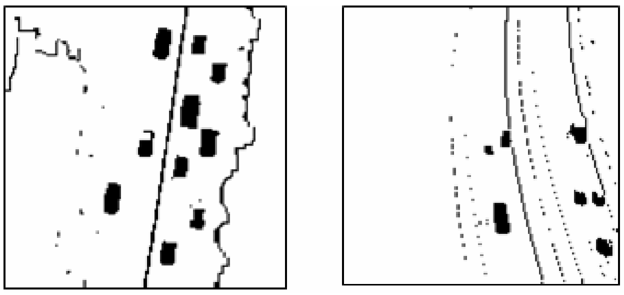

Figure 7. Binary Vehicle Imagery

The binary morphological operations are carried on the vehicle imagery to obtain complete vehicles as shown in Figure 8.

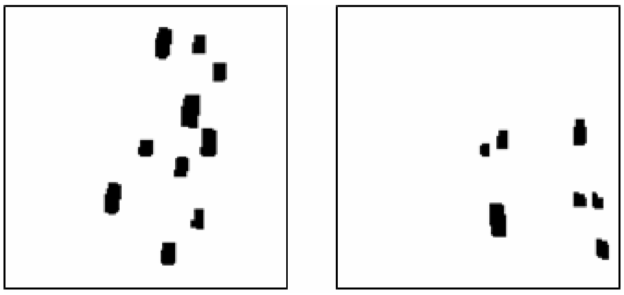

Figure 8. Filtering Binary Vehicle Imagery

The edges of the extracted vehicles are delineated using the thinning algorithm discussed above, and the results are shown in Figure 9.

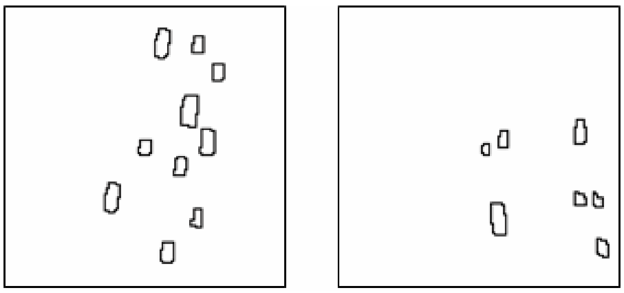

Figure 9. Outlines of Extracted Vehicle

In order to illustrate the accuracy, the outlines of the extracted vehicles are overlaid on the original imagery, see Figure 10. In the overlay imagery the thin red lines indicate the vehicle outlines. It can be obverted from Figure 10 that most vehicle outlines match well the vehicles.

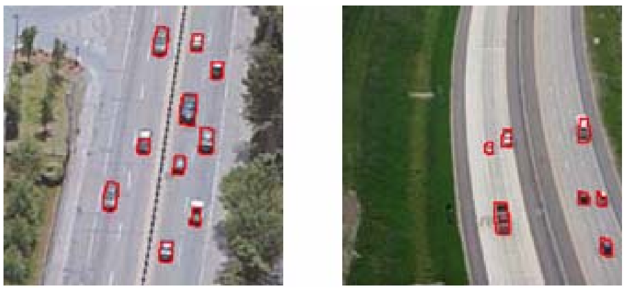

Figure 10.Overlaying Vehicle Outlines on Original Imagery

5. CONCLUSIONS

In this paper, colour related information is used for imagery segmentation purposes. The proposed algorithm combines fuzzy c-partition and the genetic algorithm to find fuzzy c-partition matrix.

In order extract the vehicle from UAV aerial imagery, we use the uniform radiation of the roads and geometric feature of vehicles.

Several experimental examples show the applicability of this approach to vehicle information extraction and the information can be used for traffic flow computation and vehicle classification.

ACKNOWLEDGEMENTS

The research was partially supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) discovery grant. The PhD research scholarships provided by the Public Safety And Emergency Preparedness Canada Research Fellowship, the Ontario Graduate Scholarship and School of Graduate Studies at University of Waterloo are also acknowledged.

REFERENCES

Angel, A., M. Hickman, P. Michandani and D. Chandnani, 2003. Methods of analyzing traffic imagery collected from aerial platforms, *IEEE Transactions on Intelligent Transportation Systems*, 4(2): 99-107.

George, J. K. and Y. Bo, 1995. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall PTR, Upper Saddle River, New Jersey, USA.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

Grejner-Brzezinska, D., C. Toth and E. Paska, 2007. Airborne remote sensing supporting traffic flow estimates, C. V. Tao and J. Li (eds) *Advances in Mobile Mapping Technology*, Taylor & Francis, London.

Kumar, R., R., H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Gao, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt, 2001. Aerial video surveillance and exploitation," *Process IEEE*, 99: 1518-1539.

Pratt, W. K., 1991. *Digital Image Processing*, Wiley, New York, USA.