Real-time Vehicle Detection and Tracking Using a Mean-Shift Based Blob Analysis and Tracking Approach

Rhen Anjerome Bedruz[#1], Edwin Sybingco, Argel Bandala, Ana Riza Quiros, Aaron Christian Uy, Elmer Dadios *Electronics and Communications Engineering Department, De La Salle University*
2401 Taft Avenue, Malate, Manila, Philippines

***Abstract*—Implementing computer vision on traffic scenarios are one of the most widely sought area in the field of vision research. In dealing with the surveillance in traffic scenarios, every vehicle in the scene must be observed which results to problem arising from instances whenever the traffic density in an area is high due to occlusion caused by the large number of vehicles being observed. Thus, this paper proposes a vehicle detection and tracking algorithm whose main purpose is to detect and track vehicles entering an intersection and track them robustly in real-time. The algorithm which was used is a blob analysis and tracking based on a mean-shift kernel. The blob approach acts as the main tracking and will use the mean-shift in the event of blob merging or occlusion. In this paper, the proposed tracking method is tested using a CCTV camera on an intersection with high traffic density to illustrate the capability of solving occlusion and observe the robustness of the algorithm in the scene. The results show that the proposed system successfully tracks the vehicles during and after occlusion with other vehicles or other types of objects in the scene.**

***Index Terms*— Vehicle Detection, Vehicle Tracking, Blob Analysis, Blob Tracking and Mean-Shift Algorithm**

I. Introduction

Intelligent Traffic System is a traffic system which is formed from the integration and management with all the components of a traffic including vehicles, drivers, and even pedestrian through the use of a variety of medium such as wireless sensor networks, RFID, machine vision, and computer vision [1] [2] [3] [4] . One of its main goal is to apprehend traffic law violators by adding and implementing a contactless apprehension component in the system [5]. The contactless apprehension component of the ITS uses computer and machine vision comprises of three parts which are the vehicle identification, violation detection, and vehicle plate recognition, respectively [6]. Vehicle identification involves the detection and tracking of vehicle along the road while profiling or classifying them according to their vehicle types and color [7] [8]. Violation detection involves the analysis of the behavior of all the vehicles tracked while identifying all the traffic violations each vehicle committed on the road. Vehicle plate recognition involves plate localization and plate identification using optical character recognition [9]. Among these three parts, the most crucial part is the vehicle detection since the other two parts cannot be implemented because there will be no detected vehicle to be evaluated. In this regard, the main problems to be addressed are the vehicle detection, and vehicle tracking respectively [10].

In that regard, vehicle detection and tracking is difficult to implement in real time in a traffic congested area. This is because vehicles clutter together having an insignificant distance with one another makes it hard for the computer to separate the vehicles [11]. In addition, tracking is difficult due to vehicle congestion which leads to the occlusion of vehicles in the camera's point of view. These are the current problems which must be addressed for implementing the Intelligent Traffic System.

This paper presents a way to minimize the effect of occlusion on dealing with real-time vehicle detection and tracking. The vehicle detection system will employ the detection of vehicles going to the region of interest (ROI). The technique used in the vehicle detection system will employ different processes. These processes are the background modelling, background subtraction, Otsu binarization [12] and blob clustering. The approach will take advantage of vehicle detection which is used to locate moving vehicles prior to vehicle tracking. On the other hand, the vehicle tracking system will take care of the tracking of vehicles detected by the vehicle detection system. The technique used in the vehicle tracking system will take advantage of the quick processing time of blob analysis approach which was embedded with a mean-shift kernel to remove inaccuracies from occlusion.

II. Vehicle Detection

As aforementioned, vehicle detection involves the detection of all the vehicles entering a specific ROI. The ROI which was used in the study was based on polygon filling operations. After setting the ROI, background modelling, background subtraction, Otsu binarization and blob clustering was used sequentially. To summarize, vehicle detection block diagram is shown on Figure 1.

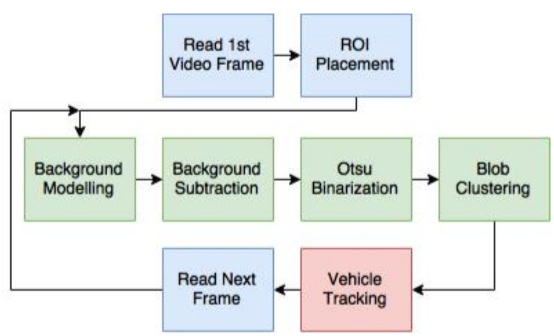


Figure 1. Vehicle Detection Block Diagram

As shown on the figure, ROI was initialized and placed on the first frame. The succeeding frames then used the placed region. Meanwhile, the second row on the figure shows the processes of the vehicle detection used. It can be seen that after the blob clustering, vehicle tracking on the current frame and vehicle detection on the next frame will be done consecutively.

A. Background Modelling

The study took into consideration two background modelling methods. These were the Mixture of Gaussian (MOG) and the K-Nearest Neighbor (KNN) background modelling techniques. The difference between the two methods is the way on how they determine the reference frame for background subtraction. So, the

978-1-5386-0912-5/17/$31.00 ©2017 IEEE

video streamed real-time in succession is noted to have number of frames which increases as time goes by. So, each pixel location in every image of each frame can be represented using (, ,) for = (1,2, … ,) [13]. The background must be represented by a separate image which is represented as the reference frame which was obtained using MOG or KNN. This is represented using the function (, ,) where is the reference frame. A counter for each pixel location in the accumulative image is incremented every time a difference occurs at that pixel location between the reference and the image in the sequence. So, when the -th frame is being compared with the reference, the entry in a given pixel of the accumulative image gives the number of times when the gray level of a frame varies from that of the corresponding pixel value in the reference image.

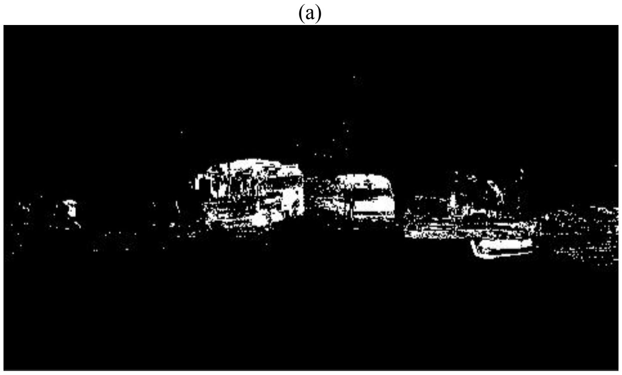
B. Background Subtraction and Otsu Binarization

The background subtraction operation was used to detect moving vehicle with background model acquired by difference accumulative method mentioned above. It will be assumed that the background difference image after background subtraction operation is already obtained. Then, Otsu [14] method will be used to threshold the background difference image. From this, binary image value of each pixel is 0 or 1. The pixel value 1 stands for background and 0 stands for foreground. Furthermore, the pixel value 1 corresponds to white color and 0 corresponds to black color.

$$F(x,y,t_i) = \begin{cases} 1 & |f(x,y,t_i) - f(x,y,t_r)| > T_f \\ 0 & otherwise \end{cases} \quad (1)$$

The variable denotes the threshold which is used to determine the maximum deviation for determining the change in the gray level of the current frame from the reference. This differentiates the background and the foreground. To visualize the difference between MOG and KNN, a sample scenario as shown on Figure 2a was subjected to background modelling, background subtraction and Otsu binarization.

(a)



(b)

(c)

Figure 2a. Sample for Background Modelling; Figure 2b. MOG Background Modelling output; Figure 2c. KNN Background Modelling output

Figure 2b shows the actual binary image output of the Mixture of Gaussian background subtraction for the sample traffic scenario shown. It can be seen on the figure that it has a high noise elimination but also shows a moderate foreground elimination. Thus, reconstruction through clustering must be used. Subsequently, the next figure shows the actual binary image output of the K-nearest neighbor subtraction for the sample traffic scenario shown. It can be seen on the next figure that it has a low noise elimination but shows a minimal foreground elimination

C. Blob Clustering

In order to find out each vehicle's exact spatial position, locating each vehicle after getting binary image, f(x,y,t), is necessary. So, the blob clustering algorithm was used to help locate vehicles [15]. In the algorithm, each data point is considered as a potential cluster center. Blob Clustering is a technique which was based on analysis of consistent image regions, or commonly known as static background. As such, it is also method for applications wherein the objects being inspected are clearly discernible from the background. Diverse set of Blob Clustering methods allows to create tailored solutions for a wide range of visual inspection problems, yet only one technique is discussed and applied later in this study. The main advantages of this technique include high flexibility and excellent performance; however, its limitations are: clear background-foreground relation requirement and pixel-precision [16].

The Blob Clustering algorithm used consists of three steps, which are extraction, refinement, and analysis, respectively. Two consecutive frames in a video were converted to image's greyscale from RGB, using the luminance equation below [17]:

$$Y = 0.299\,R + 0.587\,G + 0.114\,B \quad (2)$$

The next procedure was to design a Gaussian filter. For the smoothing of the two frames, shown below is the equation describing the convolution process [18]. Thus, done using a 2-dimensional Gaussian Filter. The Gaussian function which will be used is:

$$G(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

Since the extracted region was often flawed by noise of various kind due to inconsistent lightning or poor image quality, the refinement of the image took place. Typically, region morphology is used for such transformation [16]. After such process, morphological transformations, in particular, dilate and erode were used. The mathematical approach to the morphological dilation and erosion is that it sets a pixel at (i, j) to the minimum over all pixels in the neighborhood centered at (i, j) as shown in the equation[17]:

$$I_{Dilate/Erode}(x,y) = \min_{(x',y'):element(x',y')\neq 0} I(x+x', y+y') \quad (4)$$

where, x' and y' are the neighboring row and column pixels in the subjected region of the image. In effect, erosion enlarges dark regions and shrinks bright regions. For this case, the kernel was scanned over the image, then computed the minimal pixel value overlapped and replaced the image pixel under the anchor point with that minimal value. After the refinement, the contour of the object captured from the two consecutive frames were connected so that it will form a blob. Further, every enclosures will be subjected to a bounding convex hull [19].

III. Vehicle Tracking

After the vehicles entering were detected and clustered. Blobs were analyzed and track using the Blob analysis embedded with a Mean-shift Kernel. The Mean-shift Kernel used a background weighted histogram in order to determine the centroid at a higher accuracy as compared to an ordinary Mean-shift Kernel.

A. Mean-Shift Algorithm

Mean shift is an algorithm that iteratively shifts a data point to the average of data points in its neighborhood. It is widely-known since it is useful for clustering, mode seeking, probability density estimation, tracking, etc. is a semi-automatic tracking method. Normally, the desired object is manually selected in the initial frame. The inside pixel location of the rectangles is presented as for = 1,2,3, … , . The selected area is then considered as the object model. Afterwards, the color histogram will be calculated using the equation:

$$\widehat{q_u} = C\sum_{i=1}^{n} k(x_i^2)\,\delta[b(x_i)-u] \quad (5)$$

In (5), u represents the feature value of the target model, $b(xi)$ associates the pixel xi to the histogram bin. The function δ is the Kronecker delta function. The function $k(x)$ is a kernel function and C is the histogram coefficient. The histogram coefficient is calculate using the equations:

$$C = \frac{1}{\sum_{i=1}^{n} k(x_i^2)} \quad (6)$$

$$\hat{p}_u(y) = C_h\left(\sum_{i=1}^{n} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)\right)\delta[b(x_i^*)-u] \quad (7)$$

The Bhattacharyya coefficient is used to find the dissimilarities between the target in current frame and previous frame. Maximizing the Bhattacharyya coefficient minimizes the dissimilarities. In essence, it is a metric which aims to estimate the similarity between the target model and the candidate model. When there is a greater similarity between the two density distributions, the value for will yield a higher value. In order to obtain the new location of y, obtaining the maxima of the density estimation is vital. This is done by using the Mean-Shift Vector:

$$y_1 = \frac{\sum_{i=1}^{n_k} x_i w_i g\left(\left|\frac{(y_0-x_i)}{h}\right|^2\right)}{\sum_{i=1}^{n_k} w_i g\left(\left|\frac{(y_0-x_i)}{h}\right|^2\right)} \quad (8)$$

$$w_i = \sum_{u=1}^{m}\delta[b(x_i)-u]\sqrt{\frac{q_u}{p_u(y_0)}},\quad g(x) = -k'(x) \quad (9)$$

The Mean-Shift vector points towards direction of maximum density. This is used to calculate the mean of all points inside the region of interest, then move the initial point to the mean as the new initial point, and repeat the procedure until it converges to maximum density area.

B. Background Weighted Histogram

In tracking of targets, the background information is often included in the detected target region. If the correlation between target and background is high, the localization accuracy of the object will be decreased. The interference of salient background features in target localization can be reduced by a representation model of background features [21]. For this case, the background-weighted histogram (BWH) algorithm can be used which selects discriminative features from the target region and the target candidate region. A target model can be solved using the equation:

$$\hat{q}'_u = C' v_u \sum_{i=1}^{n} k(|x_i|^2)\,\delta(b(x_i)-u) \quad (10)$$

The weight of point computed by BWH in the target candidate region can be derived from on equation (11) using the new target and candidate model. The equation will then be:

$$w'_i = \sum_{u=1}^{m}\delta[b(x_i)-u]\sqrt{\frac{q'_u}{p'_u(y)}} \quad (11)$$

The above BWH transformation aims to reduce the effects of prominent background features in the target candidate region on the target localization.

C. Vehicle Tracking Process

The Mean-Shift vector obtained from (9) was used to locate the Mean of all points of a single vehicle. Using BWH, the Mean will be optimized. The new point obtained is the adjusted mean order \P{{ pweighted centroid of the detected vehicle. A visualization of the Mean (red mark) and the weighted centroid (green mark) of the vehicle is shown on the next figure.

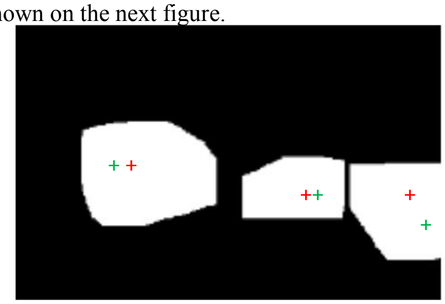


Figure 3. Mean and Weighted Centroid points

In the event wherein clustered blob merges, or a certain blob identified another vehicle on the scene as part of the blob, the change in centroid will be observed and evaluated. In particular, if a blob changes its centroid by more than 6 pixels, it will retain its old centroid position and declare a new blob. As an example, on the instance that the middle and the right blobs (as shown on figure 4) clusters with each other, there will be a new centroid which will be obtained. The next figure shows the new centroid obtained(red mark) as well as the old centroids (green mark).

Figure 4. New Centroid due to Clustering

In this case, if the distance between the green marks and the red mark are more than 6 pixels, then it will still retain the old blob size forcing it to regroup the blob into two separate groups. This is the way how the algorithm deals with the high volume of vehicles in a certain scenario. Accuracy of the tracking will be observed using the equation:

$$A_{track} = \frac{T_{track}}{T_{actual}} \quad (12)$$

The tracking accuracy value, A, will determine whether the vehicle is fully tracked, partially tracked or untracked by using thresholding values. The thresholding values are 0.8 for fully tracked vehicles and 0.2 for untracked vehicles. Tracking values from 0.2 up to 0.8 are considered as the partially tracked vehicles.

IV. RESULTS

A. *Vehicle Detection*

Initially, both KNN and MOG algorithms were tested using varying history length and varying neighbor sizes ranging. The history length varies from 1 frame to 100 frames while the neighbor sizes varies from 10 pixels up to 100 pixels. The processing time for the first 100 frames were computed and recorded. It was observed that the processing time for Mixture of Gaussian is much lower as compared to the K-Nearest Neighbor algorithm. The Mixture of Gaussian algorithm took an average of 0.0035 seconds over the K-nearest neighbor algorithm which took an average of 0.02937 seconds.

Meanwhile, the blob clustering algorithm takes the last stream of the processing time for vehicle detection. It can be seen that the clustering algorithm for the first 100 frames are shown on the next figure.

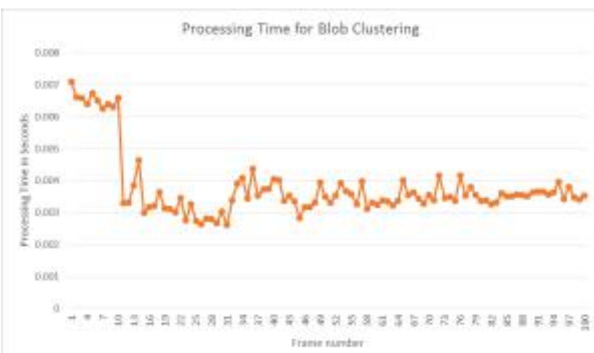


Figure 5. Processing Time for Blob Clustering

The blob clustering algorithm takes an average of 3.772 milliseconds in order to perform clustering. Thus, these clusters were the detected vehicles or a large group of pedestrian walking in the area. These clusters were encapsulated in a red bounding box which is shown in the next figure. Blobs entering the scene are tagged with a blob number since the license plate of the vehicle has not been gathered at this point. The next table shows the confusion matrix derived from the compiled from the experimental traffic scenario. The total number of objects found in the scenario were 102; this includes but is not limited to vehicles, pedestrian, motorcycles, and tricycles. The true negative state describes that there were 11 non-vehicle objects that was not detected by the system. On the other hand, the false positive condition defines that there were 8 detected vehicles that are in fact non-vehicles. The true positive state portrays that there were 83 vehicles that are indeed vehicles. Fortunately, there were no vehicles that was not detected, as this describes the false negative condition. Given the following data, the accuracy that declares how often the vehicle detection system is correct, can be computed as 92.157%. Whereas, the precision that dictates the precise detection rate of the system, is calculated as 91.209%.

Table 1. Confustion Matrix for Clustering

N=102	Detected: No	Detected: Yes
Actual: No	11	8
Actual: Yes	0	83

B. *Vehicle Tracking*

The processing time per frame for the first 485 frames was shown on the next Figure. It was observed that the average runtime yielded a value of 1.77 milliseconds which can be considered garnering a real-time effect.

Figure 6.Mean Shift Tracking Processing Time

Using equation 24, the value of A_{Track} is plotted per vehicle and shown on the next figure. The $A_{SysTrack}$ computed yielded a value of 87.334%. which is considered a high value for $A_{SysTrack}$. This is visualized on the next figure.

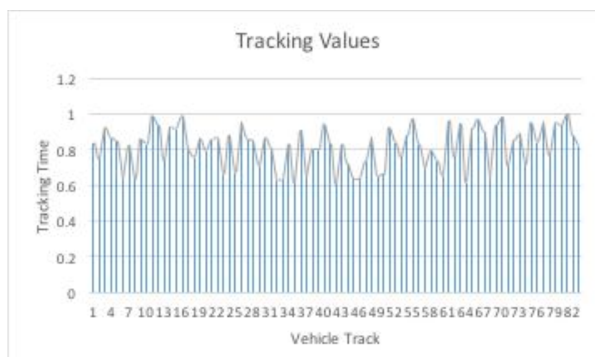


Figure 7. Tracking Values

V. CONCLUSIONS AND RECOMMENDATIONS

It can be concluded in this study that it is possible to implement and ITS and detect vehicles using blob clustering and incorporate mean-shift algorithm in order to track them. A good method for background subtraction is the MOG due to high noise elimination even though the foreground rejection is high. In addition, a high foreground rejection can be compensated through reconstruction using morphological transformation. In performing erosion and dilation, there are key factors which must be taken consideration such as foreground rejection and reconstruction rate. This method of identification and tracking can harness real-time which was also accurate.

Further researches can be done by exploring feature-based tracking techniques by making them more robust for real-time applications. These feature-based tracking techniques can also be of help to reduce the processing time for the vehicle classifications to compare the robustness of that system to the system created.

VI.

VII. Bibliography

[1] L. Peng, F. Xiao and Z. Ni, "Design for wireless sensor network-based intelligent public transportation system," in *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, Hong Kong, 2009.

[2] I. Masaki, "Machine-vision systems for intelligent transportation systems," *n IEEE Intelligent Systems and their Applications,* vol. 13, no. 6, pp. 24-31, 1998.

[3] P. S. P. Wang, "Keynote Speaker 1: IPR and Applications to Computer Vision - Security, Safer Transportation, and Greener World, by Patrick S.P. Wang," in *8th Asia Modelling Symposium*, Taipei, 2014.

[4] S. P. Biswas, P. Roy and N. Patra, "Intelligent Traffic Monitoring System," *Advances in Intelligent Systems and Computing,* vol. 2, no. 58, pp. 535-540, 2015.

[5] R. Ramirez, "Quezon City to adopt contactless apprehension for traffic violators," The Philippine Star, 22 September 2015. [Online]. Available: http://www.philstar.com:8080/metro/2015/09/22/1502606/quezon-city-adopt-contactless-apprehension-traffic-violators. [Accessed 28 July 2016].

[6] A. C. P. Uy, R. A. Bedruz, A. R. Quiros, A. Bandala and E. P. Dadios, "Machine Vision for Traffic Violation Detection System through Genetic Algorithm," in *8th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM)*, Cebu, 2015.

[7] Y. Du, C. Liu and W. Shi, "A Real-time Vehicle Recognition Method based on Video Sequence Images," in *The Ninth International Conference on Electronic Measurement & Instruments*, Beijing, 2009.

[8] S. A. Rajab and H. H. Refai, "A single element piezoelectric sensor for vehicle classification using the ratio of track width to length," in *IEEE Intelligent Vehicles Symposium* , Michigan, 2014.

[9] C. Anagnostopoulos, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 7, no. 3, pp. 377-389, 2006.

[10] Parliamentary Office of Science and Technology, "Intelligent Transport Systems," *PostNote,* pp. 1-4, January 2009.

[11] B. Dabu, "There's just too many cars on EDSA," GMA News Authority, 7 September 2015. [Online]. Available: http://www.gmanetwork.com/news/story/535945/news/metro/there-s-just-too-many-cars-on-edsa-mmda. [Accessed August 2016].

[12] Z.-h. Su, W.-h. Zhou and W. Zhang, "Vehicle detecting in traffic scenes with introduction of subtractive clustering algorithm," in *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Hangzhou, 2010.

[13] C. Rafael and Gonzalez, Digital Image Processing Sencond Edition, Beijing: Publishing House of Electronics Industry, 2002.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on System Man and Cybemetic,* vol. 9, no. 1, pp. 62-66, 1979.

[15] W. Z. Kong and S. A. ZHU, "Multi-face detection based on downsampling and modified subtractive clustering for color images," *Journal of Zhejiang University SCIENCE A,* vol. 8, no. 1, pp. 72-78, 2007.

[16] J. Gatc and F. Maspiyanti, "Red blood cell and white blood cell classification using double thresholding and BLOB analysis," in *Information and Communication Technology (ICoICT)*, Bandung, 2016.

[17] H. Zhou, L. Llewellyn, L. Wei, D. Creighton and S. Nahavandi, "Marine Object Detection Using Background Modelling and Blob Analysis," in *Systems, Man, and Cybernetics (SMC)*, Budapest, 2015.

[18] A. Chantakamo and M. Ketcham, "The multi vehicle recognition using hybrid blob analysis and feature-based," in *Information Technology and Electrical Engineering (ICITEE)*, Ladkrabang, 2015.

[19] D. Li, B. Liang and W. Zhang, "Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV," in *Information Science and Technology (ICIST), 2014 4th IEEE International Conference*, Guangdong, 2014.

[20] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 25, no. 5, pp. 564-577, 2003.

[21] M. Shah, "Lecture 11: Mean Shift," UCF Computer Vision Video Lectures 2012, 9 October 2012. [Online]. Available: https://www.youtube.com/watch?v=M8B3RZVqgOo. [Accessed 31 August 2016].

[22] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartographys," *Commun, ACM,* vol. 24, pp. 381-395, 1981.