Autonomous Vehicle Identification

Peter Pham, Juan Carlos Franco, Christopher C. Davis

Abstract

The goal of a real time autonomous surveillance system is to be able to identify interesting events (moving objects, abandoned objects, anomalous behavior, criminal acts, etc…) automatically and quickly without the need for a human operator. A complete autonomous surveillance system including cameras, wireless image transport, and "event detection" algorithms has already been developed by the Maryland Optics Group and the Center for the Networking of Infrastructure Sensors at the University of Maryland. This project has involved adding further functionality to the system by creating a program that can identify different car types in real time using Labview. Cropped screenshots of various car models were manually captured, then categorized and stored in a template database. This database is necessary to perform cross-correlation to identify vehicles found from a live camera. Experiments have shown a high success rate in identifying sedans and vans.

1. Introduction

In an autonomous video surveillance system, detailed images are necessary to perform meaningful event detection. With the advancement in technology, such as high definition cameras and high speed communication networks, it is possible to create an autonomous real-time analysis surveillance system that may replace human operators. An autonomous surveillance system could perform more consistently in a fast paced environment than a human operator since humans are vulnerable to fatigue and distractions.

Event detection algorithms in video surveillance systems have to be fast and efficient so that it can operate in real-time. Real-time analysis has an advantage over post analysis because it can detect possible crime or suspicious activities as it is happening before it becomes too late for law enforcement personnel to react.

There is already an autonomous surveillance system developed by the Maryland Optics Group and the Center for Networking of Infrastructure Sensors at the University of Maryland. This system includes of a high definition camera capable of transmitting 1080i/p resolutions and high-speed free space optical (FSO) communication links that can deliver up to 1.25 Gb/s. Using the available equipments, the goal of this project is to create a program in Labview that can identify different car types for surveillance purposes.

2. Methods

Object Detection

The first step in video surveillance is to detect moving objects. Background modeling is a core component in object and motion detection applications. There are many approaches to background modeling including building background based on average images of a scene and statistical modeling. These methods vary on complexity and offer different advantages, for example, background modeled by mixture Guassians are better suitable for outdoors scene.

The background modeling scheme used in this project is background averaging. This method creates an estimation of a scene with no moving object by averaging a series of images over a time interval. The background is adaptive and will update whenever there are prolonged changes in a scene. This method works when there are gradual changes in the scene such as lighting, but has its weaknesses since it cannot adapt to fast changes like falling leafs or waving branches. Nevertheless, it is adequate for the scope of this project in detecting large moving vehicles where the region of interest is only on pavement roads.

Motion in a scene can simply be detected by subtracting each newly received frame from the average background. The result of this subtraction is a motion mask consisting of purely white and black pixels. According to a user-defined threshold, a white pixel is assigned when there is a strong contrast between the pixels in the current frame and the background. While black pixels show that there is no noticeable difference. Motion in a scene will create a significant variation in pixel intensity and appears in the motion mask as white blobs. An ideal motion mask showing a moving object should look like a white figure with similar shape and size of the moving object. However, the motion mask in practice will show white figures with tiny holes and non-uniformed curves due to noises and blur from the camera. Morphology techniques such as dilation is used to help filter out the tiny holes so that the objects in the motion mask are more solid as shown in Figure 1.

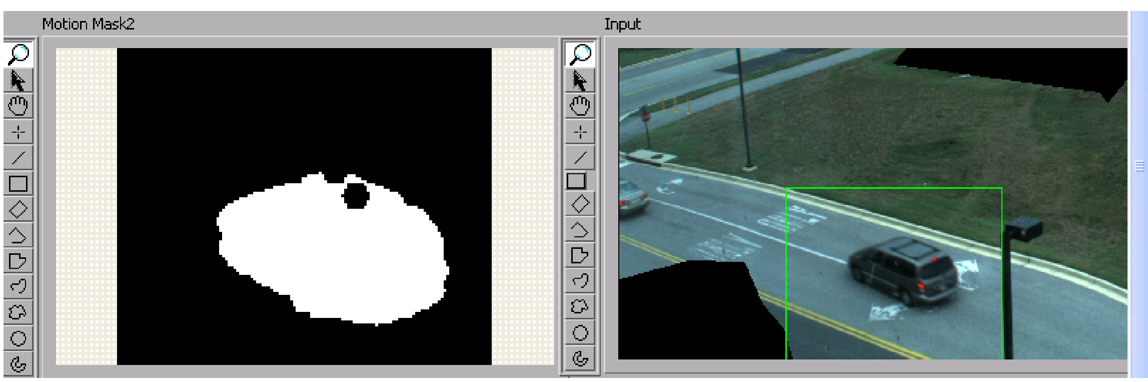

Figure 1. The moving van produces a white figure in the motion mask. The green box on the right is the user-defined region of interest.

Pattern Recognition

The next step after motion detection is to perform an image analysis to determine the shape of the vehicle. The digital camera in this setup is positioned at angle where the received car images can show both the side and backend of the vehicle as seen in Figure 1. Vehicles are categorized, for the sake of simplicity, in four categories: Sedan, truck, van and SUV. Features that could generally help to distinguish between the vehicle types are the back and side windows in a sedan, the triangular shape nose in a van, the rectangular bed in a truck, and the side doors in a SUV.

Experiments have shown that edge detection alone cannot determine the shape of the vehicle. Distinct and clear edge information is not possible due to the noises and blurring in the images. But a combination of edge and cross-correlation could be the solution for vehicle identification.

Labview Vision has a built in pattern recognition module where it uses a combination of edge and cross-correlation algorithm to compare images. This algorithm uses the following equation to perform the normalized cross-correlation:

$$(1) \quad R(i,j)=\frac{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(w(x,y)-\overline{w}\right)\left(f(x+i,y+j)-\overline{f}(i,j)\right)}{\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(w(x,y)-\overline{w}\right)^2\right]^{1/2}\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(f(x+i,y+j)-\overline{f}(i,j)\right)^2\right]^{1/2}}$$

i=0,1,2,3, …M-1
j=0,1,2,3, …N-1

Where w is the template image with dimension K*L, f is the target image with dimension M*N, $\overline{w}$ is the average intensity of the template, and $\overline{f}$ is the average intensity of the target image.

Cross-correlation algorithm finds the best location that matches gray-scale similarities between a template image and a target image. The basic principle for cross-correlation is to shift the template image around the target image and sum the entire pixel intensity products between where the template and target image overlap. The location where R(i,j) peaks is where the template image best matches in the target image. A normalized correlation factor in the denominator of equation 1 is calculated so that this process can overcome sensitivity to high pixel intensity changes in the template and target image.

The cross-correlation algorithm is complex, thus intelligent sampling and an offline template learning process is used to help reduce computational time. Execution time is very important in real-time video surveillance system since it directly determines the maximum number of objects that can be analyzed per second.

A target image usually contains redundant information so it is unnecessary to perform this time intensive correlation over the entire target image. The pattern recognition algorithm intelligently uses edge and region pixels to create reference coordinates where the best possible location match for a template may occur. Edge pixels are found by graphing a histogram of grayscale intensity versus a spatial dimension. A high contrast in the grayscale intensity indicates the location of an edge. Region pixels are found when the intensity of a reference pixel matches with its neighboring pixels.

Also, the template images is processed offline so that the average pixel intensity, $\overline{w}$, and locations of edge and region pixels can be found before the real-time correlation algorithm is executed.

Template Database

In order for the pattern recognition to function properly, the car images in the template database have to be taken from the same angle as the images received from the live camera. Therefore, the easiest way to build this template database is to manually collect car images from the live camera itself. A good database would be a large collection of car images that best represents many of the general car shapes. Also, an edge detection test was used to make sure that the collected template images contained enough details for the proper use of the pattern recognition algorithm.

Through numerous trials, a total of 69 car models were collected to use as templates after passing the edge detection tests. Some of the sedan templates images are shown in Figure 2.

Figure 2. Images of four sedan templates

Timing for the Execution of Pattern Recognition

It would be very time consuming to execute the pattern recognition algorithm for every frame received from the camera due to its complexity. This would hinder the performance of the program and reduce the number of vehicles that can be identified per second. In order for the program to be fast for real time-analysis, the pattern recognition algorithm should only execute under specific conditions.

Since the goal of this project is vehicle identification, it follows that two conditions must be met before the execution of the pattern recognition algorithm. 1) The object in motion is a vehicle 2) The vehicle is in the area of interest. The area of interest in this case is a rectangular region predetermined by the orientation of the camera. This specific region in the frame is where the best detail images of a vehicle can be captured. On occasion, there are other objects that will appear in this area of interest besides vehicles such as pedestrians or wild ducks (a common occurrence at UMD). Therefore, the size of the object must also be taken in consideration to isolate vehicles from other objects.

An approach to determine if the object lies in the area of interest is by calculating its center of mass from the motion mask. The coordinates of the center of mass can be simply calculated by assuming that the object in the motion mask contains a uniform density. In the case where the object is not convex then the center of mass may be located outside the object. Because the area of interest is much larger than the object size, this is not a problem as long as the calculated center of mass is close enough to the object. Also, the object size may also be calculated by summing the number of white pixels in the motion mask. An AND logic operation is applied to the Boolean outcome of these two conditions to trigger the execution of the pattern recognition algorithm.

Output Format

The final output of the program is an image of an identified vehicle. This image is stored in the computer hard drive and categorized appropriately in their respective folder. To help monitor the performance of the program, a match score and the closest matched template is overlaid on the top left of the identified vehicle image as shown in Figure 3.

Figure 3. An identified van is given a match score and the top left shows the closest match template. The green cross shows the location where the template best matches in the image.

3. Results and Discussion

A program to test the reliability of Labview Vision's pattern recognition was created. Several car images were tested with a single car template. The car image that was the exact copy of the car template produced a perfect score of 1000. Similar but not exact copies of the car template receive considerably high scores in the 700s or more. Meanwhile, other types of vehicles that were different from the template received lower scores in the low 500s. The results of this test show that a threshold value of around 710 should be used for positive vehicle identification when using the pattern recognition algorithm.

Since the orientation of the camera is at an angle, the captured template images will contain unwanted background (white and yellow lanes) information as seen in the car on the right in Figure 4. Additional tests were conducted to see how the background would affect the match score. These tests consist of matching two exactly similar cars except the target image would have the lanes in the background removed. The results from several tests indicate that there is about 3 to 9 percent variation from a perfect score. This means that vehicle templates images can easily be made by just cropping car images received from the camera without the need to remove background information.

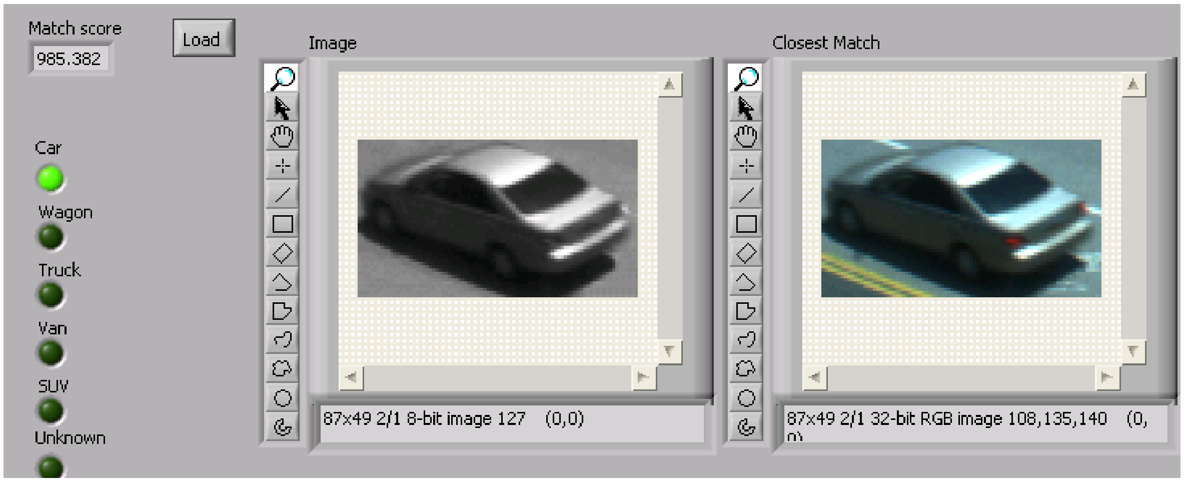

Figure 4. A test run of the pattern recognition algorithm in Labview testing two similar cars with different background. The yellow and white lanes in the background of the template image have insignificant effect on the match score.

Additional tests were needed to determine the relationship between number of templates and processing time for the pattern recognition algorithm and also the percent of positive identification. The first test ran using 31 template images over the course of 2.5 hours and their results are monitored on the computer screen. Similarly, the second test ran using 69 template images also over the course of 2.5 hours. The results of these tests are shown in Figure 6.

31 templates 69 templates

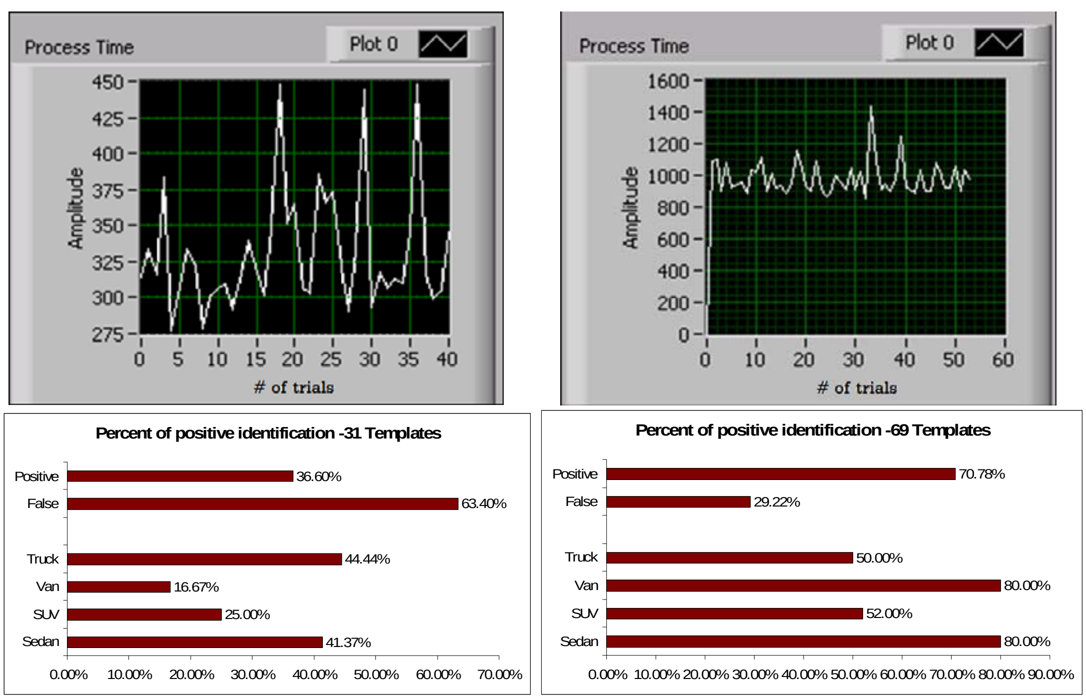

Figure 6. Graphical results of processing time and positive identification for 31 templates and 69 templates.

The results show that the increase in number of template images also significantly increases processing time. The average processing time was increased from an average of 350 ms to 1000 ms. But the tradeoff is that the percentage of positive identification also increases from 36.6 percent to 70.78 percent. While an overall score of 70.78 percent is a lot higher, it is still not good enough for practical use. The positive identification of sedans and vans are high, with scores in the 80th percentile with 69 templates. But the overall results are not as high because of the poor positive identification in trucks and SUVs which were around 50 percent. The low positive identification percentages in trucks and SUVs are because trucks with canopies and SUVs have very similar shape so the pattern recognition algorithm cannot clearly distinguish between the two different vehicles.

4. Conclusion and Suggestions

This paper introduces the basic components required for autonomous vehicle identification. These components include object detection, pattern recognition, timing for execution, and output format. The final results show potential for autonomous vehicle identification, however additional improvements should be made before it is ready for practical use. There are some suggestions to increase the percentage of positive identification. One way is to increase the number of car models in the template database at the cost of processing time. The other alternative to increase both accuracy and speed is to perform a targeted search. This can be done by searching for one type of vehicle at a time which effectively increases the number templates available to each vehicle type given a constant template database size. Also, the area of a vehicle could also be used as a feature to help with the identification process, for example, vans and SUVS will have a larger area than sedans. This could not be done with the current orientation of the camera since the vehicle’s area would change as it moves away from the camera. A solution is to mount the camera perpendicular to the motion of the vehicles so that the area of the vehicle can be constant as it is moving.

Bibliography

1. Elgammmal, A, & Harwood, D. *Non-parametric Model for Background Subtraction.* College Park, MD: University of Maryland.

2. Stauffer, C, & Grimson, W. *Adaptive background mixture models for real-time tracking*. Cambridge, MA: Massachusetts Institute of Technology.

3. NI Vision Concept Manual, November 2005.

4. Retrieved August 4, 2007. *Fast Normalized Cross-Correlation*. Web site: http://www.idiom.com/~zilla/Work/nvisionInterface/nip.html