TRAFFIC SURVEILLANCE VIDEO MANAGEMENT SYSTEM

K.Ganesan*, Kavitha.C, Kriti Tandon, Lakshmipriya.R
TIFAC-Centre of Relevance and Excellence in Automotive Infotronics*,
School of Information Technology and Engineering
VIT University, Vellore – 632 014, Tamilnadu, India
kganesan@vit.ac.in*

ABSTRACT

Now-a-days, the application of the lane surveillance system increases quickly, for example, it is widely settled at the entrance of highways and toll booths to strengthen the traffic management and survey the public security. Accordingly, the size of the captured video data increases over time. So, it is quite important to study how to manage this video data and how to extract useful information from this large amount of video data. In order to process huge volume of data efficiently in traffic video surveillance system, it is very necessary and important to build a highly efficient video management system which uses advanced video compression techniques. This paper presents an improved video management system for lane supervision, which aims at maintaining the traffic surveillance video by analyzing and storing it in such a compressed form that it reduces a considerable amount of storage space and in the mean time also keeps all the information required for the surveillance purposes.

KEYWORDS

Video Compression, Key Frame Selection, Vehicle license plate locating, Reduction of frames, Intraframe compression.

1. INTRODUCTION

As the digital video processing technology is developing, more and more lane surveillance systems that record mass visual information of vehicular traffic lane appear, which play an important role in lane security and management. A lane video of two seconds duration is observed to have a size of 1.5 MB (app.) at the rate of 15 frames per second. Hence 24 hours video will have a size of 64800 MB (approximately 63.3 GB). This is evidently huge and will occupy enormous storage space. It is necessary to compress this video before storing it. From analyzing the lane surveillance video, we notice that all the frames in the video are not necessary to be stored. Since the frames in which there are no vehicles are just still background frames and are not essential for any surveillance purposes, they can be removed from the video which consequently reduces the size of the storage of the video. This is not done by any video compression software which adds more interest to the topic.

The problems in existing lane surveillance systems are: Conflict between size of recorded video data and efficiency of video retrieval i.e., as the size of video increases, the efficiency of retrieval decreases. Another problem is balancing the compression ratio and the quality of the

DOI : 10.5121/ijma.2010.2403

image compressed. General video compression techniques which adopt uniform compression method to the whole image and as a result, much important information for the lane surveillance systems such as the vehicle- license plate will not be clear enough.

Under the conditions of current techniques of video processing, there are only two choices: either to decrease the compression ratio for higher quality video or to elevate the compression ratio for reducing the size of video data recorded in the memory device. Accordingly, we can only get two results: we either keep much shot including redundant information, or lost much key information for the lane surveillance system. Hence, it is not quite adaptable to the lane supervision to use conventional techniques of video compression in which MPEG4 or H.264 video coding standard is usually adopted.

In order to overcome the problems described above, this paper uses an improved video management system for traffic surveillance video. The system we propose here mainly consists of these steps: Video Acquisition, Reduction of frames, License plate locating, Key frame selection, and Intraframe compression. Fig1 shows the whole process as a flow.

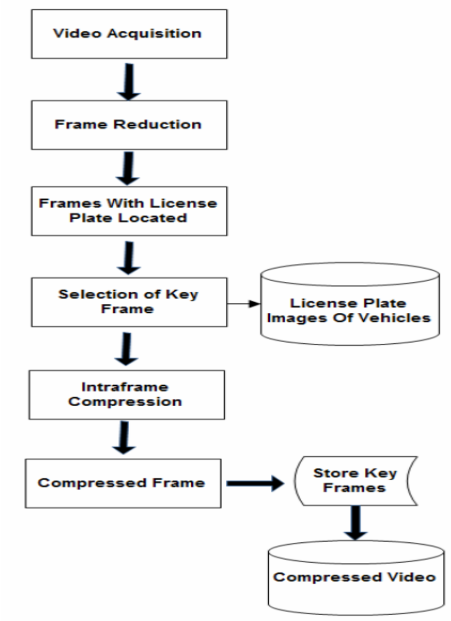


Figure 1. Process Flow Chart

2. VIDEO ACQUISITION

In the lane surveillance system, the camera is placed at certain place, focusing on one or two lanes. So the background in the video sequence is almost fixed too, the foreground object is the moving vehicle which arrives or leaves in vertical direction in each frame.

3. REDUCTION OF FRAMES

The impression of motion is illusory. A video sequence is found to be composed of a number of frames taken with a small time interval in between each. In this paper we propose to reduce the number of frames by means of vehicle detection. Current approaches for object detection use subtraction techniques [3] or edge detection techniques. In our paper we use the following algorithm [1] for the same.

3.1 Algorithm

- Read the frames of the video in pairs.
- Apply discrete wavelet transform to both the frames which divides them into low and high frequency bands.
- For the first frame-
    - Make all the high frequency coefficients as zero.
    - Apply inverse wavelet transform to all the four bands with new coefficients.
- For the next frame- Repeat the above step.
- Calculate the difference between the above resulting frames to get an image having low frequency coefficients alone.
- For the first frame-
    - Make all the low frequency coefficients as zero.
    - Apply inverse wavelet transform to all the four bands with new coefficients.
- For the next frame, repeat the above step.
- Calculate the difference between the above resulting frames to get an image having high frequency coefficients alone.
- Find the gray thresholds of the low and the high frequency coefficients.
- Convert the threshold images to black and white.
- Perform logical AND operation on these two black and white images to obtain the foreground moving vehicles.
- Fill the small holes and dilate the image.

After the vehicles have been detected, eliminating the frames which do not have any vehicle can reduce the number of frames to be processed for license plate locating. Consider all the frames which have been binarized and have the vehicles detected. For each of them: The nearer the vehicle, the clearer the license plate and hence it is enough to consider only 60% of the frame from the bottom. Calculate the total number of white pixels in this area. The average number of white pixels for the bottom 60% of all the frames is calculated. The frames in which the white pixels count is more than 1.5 times the average are stored as the important frames.

4. LICENSE PLATE LOCATING

License plate location is an important stage in vehicle license plate recognition. License plate area contains rich edge and texture information. License plate recognition becomes a key technique to many automated transport system such as road traffic monitoring, automatic payment of tolls on highways or bridges and parking lots access control. The algorithm proposed in this paper for license plate location contains four parts: Image enhancement, Vertical edge extraction, Background curve & noise removing and Plate search & segmentation.

The license plate of the car consists of several characters, so the plate area contains rich edge information. But some times the background of the car image holds much edge information too. One has to consider two facts: one is that the background areas around the license plate mainly include some horizontal edges whereas the edges in the background are mainly long curves and random noise. Second is that the edges in the plate area cluster together and produce intense texture feature. If only the vertical edges are extracted from the car image and most of the background edges are removed, the plate area will be isolated out distinctly in the whole edge image [2]. Thus we propose to locate the license four stages given above.

4.1 Image Enhancement

In the car images captured in the gloomy days or dim nights, the gradients in the license plate area are much lower than those in the contour areas of the car, which is caused by the car shadow in the dazzling sunshine. This will often bring out weak gradients in plate areas too. A few vertical edges will appear in the plate areas, if we extract edge images directly from these car images. Therefore it is important to enhance the car images. Fig.2 shows the original car image and Fig.3 shows the enhanced car image.

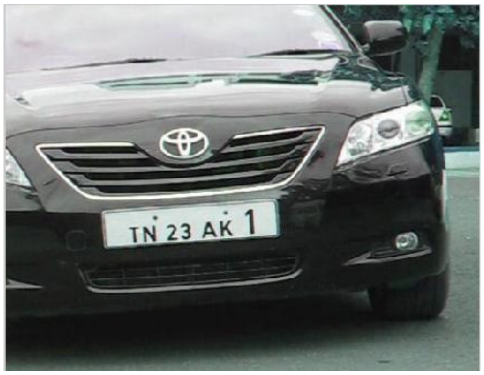


Figure 2. Original car image

We use the following formula in Equation (1) to enhance the image.

$$I'_{i,j} = f(\sigma_{W_{i,j}}) \cdot (I_{i,j} - \bar{I}_{W_{i,j}}) + \bar{I}_{W_{i,j}} \quad \rightarrow (1)$$

where $I_{i,j}$ denotes the luminance of the pixel $P_{i,j}$, $I'_{i,j}$ denote the luminance in the enhanced image, $W_{i,j}$ is a window centered on pixel $P_{i,j}$, $\grave{I}_{Wi,j}$ and $\sigma_{Wi,j}$ are the mean luminance and standard deviation of the pixels in the window $W_{i,j}$, $f(\sigma_{Wi,j})$ is an enhancement coefficient.

$$f(\sigma_{W_{i,j}}) = \begin{cases} \dfrac{3}{\dfrac{2}{400}(\sigma_{W_{i,j}} - 20)^2 + 1} & \text{if } 0 \leqslant \sigma_{W_{i,j}} < 20. \\ \dfrac{3}{\dfrac{2}{1600}(\sigma_{W_{i,j}} - 20)^2 + 1} & \text{if } 20 \leqslant \sigma_{W_{i,j}} < 60. \\ 1 & \text{if } \sigma_{W_{i,j}} \geqslant 60. \end{cases} \quad \rightarrow (2)$$

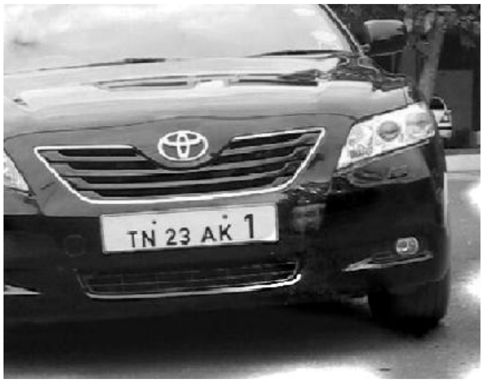


Figure 3. Enhanced image

4.2 Vertical Edge Extraction

We select the vertical Sobel operator as given in equation 3 below to detect the vertical edges, because the simple operator costs us a little computational time. Fig.4 shows the vertical edge image.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \rightarrow (3)$$

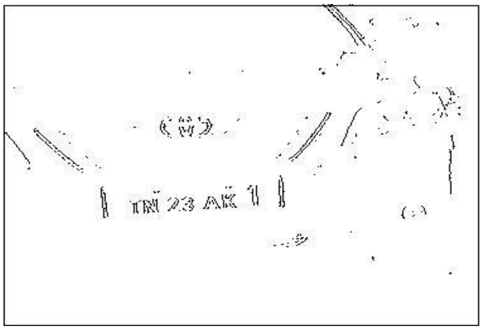


Figure 4. Vertical edge image

4.3 Background Curve and Noise Removing

From Fig.4, we can see that there are many long background curves and short random noisy edges in the vertical edge image besides the license plate edges. These background and noisy edges may interfere in the license plate location. We have proposed a simple algorithm to remove them from the edge image. This algorithm only requires us to scan the edge image for three times. The first scan will record the edge lengths away from the top (or left) start points. And the second scan will record the edge lengths away from the bottom (or right) end points. And the last scan will add up the two kinds of lengths to denote the actual edge lengths; if the edge point has a very long (background curve) or very short (noisy edge) actual edge length, then the edge point will be removed from the edge image.

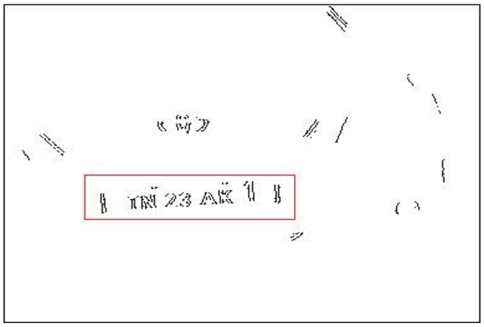


Figure 5. Image after removing background curves and noise.

4.4 Plate search and segmentation

After non-plate edges have been removed considerably, license plate location becomes much easier. We can shift a rectangle window whose size is just bigger than that of the license plate from left-to-right and top-to-bottom in the edge image. Count the total number of the edge points in the window. If the number is above a certain percentage of the area of the window, there may be a license plate in the corresponding window. Fig.6 shows the located license plate.

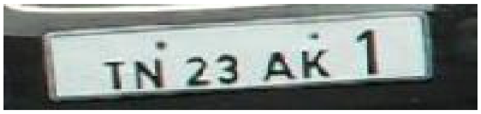


Figure 6. Located license plate

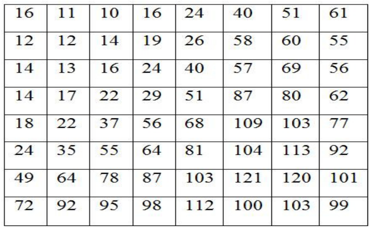
5. KEY FRAME SELECTION BASED ON LICENSE PLATE LOCATING

For analyzing the character of lane surveillance video, this paper proposes the special algorithm for key frame selection. The key frame, which is usually used to describe the main content of a shot, may include one or more frame of image [5]. In this paper, the key frame is the very image that contains the most information of the vehicle. The vehicle license plate lies in the down side of the vehicle. So the nearer the vehicle license plate is to the bottom of the image, the more information about the vehicle can be captured. After we obtain the location of the vehicle license plate in every frame, we select the frame in which the location of the vehicle license plate is nearest to the bottom as the key frame.

6. INTRAFRAME COMPRESSION

The license plate images hence obtained can be compressed further by exploiting the spatial redundancy between the pixels. For this, the image is first “level shifted” by $2^{(p-1)}$, i.e. we subtract $2^{(p-1)}$ from each pixel value where p is the number of bits used to represent each pixel. For example, if we are dealing with 8 bit images whose pixels take on values between 0 and 255, we would subtract 128 from each pixel so that the value of the pixel varies between -128 to 127. The image is then divided into blocks of size 8X8, which are then transformed using an

8X8 forward DCT. This means we take the DCT of this level shifted block in order to obtain DCT coefficients. The lower frequency coefficients in the top left corner of the table have larger values than the higher-frequency coefficients. Each DCT coefficient is quantized to reduce its precision. The quantizer step size can be different for different coefficients in a block. The quantization step size table that we have used is shown below in Fig. 7.



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 7. Quantization table

The quantized blocks are then zigzag coded. This results in every block having a certain number of coefficients followed by a continuous run of zeroes. These values are further runlength coded and stored in a 2-column vector with the first column containing the coefficients in the zigzag coded order and the second containing their corresponding count. The runlength encoded data is finally Huffman coded and stored in a binary file. The compressed data can be decoded to obtain the original license plate image.

7. EXPERIMENTAL RESULTS AND ANALYSIS

To illustrate the effectiveness of the proposed system, a video of duration 2 seconds which has one car in it is captured. It is noted to have a size of 1.50 MB and 41 frames. After applying compression, we obtain the license plate of the car in the video which is of the size 3 KB. This license plate image alone is saved by the system automatically. Hence, the compression percentage is 99.89%.

Table 1 shows the comparison between MPEG compression and the compression done with the proposed system for two samples.

Original Video size	MPEG Compression	Compression by the proposed system
1.5 MB	0.78 MB	3 KB
318 KB	167 KB	2.85 KB

Table 1. MPEG Vs Proposed System

If the number of vehicles in the video increases, the compression percentage will decrease accordingly. For example, if there is one more car in the same video, two license plates will be stored. Let us say the other car's license plate has a size of 2.8 KB. Then the compression percentage will be 99.62%. The algorithm that the proposed system uses for license plate location is observed to detect the license plates for 98% of the samples.

8. CONCLUSION

We have proposed a system exclusively for the compression of lane surveillance videos by making use of the concepts of vehicle detection and license plate location. We first reduce the number of frames in the lane video by using vehicle detection to eliminate the background frames and also those which do not carry clear license plate information. License plates are located for the reduced number of frames and the frames in which the license plates are clearest for every vehicle are marked as key frames. From these frames, the license plate images are extracted. The license plate images hence obtained are compressed using intraframe compression and stored. The system is robust against most of the conditions and shows considerably high compression results as compared to existing video compression techniques.

ACKNOWLEDGEMENT

This work forms part of the R&D activities of the TIFAC Centre of Relevance and Excellence in Automotive Infotronics located at VIT University, Vellore, India. The authors would like to thank the TIFAC- CORE for providing the necessary hardware and software facilities to carry out this work.

REFERENCES

[1] Budi Sugandi, Hyoungseop Kim, Joo Kooi Tan, Seiji Ishikawa, “Real Time Object Tracking and Identification Using a Camera”, The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008), Fukuoka, Japan, pp. 1585-1588, 2008.

[2] Danian Zheng, Yannan Zhao, Jiaxin Wang, “An Efficient Method of License Plate Location”**,** Pattern Recognition Letters, pp. 2431-2438, 2005.

[3] David Strachan, “Video Compression”, SMPTE Journal, pp. 68-73, February 1996.

[4] Resha Hashemian, “Direct Huffman Coding and Decoding Using the Table of Code-Lengths”, Proceedings of the International Conference on Information Technology: Computers and Communicatios (ITCC’03), Illinois, USA, 2003.

[5] W.B Mikhael and A.Ramaswamy, “Application of Multitransforms for Lossy Image Representation”, IEEE Transactions on Circuit and Systems-II, Vol. 41, No: 6, June 1994.

[6] Yu Shen, Cheng and Gen Dow Huang, “Audio/Video Compression Applications Using Wavelets”, pp.2214-2218, 2002(Private Communication)

[7] ZHAO Shu-long, YOU Zhi-sheng, LAN Shi-yong, ZHOU Xin, “An Improved Video Compression Algorithm for Lane Surveillance”**,** Fourth International Conference on Image and Graphics, Chendu, China, pp. 224-229, 2007.

Authors

K. Ganesan obtained his Ph.D from Bharathidasan University, Tiruchirapalli, India in 1993. Then he worked as a Post Doctoral Fellow at Queen's University of Belfast, United Kingdom for 3.25 years. He was heading the Computer Science and Engineering Department at Vellore Institute of Technology, Vellore, India during 2002-2005. Currently he is the Director of TIFAC Centre of Relevance and Excellence in Automotive Infotronics located at VIT University, Vellore, India. He is guiding 5 Ph.Ds. He has visited more than 15 countries abroad. He has got more than 50 journal and International conference publications. He has recently filed a patent. His areas of interest include Image and Video processing, Data security, Wireless and embedded systems, Mobile computing. His profile has been included in the 9th and 10th anniversary edition of Marquis Who's Who in Science and Engineering. He has been identified as one of the Top 100 Scientists 2008 by International Biographical Centre, Cambridge, England.

Kriti Tandon is the final year students of M.S (Software Engineering) studying at VIT University in the School of Information Technology and Engineering,Vellore,India.

Kavitha C is the final year students of M.S (Software Engineering) studying at VIT University in the School of Information Technology and Engineering,Vellore,India.

Kavitha C is the final year students of M.S (Software Engineering) studying at VIT University in the School of Information Technology and Engineering,Vellore,India