Video-based License Plate Reader

Qing Li
SCPD student, Stanford University
qlicub@gmail.com

***Abstract*— This paper proposes an algorithm that automatically reads license plate from videos that has taken from personal devices. Unlike commonly used license plate reader in toll payment or road control system, video from personal devices usually has more variation in viewing angle, resolution, shutter speed and so on. This work read single frame from video first, then extract the license plate by detecting high-density vertical edge areas and filtered by color and boundary features. Then the pre-processing step automatically correct the rotation of the image, remove the background noise and segment the character portion of the license plate. Lastly, this algorithm recognizes plate with template matching and filters the peaks by peak intensity and coordinates. Based on these, videos with multiple frames can improve the overall accuracy and reliability.**

Keywords*— *Edge detection; Hough transform; Morphological image processing; template matching

Automatic license plate reader, also known as Automatic number plate recognition (ANPR), was first invented in 1976 in at the Police Scientific Development Branch in the UK and are widely used among police forces worldwide today as it is essential for numerous real-life applications, such as traffic control, automatic toll collection and road traffic monitoring[1]. These systems usually require special hardware: one or more road-rule enforcement cameras, a single camera with assistant IR illumination, or closed-circuit television are often used.

On the software side, ANPR typically detect the location of the license plate in an image first, pre-process the plate image by orientation and sizing correction, and segment the useful information portion of the license plate, finally it uses optical character recognition (OCR) to read the plate. Additional processing to check characters and positions to classify license plate and link license plate information to some database are often needed depending on the applications.

The variation of the plate types and environmental differences increases the complexity of both the detection and recognitions part [2]. The plate variation include: 1) plate location including both plate image locates at different position of an image, and also the numerical information locates at different place of the license plate; 2) language and character font, special characters, total number of characters; 3) Color, includes both character and background color and pattern; 4) special plate such as for disabled people and special characters and so on. Fig. 1 shows that license plates in California has different color, location, fond from the plate from Colorado; even within California, the background/foreground color, background pattern, special characters, and special-use plates all exist. Also variety of plate frames that are commonly used on the car can also cause additional noise to the recognition results.

Fig. 1. Variety of license plates.

There are also imaging variations when pictures or videos are taken, these includes illumination (day, night, additional illumination beam etc.), similar pattern as background noise, camera resolution, camera shutter speed, motion blur (both from camera and from car), viewing angle, number of cameras and so on.

The accuracy and reliability of the ANPR system is always a key part of the algorithm, thus much effort has been made to increase the ALPR reliability and accurate nowadays [3-4]. Recently, ALPR also gets interest beyond the police forces as personal cameras are more available and the computing hardware gets cheaper and cheaper.

My project here is to detect and recognize the California license plate by analyzing videos from personal devices, such as cell phones, camera carried by hobby drones, which usually has low resolution or speed limit. In this case, it introduces more complexity and challenges (Fig. 2) as the personal devices usually does not have fixed viewing angle or field of view as the road cameras, they may suffer from the low resolution, slower speed and limited computing power, all results in complexity of the algorithm and longer computing time.

Fig 2. Variety of viewing angle, extreme illumination condition and resolution.

I. Method and Results

From the video data, first a single frame image is read form the video. It goes through 4 algorithm steps to reads the plates, as shown in Fig.3.

Fig 3. Algorithm flow chart (top) and Single frame example read from a video (bottom).

A. License Plate Extraction

First step is to detect the location of a license plate and extract the sub-image that includes the license plate for further processing steps. This removes most of the background noise and enables faster processing speed. This stage influences the accuracy of the recognition algorithm.

There have been many different approaches used in previous studies to extract license plate and filter out false results. Four most commonly ones are:

1) *Boundery and Edge information [5]*
2) *Texture Features[6]*
3) *Color features[7]*
4) *Character features [8]*

Since the license plate normally has a rectangular shape and a fixed aspect ratio, we can use edge detection to finding all rectangles and filter it with the known aspect ratio. However, due to the viewing angle, the license plate may shows a perspective effect or distortion from camera, occlusion or frames may all change the ratio; bumpers in the car increase the noise of finding rectangles, so many literatures recommend using vertical edges to find the license plate [3, 9].

Most license plates involve rapid characters and background change, also known as the texture feature of the license plate. This feature results in high edge density areas. Scan-line technique [10], sliding concentric window [11], adaptive boosting(AdaBoost) combining with Haar-like feature [6] and other techniques have been used to detect license plate accordingly.

For many countries, there are specific colors for backgrounds and foregrounds are allowed in the license plate, in that case, color feature can be used for plate extraction. However, defining the pixel color using the RGB value is very difficult, especially in different illumination conditions. Many techniques would convert the RGB color to hue, lightness, and saturation (HLS) or other color models to improve robustness. For California, there is variation in plate color, making it more difficult to use color feature.

Lastly, character feature can also be used to detect license plate. In this case, we can directly read the license characters, especially for some countries where there are special limited characters exist on the license plates that can be used to detect features. Scale-invariant feature transform (SIFT) can be used to detect license plate [13], and maximally stable extremal regions (MSER) can also be used to assist this process [14]. However, these feature detection are time consuming and often requires high quality images (focus and resolution).

In my project I combined the first 3 methods to detect the license plates. I first use Sobel filter for vertical edge detection of the image frame, as shown in Fig 4.

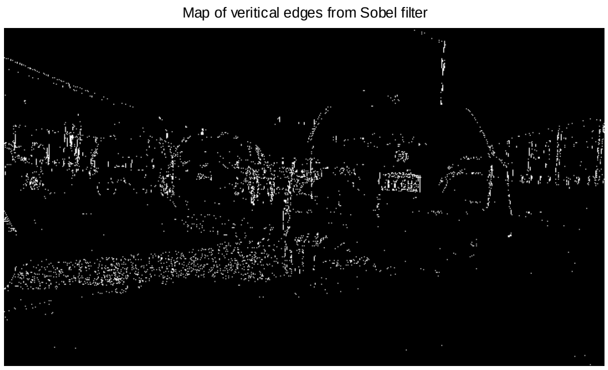

Fig. 4 Map of the vertical edges from Sobel filter.

Then I apply Gaussian filter to blur and find the high-density area in the edge map. The California license plates often involve high contrast of colors between characters and the background, and the texture feature also has rapid color change. Here I take this advantage and applied the same filtering for all 3 RGB channels, and filter out single color edges, such as that from rear light or environmental background, as shown in Fig. 5.

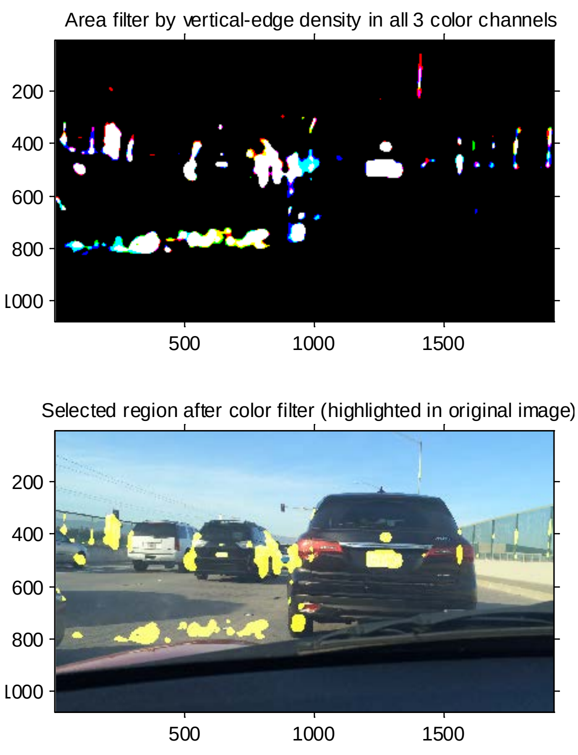

Fig. 5: High density vertical-edge regions (top) and selected regions after color feature filtering (bottom);

Then I use region labeling to label all possible regions. Odd shape regions are filtered out by a threshold of the ratio of region area versus rectangular bounding box area. Lastly I use a fixed aspect ratio of 2:1 with a small variation to select possible regions of license plates, as shown in Fig 6.

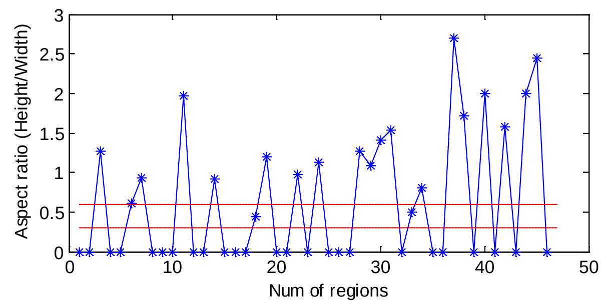

Fig 6. Filter regions from boundary feature in license plate extraction step.

SIFT with character template is also used and evaluated in the extraction step. However, I found it has very high requirement on the image quality as it capture matched feature with high resolution license plate, but no results at low resolution ones. Also SIFT take much longer time. The final algorithm doesn't include SIFT part.

Detected region is highlighted in a yellow box in the original image as part of the final results with labeling of the detected license plate number (Fig 7).

Fig. 7. Detected license plate highlighted in original image

B. License Plate autocorrection

After extract the license plate, I applied a pre-processing algorithm to autocorrect the sub-image in order to remove noise, segment the character region, and enhance the signal.

Given the fact that personal devices may take videos/photos with random angle, the first pre-processing is to correct the rotation of the plate.

I apply horizontal edge detection to generate horizontal edge maps, then apply Hough transform to the center portion of the image and detect Hough peaks to correct the rotation angle to the image.

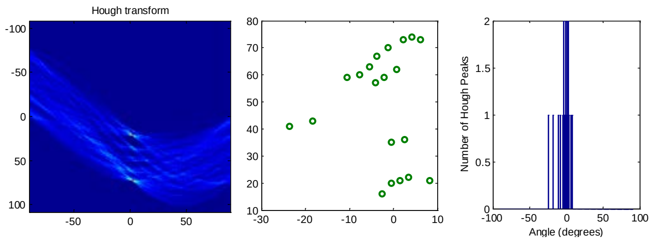

Fig 8. Finding peak from Hough transform to correct plate rotation.

I then apply both horizontal and vertical detection together with the morphological image processing techniques to find the dense edge area and edge peaks so that crop the plate image to an even smaller image that contains mostly the useful information (character) region, as shown in Fig. 9.

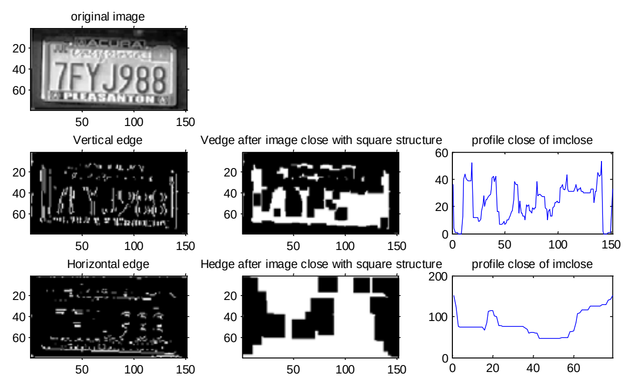

Fig. 9 Vertical and horizontal edge detection with image dilation and erosion.

Several steps are then employed to rotate, cut background, and segment only the character portion. The results of extracted image and that after autocorrection are shown in Fig. 12 together with the result of character recognition. . Please note that the license plate frames may add difficulties in the results, which require some iteration of sizing test in the recognition algorithm.

C. License Plate Recognition

After correct the rotation, sizing, and background noise, I can use the simple and straightforward method, template matching, for license plate recognition as the last step of the single-frame image processing.

There already have been several literatures using the template matching technique for ANPR [5] and encounter difficulties due to font change, rotation, sizing variation and noise, as well as time consuming process. Besides the pre-processing correction, I also limited my project to detect only California license plate, thus only one fixed font is used here. Also, this is not real-time processing, which relaxes the runtime requirement.

A digital recreation of the letters used on California state license plates found online [15] and I manually made a template database with 26 letters plus 10 number and some additional special characters such as heart symbol from an image of all characters. A more official template with higher resolution should improve the performance of the recognition.

Because it is extremely difficult to segment the plate the same way for all plates in all frames from the video, I relaxed the requirement for the pre-processing, and added pre-run of recognition in this step: the segment image was first template matching with 0-9 number and several letter (A, E and I) templates to increase the successful rate. The template is resized to match the input image with flexible scale of 60% to 110%, the signal level at each height ratio is recorded (shown in Fig. 10) and applied for full plate matching.

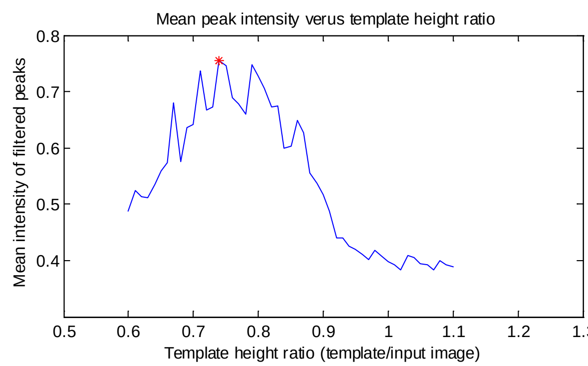

Fig. 10 Intensity of template matching peaks with special character template sets at different image scaling level.

After the scaling ratio is determined, the plate image is running template matching for each character template. I recorded the peaks above 85% of the maximum peak in the result of each template, together with the XY coordinates. An example of image filtering with template "8" is shown in Fig 11, where two peaks have found and highlighted.

After all template matching, the highest peaks are selected based on their X and Y locations, since all the characters in the license plate are in the similar height, and separated in X with nearly fixed width. The signal intensity of the recorded peaks is used when multiple peaks are found at the same character position.

The final results are printed out, and also displayed next to the autocorrection results as part of the final results, two examples are shown in Fig 12.

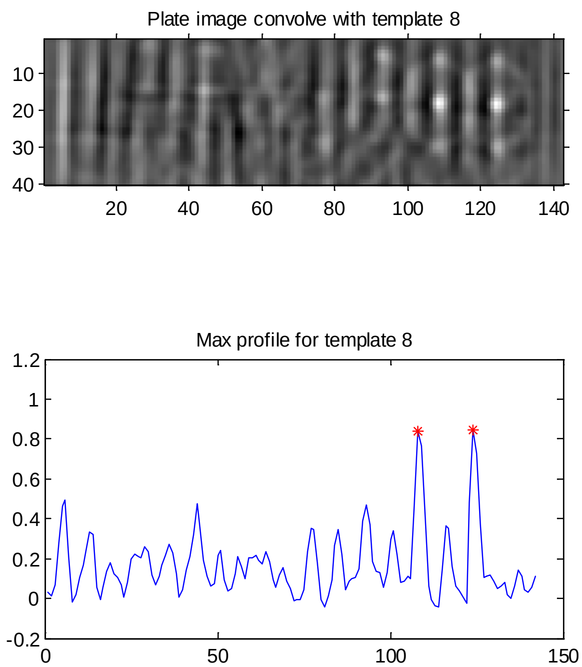

Fig. 11: License plate (7FYJ988) matching with template of "8", two peaks are detected.

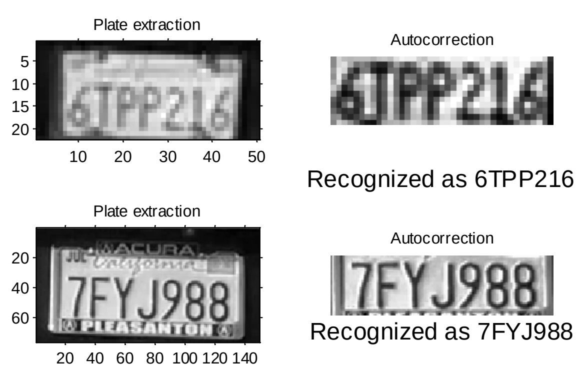

Fig. 12: Final results from the VLPR algorithm, indicating the extracted plate image, and the one after autocorrection. The recognition results are displayed in text.

D. Multi-frames from videos

Videos have the luxury of multiple images from similar scene. Multiple frames can go through the same algorithm to read the same license plate over multiple times, it can help find the best fit since not all images provide the same image quality as the car or camera usually is moving when video is taken. It also helps to remove outliers as compare with multiple frames so that it can enhance the reliability and accuracy. Videos provide easy way to evaluate the algorithm as I will discuss in next session.

Videos can also be used to track the vehicle over time and estimate the license plate motion thus predict the next frame position, this can improve the detection and reduce the processing time.[2] Another unique application from video license plate reader is to reconstruct high-resolution image by combining multiple subpixel shifted, low-resolution images, also known as super-resolution reconstruction.[16]

II. Evaluation

The videos also provide a convenient way to evaluate the algorithm as it automatically provides tens or hundreds of image for testing statistics.

Due to the complexity of the imaging condition especially the illumination, resolution and size of the license plate difference, it's difficult and unfair to evaluate a "good" quality license plate image of bright, clear, high-resolution images with a dark, low-resolution images.

Here I targeted a "good" license plate in a video over 110 frames. The recognition correctly reads 104 frames and has 3 false reading from autocorrection error and 3 false reading due to recognition algorithm, while 100% accurate on extraction (see table 1). The overall ratio of accuracy is about 94.5%. If we consider the false results come from camera stability, environmental noise and so on, which is independent between frames, the successful rate with two frames reading is 99.7%.

The low-quality data has much worse successful rate, in the example of the plate in Fig , the algorithm detected the plate only in 3 frames (as the vehicle get further and license plate gets smaller in later frames), and the detection rate is 0 after 3 frame. Within the first 3 frames, it reads 6TPPZ16 instead of 6TPP216. Further tests on a different license plate in the same movie or another low-resolution plate in a different movie show similar low successful ratio in detection, pre-preprocessing or recognition. The blur of the image significantly affect the extraction rate, and the resolution usually limit the accuracy of the recognition.

This image processing is done offline and not in real-time. The total processing time to read 1 frame is roughly 5s assuming detecting two license plates on my laptop.

TABLE I. EVALUATION OF SUCCESSFUL RATE OVER 110 FRAMES

Results	***Error from***	***Frames***
7FYJ988 (correct reading)	-	101
7FYJ9HH (x2), 7FYJ900	Recognition	3
N.A.	Extraction	0
Segmental wrong	Autocorrection	3

III. SUMMARY

In summary, this project is trying to read license plate for videos that has taken by personal devices, which involves more complexity due to hardware limitation.

In this work, I first read the video into single frame images. Then I extract the license plate part from the image by detecting high-density vertical edge areas and filtered by color feature and boundary conditions. I run pre-processing to correct the rotation of the image, remove the additional noise and background, and segment the character portion of the license plate. Lastly, I use a template matching on small sets to find best sizing scaling factor and then with all 26 letter, 10 numbers and special characters to find the peaks and filter out the false signal by peak intensity and coordinates.

The software successfully reads videos and displays the license plate location in the original image with the recognition results.

The results processing time is roughly 5 second to read single frame of a video. The successful rate of extraction is low for low-quality (resolution, blur etc.) images, the auto-correction and recognition suffers accordingly. For good quality images with high-resolution, bright illumination condition, the successful rate of the license plate is estimated as 94.5%, will ~3% loss on both preprocessing and ~3% loss on recognition, while solid good on extraction. Multiple frames in a video can be used to improve the reliability and accuracy.

REFERENCES

[1] G. Liu, Z. Ma, Z. Du, and C. Wen, “The calculation method of road travel time based on license plate recognition technology,” in Proc. Adv. Inform. Tech. Educ. Commun. Comput. Inform. Sci., vol. 201. 2011, pp.385–389.

[2] Wikipedia: https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_California

[3] Shan Du et al. “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review”, IEEE Transactions on Circuits and Systems for Video Technology, Volume:23 , Issue 2, Page 311, 2012

[4] Shan Du et al. “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review”, IEEE Transactions on Circuits and Systems for Video Technology, Volume:23 , Issue 2, Page 311, 2012

[5] M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, “Saudi Arabian license plate recognition system,” in Proc. Int. Conf. Geom. Model. Graph.,2003, pp. 36–41.

[6] H. Zhang, W. Jia, X. He, and Q. Wu, “Learning-based license plate detection using global and local features,” in Proc. Int. Conf. Pattern Recognit., vol. 2. 2006, pp. 1102–1105.

[7] W. Jia, H. Zhang, X. He, and M. Piccardi, “Mean shift for accurate license plate localization,” in Proc. IEEE Conf. Intell. Transp. Syst., Sep. 2005, pp. 566–571.

[8] J. Matas and K. Zimmermann, “Unconstrained license plate and text localization and recognition,” in Proc. IEEE Conf. Intell. Transp. Syst., Sep. 2005, pp. 572–577.

[9] S. Wang and H. Lee, “Detection and recognition of license plate characters with different appearances,” in Proc. Int. Conf. Intell. Transp.Syst., vol. 2. 2003, pp. 979–984.

[10] Y. S. Soh, B. T. Chun, and H. S. Yoon, “Design of real time vehicle identification system,” in Proc. IEEE Int. Conf. Syst. Man Cybern., vol. 3. Oct. 1994, pp. 2147–2152.

[11] C.-N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, “A license plate-recognition algorithm for intelligent transportation system applications,” IEEE Trans. Intell. Trans. Syst., vol. 7, no. 3, pp. 377–392, Sep. 2006.

[12] X. Shi, W. Zhao, and Y. Shen, “Automatic license plate recognition system based on color image processing,” Lecture Notes Comput. Sci., vol. 3483, pp. 1159–1168, 2005.

[13] W. T. Ho, H. W. Lim, Y. H. Tay, and Q. Binh, “Two-stage license plate detection using gentle Adaboost and SIFT-SVM,” in Proc. 1st Asian Conf. Intell. Inform. Database Syst., 2009, pp. 109–114.

[14] H. W. Lim and Y. H. Tay, “Detection of license plate characters in natural scene with MSER and SIFT unigram classifier,” in Proc. IEEE Conf. Sustainable Utilization Development Eng. Tech., Nov. 2010, pp. 95–98.

[15] Penitentiary Gothic, Identifont website: http://www.identifont.com/similar?2NK .

[16] K. V. Suresh, G. M. Kumar, and A. N. Rajagopalan, “Superresolution of license plates in real traffic videos,” IEEE Trans. Intell. Transp. Syst., vol. 8, no. 2, pp. 321–331, 2007.