ROBUST NUMBER PLATE RECOGNITION IN IMAGE SEQUENCES

Andreas Zweng and Martin Kampel
Institute of Computer Aided Automation, Vienna University of Technology, Pattern Recognition and Image Processing Group, Favoritenstr. 9/1832, A-1040 Vienna, Austria

Keywords: License Plate Localization, License Plate Recognition, Character Classification, Character Segmentation, Image Sequences, Blob Analysis.

Abstract: License plate detection is done in three steps. The localization of the plate, the segmentation of the characters and the classification of the characters are the main steps to classify a license plate. Different algorithms for each of these steps are used depending on the area of usage. Corner detection or edge projection is used to localize the plate. Different algorithms are also available for character segmentation and character classification. A license plate is classified once for each car in images and in video streams, therefore it can happen that the single picture of the car is taken under bad lighting conditions or other bad conditions. In order to improve the recognition rate, it is not necessary to enhance character training or improve the localization and segmentation of the characters. In case of image sequences, temporal information of the existing license plate in consecutive frames can be used for statistical analysis to improve the recognition rate. In this paper an existing approach for a single classification of license plates and a new approach of license plate recognition in image sequences are presented. The motivation of using the information in image sequences and therefore classify one car multiple times is to have a more robust and converging classification where wrong single classifications can be suppressed.

1 INTRODUCTION

Automatic license plate recognition is used for traffic monitoring, parking garage monitoring, detection of stolen cars or any other application where license plates have to be identified. Usually the classification of the license plates is done once, where the images are grabbed from a digital network camera or an analog camera. The resolution of the license plates has to be very high in order to correctly classify the characters. Typically localization of the license plate is the critical problem in the license plate detection process. For localization, candidate finding is used in order to find the region mostly related to a license plate region. If the license plate is not localized correctly for any reason (e.g. traffic signs are located in the background), there is no other chance to detect the license plate when taking only one frame into consideration.

In this paper we propose an approach of automatic license plate detection in video streams when taking consecutive frames into consideration for a statistical analysis of the classified plates in the previous frames. The idea is to use standard methods for the classification part and use the classification of as many frames as possible to exclude classifications where the wrong region instead of the license plate region was found or the classification of the characters was wrong due to heavily illuminated license plates, partly occluded regions or polluted plates. The algorithm has to perform with at least 10 frames per second if the car is visible only for some seconds to retrieve enough classifications for the statistical analysis. To reduce the amount of computation, grayscale images are computed from the input images where it is also not necessary to keep the information of all three color channels for our approach.

The remainder of this paper is organized as follows: In section 2, related work is presented, in section 3 the methodology is reviewed (plate localization, character classification and the classification of the plate). In section 4 experimental results are shown and section 5 concludes this paper. Our approach has the following achievements:

Zweng A. (2009).
ROBUST NUMBER PLATE RECOGNITION IN IMAGE SEQUENCES.
In *Proceedings of the First International Conference on Computer Imaging Theory and Applications*, pages 56-63
DOI: 10.5220/0001801200560063

- High improvement of existing approaches
- Real time classification in image sequences
- Converging progress of classification

2 RELATED WORK

Typically license plate recognition starts by finding the area of the plate. Due to rich corners of license plates Zhong Qin et.al. (2006) uses corner information to detect the location of the plate. Drawbacks of this approach are that the accuracy of the corner detection is responsible for the accuracy of the plate localization and the detection depends on manual set parameters. Xiao-Feng et al. (2008) use edge projection to localize the plate which has drawbacks when the background is complex and has rich edges. For character segmentation Xiao-Feng et al. (2008) made use of vertical edges to detect the separations of characters in the license plate where Feng Yang et.al. (2006) use region growing to detect blobs with a given set of seed points. The main problem of this approach is, that the seed points are chosen by taking the maximum gray-values of the image which can lead to skipped characters due to the fact that no pixel in the characters blob has the maximum value. For character classification, neural networks are used (Anagnostopoulos et al., 2006; Peura and Iivarinen, 1997). Decision trees can also be used to classify characters where each decision stage splits the character set into subsets. In case of a binary decision tree, each character-set is divided into two subsets.

3 METHODOLOGY

The methodology is divided into three main points. The license plate localization algorithms which finds the region of the plate, the character segmentation which extracts the digits on the plate and the license plate classification which classifies the extracted blobs. The license plate classification is separated in three steps, namely the single classification, which is done in each frame, the temporal classification which uses the single classifications and is also done in each frame and the final classification which uses the temporal classifications and is done once per license plate at a triggered moment (e.g.: after 20 classifications; when the plate is not visible anymore). The algorithms in this paper are summarized in the graph shown in fig. 1.

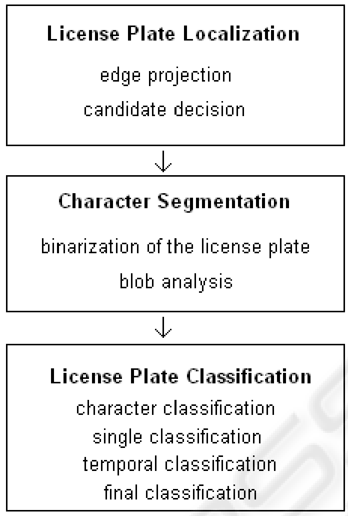

Figure 1: Summary of algorithms.

3.1 License Plate Localization

In video streams license plates are not present in all frames so the first part of the localization is to detect whether a license plate is present or not. Edge projection is used to find plate candidates, which is described in section 3.1.1. To detect if a plate is located in the region of interest in the image, the highest edge projection value is stored for the past frames. In our approach the past 1500 frames are taken to compute the edge projection mean value. If the highest edge projection value in the actual frame is 10% higher than the mean value, further computation in license plate detection is done, otherwise the frame is rejected.

3.1.1 Localization of License Plates

License plates usually have high contrast due to the characters in the plate region. This property can be used to detect license plates localization. The localization process is done in three steps. The first step is to detect vertical positions of possible license plates by projecting vertical edges onto the vertical axis. The second step is to detect the horizontal boundaries of the license plate by projecting horizontal edges onto the horizontal axis (Martinsky, 2005).

$$P_y = \sum_{i=0}^{h-1} I(x,i) \qquad (1)$$

$$P_x = \sum_{i=0}^{w-1} I(i,y) \qquad (2)$$

P_y is the horizontal edge projection of line y and P_x is the vertical edge projection of column x. Regions of license plates produce a high edge projection at their position, so the highest peaks of the edge projections are the best candidates for license plates. Sometimes other regions like the regions of traffic signs or the cars grille produce even higher projections at their positions than license plates. An approach to suppress such objects, is to find a certain amount of license plate candidates and discard candidates which do not satisfy the license plate properties (e.g. characters are in the license plate or the regions width-to-height relation is higher than 1, usually about 4.5 to 5). Edge projection is affected by noise, so horizontal and vertical median filtering is done before edge detection. The horizontal median filter usually has more columns than rows and vice versa for the vertical median filter. We have used a horizontal median filter of size 20 by 1 and a vertical median filter of 1 by 20 pixel. For edge detection, the Sobel Kernel was used to detect edges in horizontal and vertical detection separately. Horizontal and vertical median filtering results can be seen in fig. 2 and horizontal edge detection results are shown in fig. 3.

Figure 2: Horizontal and vertical median filtering.

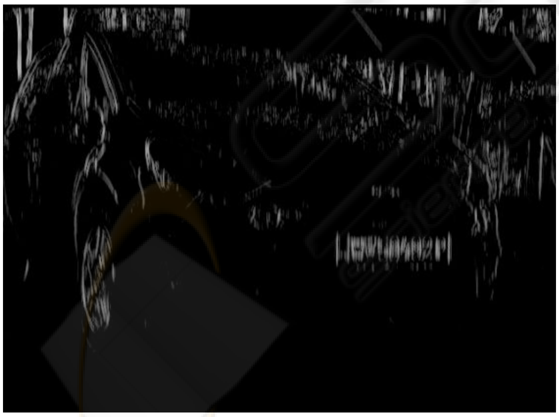

Figure 3: Result of horizontal edge detection on a median filtered image.

3.1.2 Band Clipping and Plate Clipping

The clipping of the top and bottom boundaries of the license plate is called band clipping and the clipping of the left and right boundaries is called plate clipping (Martinsky, 2005). In order to find the best candidates, the highest peaks of the projections have to be found which is computed as follows:

$$Hy_{max} = \arg\max\{ P_y \} \tag{3}$$

Where P_y is the resulting histogram of the horizontal projection and Hy_{max} is the maximum value of this histogram. To be able to find the boundaries of the plate, the histogram has to be iterated from the position of the maximum value in both directions until the histogram value reaches a value which is lower than Hy_{max} multiplied by a factor which is the threshold value usually chosen as 0.42. This value was chosen by experimental observation of the results. This value turns out to be the best value for the given image quality and compression which sparsely influence the edge projection. Equation 4 and 5 describe the computation of these boundaries where H_{y0} is the lower boundary and H_{y1} is the upper boundary of the license plate, c_y is the threshold value and P_y is the histogram (Martinsky, 2005). y_0 and y_1 denote the image-height limitations (y_0=0, y_1=image-height) A possible result of this process is illustrated in fig. 4.

$$Hy_0 = \max_{y_0 \leq y \leq Hy_{max}} \{P_y \leq c_y \cdot P_y(Hy_{max})\} \tag{4}$$

$$Hy_1 = \min_{Hy_{max} \leq y \leq y_1} \{P_y \leq c_y \cdot P_y(Hy_{max})\} \tag{5}$$

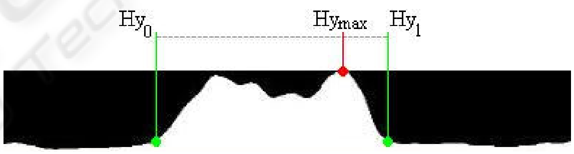

Figure 4: Boundary detection of the edge projections.

Equivalent calculations are done when calculating the right and left boundaries where the edge values are only taken from the clipped boundaries of the previous found horizontal band (Martinsky, 2005).

3.1.3 License Plate Candidates

The candidate region found by taking the highest peak of the edge projection is not necessarily the region of the real license plate. The lights of a car and the cars grille also produce high edge projection values, so these candidates have to be rejected and other peaks have to be analyzed. Our experience was that three candidates for the horizontal projection and three candidates for the vertical projection (for each of the three horizontal projections) are enough to find the real license plate. In fig. 5, the edge projection at the position of the light is higher than the projection of the plate. This is the reason why the regions at the lights are detected as candidates (red rectangles). After analyzing the regions, no information of a license plate could be found, so the three best candidates are rejected and the plate was found at the second highest projection of the horizontal edge histogram. The algorithm for finding the license plate can be described as follows:

```
FOR i = 1 to 3
    FOR j = 1 to 3
        Find top and bottom limits
        Find left and right limits
        IF region is a license plate
            FOUND = true
            BREAK
        ELSE
            reject region
            continue
        ENDIF
    ENDFOR
ENDFOR
IF FOUND == true
    further classification
ELSE
    reject frame
ENDIF
```

Figure 5: Candidate finding and rejection.

3.2 Character Segmentation

Before the license plate can be classified, each character has to be extracted and classified separately. There are several possibilities to extract the characters like blob analysis or segmentation using the vertical edge projection. We used blob analysis because the information of the blobs can be used for candidate selection. To reduce the complexity, the license plate region is converted into a binary image. In order to detect connected components (blobs) in the region, binarization of the plate region has to be done to reduce the complexity. After the binarization, the license plate is segmented into foreground and background, where the foreground are the characters and the background is the plate itself. A threshold value is calculated using the algorithms in (Ridler and Calvard, 1987). For local non-uniform illumination conditions, the algorithm in (Rae Lee, 2002) should be used.

The threshold is then used for the whole plate region. This can lead to problems when the plate is more illuminated in some regions than in others or the plate is polluted in some regions. To solve that problem the approach of Chow und Kaneko is used which separates the region into sub-regions (e.g. 2 by 2 or 3 by 2 etc.), calculates the threshold for each sub-region as described in the steps above and the threshold for pixels between sub-regions are interpolated using the weight of all neighboring subregions. The result of this process is illustrated in fig. 6.

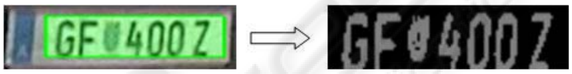

Figure 6: Binarization of the license plate.

3.2.1 Segmentation using the Vertical Edge Projection

This approach takes the vertical edge projection of the plate region which is calculated in the localization process and computes the first derivative of the histogram to find black to white and white to black changes (Martinsky, 2005; Zhang Y., Zhang C., 2003). From the resulting histogram, the highest peaks are at the positions of the character separations. The algorithm takes the highest peak and separates the plate into two subregions at this position. For each sub-region, the highest peak is taken and the plate is separated again. This process is done until the region reach a certain width or the peaks are to low compared to the highest value of the whole histogram. A possible result can be seen in fig. 7. For each sub-region, noise is removed and the character blobs are extracted for further classification.

3.2.2 Segmentation using Blob Analysis

Feng Yang et.al. uses region growing to detect blobs with a given set of seed points (Yang et al., 2006). The main problem of this approach is, that the seed points are chosen by taking the maximum gray-values of the image which can lead to skipped characters when no pixel in the characters blob has the maximum value. In order to solve this problem all blobs should be considered as characters. In fact the algorithm will find more blobs than characters so further analysis is necessary. From the binary image, the characters can be extracted directly by analyzing the connected components of the region. For each blob the size and the upper and lower boundary is stored for the analysis. The median value of the sizes is computed and all blobs with a difference of more than 10% are discarded. The same procedure is done for the positions of the blobs. Blobs like the one in fig. 7 after “SB” are discarded in this process. A typical result is illustrated in fig. 8. The third character from the left is not classified as a character due to the size of the blob and therefore discarded.

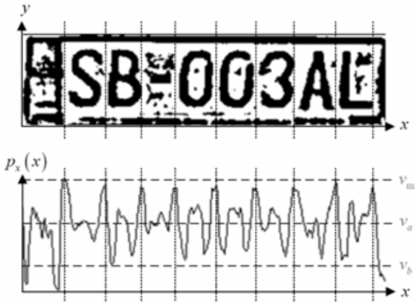

Figure 7: Segmentation using the vertical edge projection from (Martinsky, 2005).

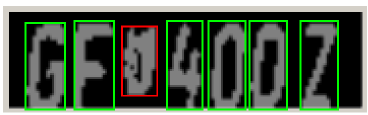

Figure 8: Blob extraction using blob analysis.

3.3 Feature Extraction and Character Classification

In our approach, character classification is done by using a decision tree. In each stage, characters are separated by using a comparison of features until only one character is in each leaf. The advantage of this approach is, that it is not relevant that one feature alone cannot separate characters and the decision tree is better, the more features are used. For each character features are extracted like:

- center of mass
- compactness
- euler number
- signature

and other features (Kim H. and Kim J, 2000; Peura and Iivarinen, 1997; Siti Norul Huda Sheikh et al., 2007). The signature of the blob is computed by calculating the distance from the image border to the first blobs border-pixel for each side (top-to-bottom, left-to-right, bottom-to-top and right-to-left). It is then normalized to a signature of 32 values (eight on each side) and each value of the signature is used as a feature. Another set of features are extracted by so called “zoning” of the blob. Zoning separates the blob into zones where local features of each zone are computed. The signature of the zones can also be used as features.

Before we are able to classify characters, we first have to train the characters by computing all features for the whole training set (about 250.000 characters are used) and compute feature-to-feature relations which separates the characters best. Decisions like

```
If feature_1 > threshold then
```

are not used due to the variance to rotation and scale of most features. Instead following relations are used:

```
if feature_1 > feature_2 then
```

The results are decisions like “the lower left zone’s average gray level is higher than the top right zone’s gray level”. The decision tree can look like the tree in fig. 9. Due to the decisions, the tree is a binary decision tree. The result of a decision stage in the training phase could be that 99.5% of all samples of *character1* are classified to be on the left sub-tree, 99.7% of all samples of *character2* are classified to be on the right sub-tree and 70% of all samples of character3 are also classified to be on the right subtree. In such cases *character3* is taken in both subtrees for a more precise decision when other features are taken for classification.

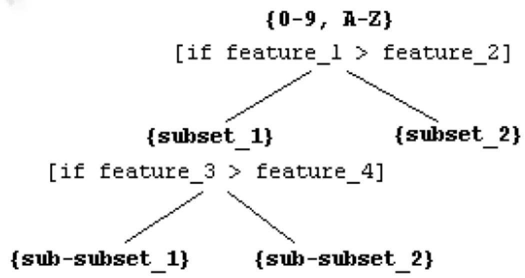

Figure 9: Binary decision tree.

3.4 The Use of Temporal Information

A license plate of a car may be present for 30 frames, which means that 30 classifications are stored for this plate. The classifier stores the data of each character for all frames. In each frame all characters are classified and the number of characters is stored. At the actual frame the classification includes all previous classifications of that license plate. Consider the following example. The license plate is classified four times:

- 128A
- 28A
- 12BA
- 128H

We call these classifications *single classifications*. The second classification did not recognize the first character so all characters are shifted to the left which causes errors in classification. For that reason the first step is to compute the median of the number of characters of the classifications. In this case the median value is four characters. For further analysis, only classifications with the number of four characters are taken to prevent, that classifications like the second one in the example affects the classification. After this step the following license plates are left:

- 128A
- 12BA
- 128H

The second step is the computation of the median of the characters for each position which results in the classification “128A”. This process is done in each frame which leads to a converging classification what we call the *temporal classification* because it is not the final decision of the license plate. The car corresponding to the license plate which is in the process to be detected may drive out of the cameras view. The license plate in that case is partly outside of the image and the classification recognizes less characters than before for these frames. This can lead to misclassification and the temporal classification converges to a wrong license plate. To suppress this problem, the final decision is computed by taking the maximum occurrence of the temporal classifications. Table 1 illustrates the process of the classification steps by an example. (The real license plate is 128A).

Table 1: Example of a classification process.

# Frame	Single classification	Temporal classification
1	128A	128A
2	28A	128A
3	128H	128A
4	12B	128A
5	12B	128A
6	12B	128A
7	12B	12B

The temporal classification of the last frame where the license plate is visible should be the best classification because of the converge progress. In this example the last three frames did not recognize the most right character due to the position of the car which may be partly outside the view. Due to this problem, the median of the numbers of characters at that position is three instead of the correct number of four so the single classifications with three characters are taken for the temporal classification. The final classification in this example would be “12B” which is wrong. In that case all temporal classifications are used to compute the classification of maximum occurrence. “128A” has the maximum occurrence in this example although the single classifications are correct only in one frame of seven. Because of the fact, that later temporal classifications include more single classifications, the temporal classifications should be weighed in order to support this approach. The temporal classifications are weighed by their number of including single classifications. The temporal classification in the first frame contains only one single classification, where the temporal classification in frame 7 contains 7 single classifications. The final result of the example from table 1 can be calculated as follows. The final classification is “128A” with a score of 21:

Score of “128A” = 1+2+3+4+5+6 = 21

Score of “12B” = 7

4 EXPERIMENTAL RESULTS

The huge amount of classifications for each license plate requires a good computational performance of the algorithm. Our implementation was tested on an Intel Pentium M 1.8GHz where the performances are shown in table 2. The input images are of dimension 640 by 480 pixels and three channels. The vertical edge detection was only computed on the region of the horizontal band. The table shows the minimum and maximum performances if a license plate was found or not.

Table 2: CPU performances of different detections.

No plate min	No plate max	Plate min	Plate max
290 fps	350 fps	80 fps	250 fps

The classification rate of the license plate is subdivided into three categories. One category is the detection if a license plate is present in the actual frame or not. The second category is the classification of each character and the last category is the result of the final decision of the license plate classification. The results can be seen in table 3, where about 350.000 characters are classified and ~2000 license plates are found and classified.

Table 3: Classification performances.

Character classification	License plate detection	License plate classification
~98.5%	100%	~98%

The characters and the 2000 license plates are extracted from a 12 hours video sequence and manually annotated for evaluation and training tests. The conditions for the license plate classification varied over time from dawn until dusk. The video sequence was recorded with a static camera at a parking garage facing the incoming cars as illustrated in figures 10 to 14. The detection if license plates are present in one frame was correct in every case. Sometimes when no license plate was present and other objects caused a high edge projection, candidate finding was started but in such cases all candidates are rejected. Fig. 10 to 14 show results of our approach.

Under bad conditions the algorithm performs not better than the single classification because if the single classifications of a single car are incorrect in more than 50%, the statistical analysis chooses a incorrect license plate. The statistical analysis takes the correct license plate if the single classifications of a single car are correct in more than 50%. In general the percentage of correct single classifications has to be higher than the most frequent incorrect candidate for the statistical analysis to choose the correct classification.

In fig. 10 the license plate is partly polluted (“9” and “R”) so that the classification fails. Fig. 11 illustrates a correct temporal classification due to the weight of the previous classifications although the light may be a reason for segmentation errors. In fig. 14 the license plate was not correctly localized. Due to the number of single classifications, the temporal classification is not affected.

Figure 10: License plate with polluted characters.

Figure 11: Correctly identified plate although the lights of the car influence the segmentation.

Figure 12: Converging classification.

Figure 13: Correct temporal classification.

Figure 14: Correct temporal classification although incorrect single classification.

5 CONCLUSIONS AND FUTURE WORK

In this paper a new approach of license plate recognition is presented where the license plate is not recognized only in one frame but in several consecutive frames. For that a statistical approach is presented in order to improve the classification result. The single classification approach can be adapted but in case of a real-time system, the single classification approach should be able to be executed several times per second. The single classification approach used in our work performs at 70% classification rate where our statistical analysis improves that result to 98%. It can be predicted that a better single classification approach combined with our approach will achieve almost a classification rate of 100% because single classifications where the license plate is visible under bad conditions can be suppressed. Our approach extends existing approaches by analyzing the classifications in each frame by the help of the information from image sequences. This extension always leads to a better classification result. For future works the recognition should be independent for the classification of other countries. For that a decision tree for each countries license plate characters and one decision tree for a "country decision" is built. In the first step the characters are analyzed to which country the characters belong and in the second step the corresponding decision tree to this country is chosen to classify the characters.

ACKNOWLEDGEMENTS

This work was partly supported by the CogVis[1] Ltd. However, this paper reflects only the authors' views; CogVis[1] Ltd. is not liable for any use that may be made of the information contained herein.

REFERENCES

Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., et al. (2006). A license plate-recognition algorithm for intelligent transportation system applications. *Intelligent Transport Systems, 7(3):* pages 377-392.

Feng Yang, Zheng Ma, Mei Xie, (2006). A novel approach for license plate character segmentation. In *International Conference on Industrial Electronics and Applications,* pages 1-6.

Kim H., Kim J., (2000). Region-based shape descriptor invariant to rotation, scale and translation. *Signal Process. Image Commun. 16 (2000)* pages 87-93.

Martinsky O., (2007). Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems, B.SC Thesis, Brno.

Peura M., Iivarinen J., (1997). Efficiency of simple shape descriptors. In *Proceedings of the Third International Workshop on Visual Form*, pages. 443-451.

Rae Lee B., (2002). An active contour model for image segmentation: a variational perspective. *In Proc. of IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1585-1588.

Ridler T.W., Calvard S., (1987). Picture Thresholding Using an iterative Selection method. *IEEE Trans. on Systems, Man, and Cybern., vol. 8*, pages 630-632.

Siti Norul Huda Sheikh A., Marzuki K., et al., (2007). Comparison of feature extractors in license plate recognition. In *AMS '07: Proceedings of the First Asia International Conference on Modelling & Simulation*, pages 502-506.

Xiao-Feng C., Bao-Chang P., Sheng-Lin Z., (2008). A license plate localization method based on region narrowing. *Machine Learning and Cybernetics, 2008 International Conference, Volume 5, 12-15 July*, pages 2700-2705.

Zhang Y., Zhang C., (2003). New Algorithm for Character Segmentation of License Plate, *Intelligent Vehicles Symposium*, pages 106-109.

Zhong Qin., Shengli Shi., Jianmin Xu., Hui Fu., (2006). Method of license plate location based on corner feature. In *World Congress on Intelligent Control and Automation*, pages 8645-8649.

[1] http://www.cogvis.at