A CAM-Based Pattern Accumulated Vector Method for Real-Time Character Recognition of License Plates

Yon-Ping Chen, Tien-Der Yeh, Feng-Chou Ni and Chung-Hsien Chang

Electrical and Control Engineering, National Chiao Tung University,, 1001 Ta Hsueh Rd., Hsinchu / Taiwan, ROC

Abstract

A novel Pattern Accumulated Vector(PAV) method implemented by CAM-based architecture has been proposed for character recognition in license plate recognition(LPR). Instead of comparing all the pattern blocks, the PAV method adopts the principal pattern blocks as feature vectors so that the computation complexity is highly reduced. Furthermore, the PAV method adopts the extended templates for the character recognition and is proved to have higher recognition rate when dealing with license plates consisting inclined characters. Finally, the CAM architecture is used to further reduce the processing time and makes the PAV method more efficient for real-time LPR systems.

Key words:

Content Addressable Memory(CAM), Pattern Accumulated Vector(PAV), License Plate Recognition(LPR), Character Recognition, principal pattern blocks.

I. INTRODUCTION

The license plate recognition, or LPR in short, has been a popular research topic for several decades [4] [20] [28]. Up to now, an LPR system still faces some problems concerning various light condition, image deformation, and processing time consumption [20] [28]. In general, there are three fundamental steps required for the process of an LPR system, including the license plate extraction, the character segmentation, and the character recognition. Since many methodologies have been well developed for the first two steps, the license plate extraction and the character segmentation [1][4], this paper will only focus on the third step, the character recognition, especially for inclined images[20]. To deal with the problems caused by inclined images, a novel method called the Pattern Accumulated Vector(PAV) method is proposed; besides, the Content Addressable Memory(CAM) [5]-[8] technique is adopted for implementation.

The proposed PAV method is mainly based on a feature vector called the pattern accumulated vector, which is obtained by decomposing each input character image into a series of 3×3 pattern blocks, and then executes the character recognition by comparing the feature vectors between the input character image and the templates. If all the pattern blocks are used in comparison, it will cost a large amount of processing time. In order to save the processing time as well as to retain a high recognition rate, the PAV method neglects the pattern blocks useless in character recognition and only includes the so-called principal pattern blocks. As a result, the processing time is highly reduced since the principal pattern blocks are found less than 1/20 of all the pattern blocks. Meanwhile, a special methodology to determine the principal pattern blocks from all the pattern blocks is also proposed. To further save the processing time, the well-known parallel comparison technique, called Content Addressable Memory(CAM), is employed to implement the comparison of principal pattern blocks. Therefore, the CAM technique applied to the PAV method is evidently useful for real-time license plate recognition systems, not only to save the processing time but also to reduce the power consumption.

To demonstrate that the PAV method with CAM is suitable for the character recognition of inclined images, two sets of templates, standard and extended, are used in the experiment for comparison. From the experiment with standard templates, the recognition rate is decreased due to inclined character images, which often happen due to sidewise camera locations. Such drawback can be improved by the extended templates which further add the information of inclined character images into the standard templates. Significantly, with the extended templates, the recognition rate is high around 99.56% for physical character images, normal or inclined. As for the processing time, the experiment is designed to compare the CAM-based and RAM-based architectures. From the experiment results, the CAM-based architecture is faster around 40% comparing to the RAM-based architecture.

II. The PAV Method for Character Recognition

A novel methodology called Pattern Accumulated Vector (PAV) method suitable for the character recognition of license plates is proposed in this paper. This section starts from introducing the fundamentals of the PAV method, such as the basic elements “pattern block”, the decomposition process, and the character recognition by calculating distances of the feature vectors. Then the so-called principal pattern blocks (PPBs) will be introduced to replace all the 512 pattern blocks in the feature vector to reduce computation complexity in vector comparison. Meanwhile, the recursive selection method to

Manuscript received January 5, 2009
Manuscript revised January 20, 2009

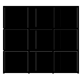
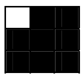
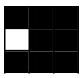
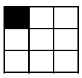
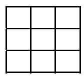
select the principal pattern blocks is also proposed to ensure high recognition rate is retained by the selected candidates. Finally, the PAV method applied to the extended templates is proposed in Section II-C to further deal with inclined characters.

A. The Fundamentals of the PAV Method

The fundamental element in PAV is called the pattern block, an l-by-l square binary image, so there are $2^{l\times l}$ types of pattern blocks totally. Since the size of the pattern block chosen for the proposed PAV technique is 3-by-3, there are $2^{3\times 3} = 512$ types of pattern blocks as shown in Table I. Each pattern block is denoted by a serial number, S.N. = $(a_1\ a_2\ a_3)$ where $a_i \in \{0, 1, 2, \ldots, 7\}$, for $i = 1, 2, 3$, and a_i is determined from the order of the pixels in the i-th column. For example, the pattern block of No.511 has a serial number (677), which means it contains pixels (1, 1, 0), (1, 1, 1), and (1, 1, 1) in the 1st, 2nd and 3rd columns, respectively. All the 512 pattern blocks are collected in the set

$$\boldsymbol{P}_{all} = \{(0,0,0),(1,0,0),(2,0,0),\ldots,(7,7,7)\} \qquad (2.1)$$

TABLE I
TYPES OF PATTERN BLOCKS.

Pixel order	1 4 7 / 2 5 8 / 3 6 9					
Pattern Block Type				(		
No. in order	1	2	3	…	511	512
Serial Number $(a_1a_2a_3)$	000	100	200	…	677	777

There are 36 binary templates shown in Fig.1 for the license plate characters, "0" to "9" and "A" to "Z", denoted as $\boldsymbol{B_0}$ to $\boldsymbol{B_{35}}$ in order, extracted from clear, non-distorted license plates. Templates $\boldsymbol{B_0}$ to $\boldsymbol{B_{35}}$ are called standard templates

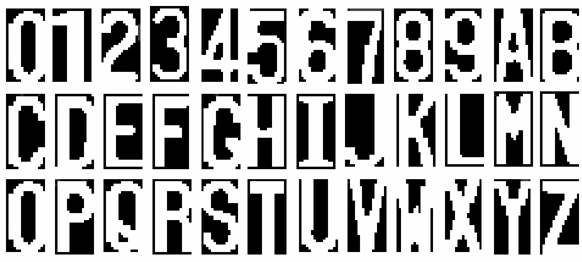


Fig. 1. The binary standard templates.

Each template is fixed at size 15×30 and decomposed into (15-2)×(30-2)=364 pattern blocks. The decomposition process is explained by an example of $\boldsymbol{B_2}$ in Fig.2, the binary template of character "2". Its top three rows are decomposed into thirteen 3-by-3 pattern blocks in a way of shifting one column by one column. Next by shifting one row down, i.e., from row1, row2, row3 change to row2, row3, row4, the same process applies to the next 3 rows to obtain the other thirteen 3-by-3 pattern blocks.

(2.1)

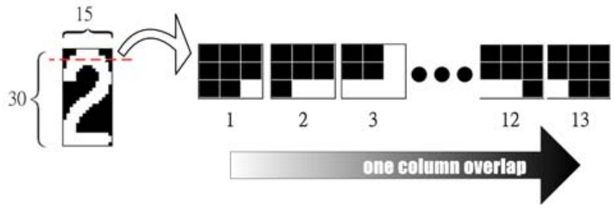


Fig. 2. The pattern blocks in the top three rows of the template "2"

By shifting one row by one row, the whole procedure of decomposition creates 364 pattern blocks as given in Fig.3, which shows that the pattern blocks combination of template "2" has 91 pattern blocks of (000), 11 pattern blocks of (100), … and 94 pattern blocks of (777).

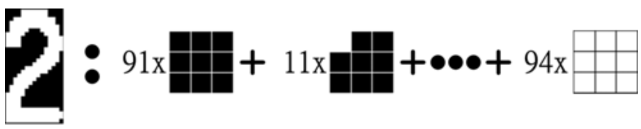


Fig. 3. The pattern blocks combination of the template "2"

There are 36 different feature vectors generated from the templates $\boldsymbol{B_0}$ to $\boldsymbol{B_{35}}$ and the n-th one is denoted as $\boldsymbol{u}_n$. The $\boldsymbol{u}_n$ is a 512-by-1 vector with entries

$$\boldsymbol{u}_n(i) = \lambda_{(a_1a_2a_3),n}, \quad 0 \le i \le 511 \qquad (2.2)$$

where $\lambda_{(a_1a_2a_3),n}$ is the number of occurrence of the pattern block $(a_1a_2a_3)$ in $\boldsymbol{B}_n$. Viewing from Fig.3 of template $\boldsymbol{B}_2$, its feature vector is $\boldsymbol{u}_2 = [91, 11,\ldots,94]^T$. Table II shows the combinations of all the pattern blocks of $\boldsymbol{B_0}$ to $\boldsymbol{B_{35}}$.

TABLE II:
THE COMBINATION OF 512 PATTERNS FOR THE 36 TEMPLATES.

Templates / Serial Number of Pattern Block	0 B_0	1 B_1	2 B_2	…	9 B_9	A B_{10}	B B_{11}	…	Y B_{34}	Z B_{35}
(000)	73	85	91	…	76	50	73	…	132	107
(100)	1	0	11	…	1	2	2	…	2	7
(200)	0	0	0	…	0	0	0	…	0	0
⋮	⋮	⋮	⋮		⋮	⋮	⋮		⋮	⋮
(773)	5	0	14	…	5	5	6	…	4	9
⋮	⋮	⋮	⋮		⋮	⋮	⋮		⋮	⋮
(677)	7	2	13	…	8	5	7	…	2	9
(777)	126	211	94	…	98	125	135	…	78	112
total	364	364	364	…	364	364	364	…	364	364

After the decomposition of the templates, the input character image for recognition, also called the test character, will be decomposed by the same process. The feature vector of the test character, denoted as $\boldsymbol{u}_t$, is then compared with each of $\boldsymbol{u}_n$ by determining all the Euclidian distances as

$$d_{t,n} = \|\boldsymbol{u}_t - \boldsymbol{u}_n\|^2 = \left[(\boldsymbol{u}_t - \boldsymbol{u}_n)^T(\boldsymbol{u}_t - \boldsymbol{u}_n)\right], \quad 0 \le n \le 35, \tag{2.3}$$

where $d_{t,n}$ can be used as similarity indices between the test character and $\boldsymbol{B}_n$. Then, the test character will be identified as the character of the template image with minimum distance. If the total number of test characters is N_1, and N_2 out of N_1 are identified correctly, then we get the recognition rate $r(\boldsymbol{P}_{all}) = N_2 / N_1$ where $\boldsymbol{P}_{all}$ is the set of all the pattern blocks as (2.1) and is what the feature vectors are composed of.

B. Selecting the Principal Pattern Blocks

If all the 512 pattern blocks are used to compose feature vectors as Section II-A, the dimension of the feature vectors will be very large and thus requires a large amount of processing time. In order to reduce the computation complexity and processing time consumption, it is necessary to delete the insignificant pattern blocks and keep those useful ones, called the principal pattern blocks or PPBs in short, for the character recognition. These PPBs will be determined by the recursive selection method shown in Fig. 4. The following three conditions are used to filter out the insignificant pattern blocks to collect the candidates of PPBs.

C1. Delete the pattern block $(a_1a_2a_3)$ if $\lambda_{(a_1a_2a_3),n} = 0$ for all the binary templates $\boldsymbol{B}_n$, $n = 0,1,\ldots,35$. Each of the remaining pattern blocks will be further checked by condition C2 and then C3 to determine whether it can be a candidate of the PPBs or not.

C2. Let

$$\bar{\lambda}_{(a_1a_2a_3)} = \max\left(\left|\lambda_{(a_1a_2a_3),n_1} - \lambda_{(a_1a_2a_3),n_2}\right|\right) \quad \text{for } n_1 \ne n_2. \tag{2.4}$$

A pattern block $(a_1a_2a_3)$ is chosen as a candidate of the PPBs if $\bar{\lambda}_{(a_1a_2a_3)} \ge 5$, which implies the pattern block $(a_1a_2a_3)$ is useful to distinguish the binary templates $\boldsymbol{B}_{n1}$ and $\boldsymbol{B}_{n2}$.

C3. Let

$$N_{(a_1a_2a_3)} = \sum_{n=0}^{35} \sigma\left(\lambda_{(a_1a_2a_3),n}\right), \tag{2.5}$$

where

$$\sigma\left(\lambda_{(a_1a_2a_3),n}\right) = \begin{cases} 1, & \lambda_{(a_1a_2a_3),n} > 0 \\ 0, & \lambda_{(a_1a_2a_3),n} = 0 \end{cases} \tag{2.6}$$

A pattern block $(a_1a_2a_3)$ not satisfying C2, is chosen as a candidate of the PPBs if it satisfies $N_{(a_1a_2a_3)} \le 2$. It implies the pattern block $(a_1a_2a_3)$ is special and then potentially able to distinguish some binary templates.

Now, let's discuss the recursive selection method in Fig.4 to select the desired PPBs. The major process in Fig. 4 can be divided into two steps. The step one of Fig.4 is made up of the three conditions C1-C3 and outputs the candidates of the PPBs, which will be further filtered by step two to find the PPBs for the recognition process.

A selection cycle is started by inputting a pattern block out of the 512 to step one and is terminated by either step one, if the pattern block is disqualified, or step two, if the insignificant pattern blocks are removed. The whole process stops after all the pattern blocks have been checked.

In step one, a pattern block is disqualified if it can't meet the requirement in C1-C3; otherwise it is left to be a candidate of PPBs and added into the candidate set $\boldsymbol{P}$. Before the completion of step one, it is necessary to perform character recognition to obtain the recognition rate $r(\boldsymbol{P})$ by recognizing the 36 standard templates for next step.

In step two, the insignificant candidates in set $\boldsymbol{P}$ will be filtered out and only those useful for the recognition rate can be kept in $\boldsymbol{P}$. Let the candidate set $\boldsymbol{P}$ after step one is

$$\boldsymbol{P} = \{P_i \mid i = 0,1,2,\cdots,h-1\}, \tag{2.7}$$

where $P_i = (a_1a_2a_3)_i$ is the *i*-th candidate and *h* is the total number of the candidates. The decision procedure is designed in a one-by-one strategy from P_0 to P_{h-1}. First, delete P_0 and form a new candidate set $\boldsymbol{P}_{\sim 0}$, which contains P_1 to P_{h-1}, then find the recognition rate $r(\boldsymbol{P}_{\sim 0})$. If $r(\boldsymbol{P}_{\sim 0}) < r(\boldsymbol{P})$, i.e., P_0 is not an insignificant pattern block since the recognition rate is reduced without it, then P_0 is kept in the candidate set, otherwise it is taken away from the candidate set.

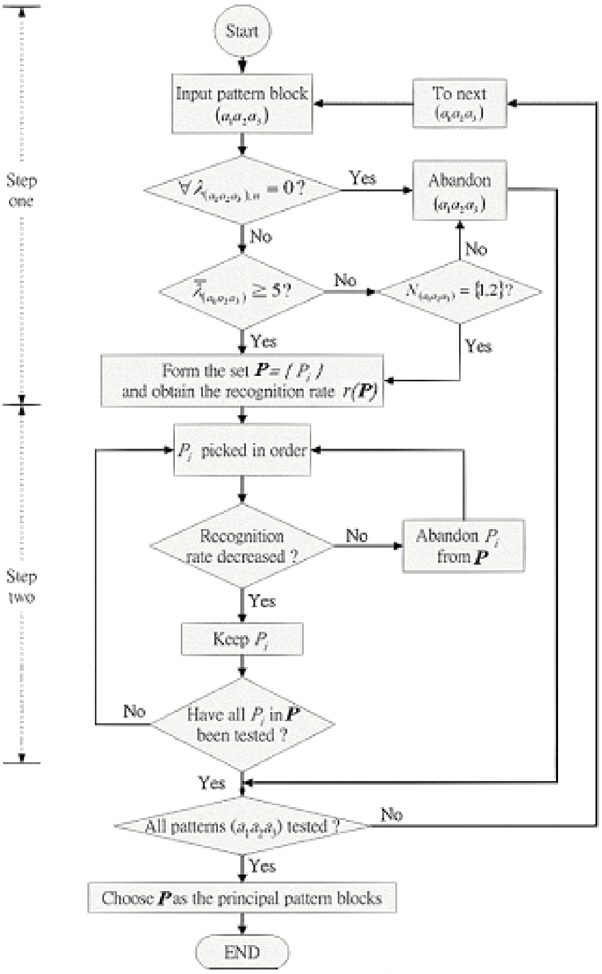


Fig. 4. The recursive selection of the PPBs.

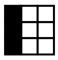
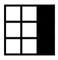
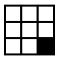
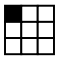
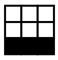
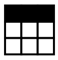
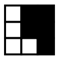
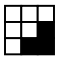
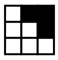
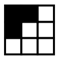
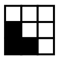
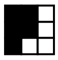
Following the same procedure for P_1 to P_{h-1}, the candidates kept in set $\boldsymbol{P}$ are all significant and the cycle is finished. If all the pattern blocks have been checked by the recursive selection method, the desired PPBs are obtained as the last candidate set $\boldsymbol{P}$, which has *m* candidates. And the new feature vector is rewritten as

$$v_n(i) = \lambda_{(P_i),n}, \quad 0 \leq i \leq m-1. \tag{2.8}$$

Note the number of entries is reduced from 512 to the last *m* which is generally less than 20, and thus will highly reduce the computation complexity of the whole process.

As in Table III, the PPBs are listed and the number of the PPBs is 12.

TABLE III
12 PPBS FOR 36 STANDARD TEMPLATES.

PPB number	Pattern	S.N.	Amount of recognized characters	Data of recognized characters
1		077	16	1, 3, 7, A, C, D, E, H, J, M, N, R, V, W, Y, Z
2		770	15	0, 1, 4, 7, 9, B, D, H, J, K, L, Q, U, V, W
3		773	14	0, 2, 4, 5, 6, 8, B, F, L, N, Q, W, X, Z
4		677	14	0, 2, 4, 5, 8, 9, B, H, N, O, S, W, X, Z
5		333	12	5, 6, 7, 8, E, F, H, I, P, R, T, Z
6		666	11	2, 3, 5, 6, B, E, G, J, L, S, Z
7		740	9	0, 3, 8, A, I, K, N, S, X
8		731	7	2, 3, 6 , C, J, O, Q
9		764	7	5, 9, G, L, P, S, Y
10		467	7	0, 2, 8, 9, M, O, U
11		137	6	0, 2, 6, B, O, S
12		047	5	4, A, O, U, Z

There are three threshold values adopted in the step one in Fig. 4, which are $\lambda_{(a_1a_2a_3),n} = 0$, $\bar{\lambda}_{(a_1a_2a_3)} \geq 5$ and $N_{(a_1a_2a_3)} \leq 2$. These values are determined from experiment which executes character recognition and records the distribution of successful recognition cases in Table IV. From the table, there are 416 types of pattern blocks which are useless since they belong to none of the characters to be recognized. As a result, the threshold $\lambda_{(a_1a_2a_3),n} = 0$ is used to delete the pattern blocks that match none of the templates.

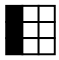
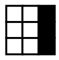
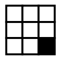
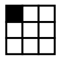
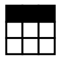
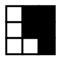
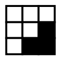
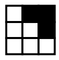
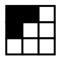
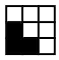
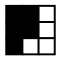
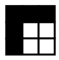
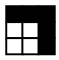
The second threshold value is $\bar{\lambda}_{(a_1a_2a_3)} \geq 5$, where $\bar{\lambda}_{(a_1a_2a_3)}$ is defined in (2.4). The major purpose of this threshold value is to find out the pattern blocks ($a_1a_2a_3$) which are significant to distinguish two templates as described in C2. According to the experiment result shown in Table III, it is found that $\bar{\lambda}_{(a_1a_2a_3)}$ is chosen to be 5 for best discrimination efficiency since a smaller $\bar{\lambda}_{(a_1a_2a_3)}$ often decreases the significance of the pattern block ($a_1a_2a_3$) in distinguishing certain templates.

The final threshold value is $N_{(a_1a_2a_3)} \leq 2$ or $N_{(a_1a_2a_3)} = \{1, 2\}$, which implies the pattern block ($a_1a_2a_3$) is special and then potentially able to distinguish some binary templates, as described in C3. From Table IV, it shows that there are 11 and 31 types of pattern blocks could be associated to recognize 1 and 2 characters, respectively. The reason to further include the pattern blocks of $N_{(a_1a_2a_3)} = \{1, 2\}$, not of $N_{(a_1a_2a_3)} = \{1\}$ or $N_{(a_1a_2a_3)} = \{1, 2, 3\}$, is based on the experiment that adding the pattern blocks of $N_{(a_1a_2a_3)} = \{1, 2\}$ results in better recognition rate than $N_{(a_1a_2a_3)} = \{1\}$ and $N_{(a_1a_2a_3)} = \{1, 2, 3\}$.

TABLE IV
THE DISTRIBUTION OF THE AMOUNT OF SUCCESSFUL RECOGNITION CASES.

Number of pattern blocks	Characters can be recognized
416	0
11	1
31	2
26	3
7	4
7	5
2	6
3	7
3	9
1	11
1	12
2	14
1	15
1	16
Sum : 512	

TABLE V
16 EPPBS OBTAINED FROM EXTENDED TEMPLATES

PPB number	Pattern	S.N.	Amount of recognized characters	Data of recognized characters
1		077	16	1, 3, 7, A, C, D, E, H, J, M, N, R, V, W, Y, Z
2		770	15	0, 1, 4, 7, 9, B, D, H, J, K, L, Q, U, V, W
3		773	13	0, 2, 5, 6, 8, B, F, L, N, Q, W, X, Z
4		677	13	0, 2, 4, 8, 9, B, H, N, O, S, W, X, Z
5		333	12	5, 6, 7, 8, E, F, H, I, P, R, T, Z
6		666	11	2, 3, 5, 6, B, E, G, J, L, S, Z
7		740	9	0, 3, 8, A, I, K, N, S, X
8		731	7	2, 3, 6 , C, J, O, Q
9		764	7	5, 9, G, L, P, S, Y
10		467	7	0, 2, 8, 9, M, O, U
11		137	6	0, 2, 6, B, O, S
12		047	5	4, A, O, U, Z
13		066	4	7, C, G, S
14		660	3	2, F, X
15		330	2	5, X
16		033	2	9, Y

C. The PAV Method Applied to the Extended Templates

Since the PPBs in Section II-B are obtained from standard templates, they are not suitable for characters with an inclined angle, which frequently occurs in LPR systems caused by factors such as sidewise camera locations.

In practical situation, the inclined angle of a test character is often smaller than 30 degree. To resolve the problems of those inclined characters, the 36 standard templates are extended to 108 templates, which are partitioned into 3 groups as below:

G1: Standard templates shown in Fig.1, including 36 templates, “0” to “9” and “A” to “Z”.

G2: Templates obtained by inclining G1 in 30 degrees clockwise(+30°) as in Fig. 5.

G3: Templates obtained by inclining G1 in 30 degrees counterclockwise(-30°) as in Fig. 6.

With the same PAV method proposed in Section II-B, the PPBs for extended templates G1, G2 and G3 will be different from the PPBs for standard templates G1. For convenience, the PPBs for extended templates G1, G2 and G3 are called the extended PPBs, or EPPBs in short. In table V, the number of EPPBs=16 is different from the number of PPBs=12 in previous section. Later, it will be found that experiment result with EPPBs has better recognition rate(99.56%) than PPBs(93.83%) for practical license plates.

Fig. 5. The binary templates with +30° inclined angle

Fig. 6. The binary templates with -30° inclined angle

In the next section, the way to implement the PAV method by CAM-based architecture will be introduced. In order to demonstrate its advantage, the PAV method is also implemented by the conventional RAM-based architecture [9]. Later, it will be found from the experimental results that the CAM-based architecture is indeed much better than the RAM-based architecture in processing time.

III. Implement by CAM Architecture

The typical architecture of Content Addressable Memory (CAM) is shown in Fig. 7 [9] [13], including the Comparand Register, the Memory Array, the Mask Register, the Word Select Register, the Responder And Contention Logic, and the Output Register.

The Memory Array provides the storage containing the associative words with contents relative to its address, which is an exceptional feature of the CAM. The Comparand Register stores the input data, which is the pattern to be compared with the words in the Memory Array. The Mask Register masks off the "don't care" bits in the input pattern. For example, if the input pattern is 01001101, but only the least four bits (bit 3 ~ bit 0) are needed for searching, then the Mask Register is enabled and set as 11110000. The Word Select Register contains *n*-bit data and is used to select which associative word in the Memory Array is active. When the associate word is active, it will be compared with the pattern output from Mask Register. The Responder And Contention Logic is capable for simultaneously receiving all the matched results from Memory Array and indicate the success or failure by setting or resetting the matched flag. If more than one word in the Memory Array is matched, the Responder And Contention Logic can resolve the contention among them and then send out the matched flag and matched address sequentially. As for the Output Register, it is employed to read out the matched data in the Memory Array.

The proposed PAV method is implemented by CAM-based architecture according to following description: load a 15×30 test character into CAM, each 3×3 pattern block of the character is transformed into a 1×9 word and stored into the Memory Array sequentially as in Fig. 8. As a result, a 15×30 test character will occupy 364 associate words. For easy implementation the Memory Array is made by 512 words and 148 of them will not be used. The Mask Register is set to be "000000000" since the comparison doesn't contain any "don't-care" bits. The Word Select Register selects the address range from 0 to 363 of the associate words occupied by the test character, not the total range from 0 to 512. Input the first pattern block of the 16 EPPBs into the Comparand Register and it will compare with all the Associate Words simultaneously. Then start the Responder And Contention Logic to generate the Matched Flag sequentially. The number of Matched Flag is accumulated externally as the matching number related to the first EPPB, and the matching number is stored in external memory. Note that the Matched Addresses are ignored since they are not used in the proposed PAV method.

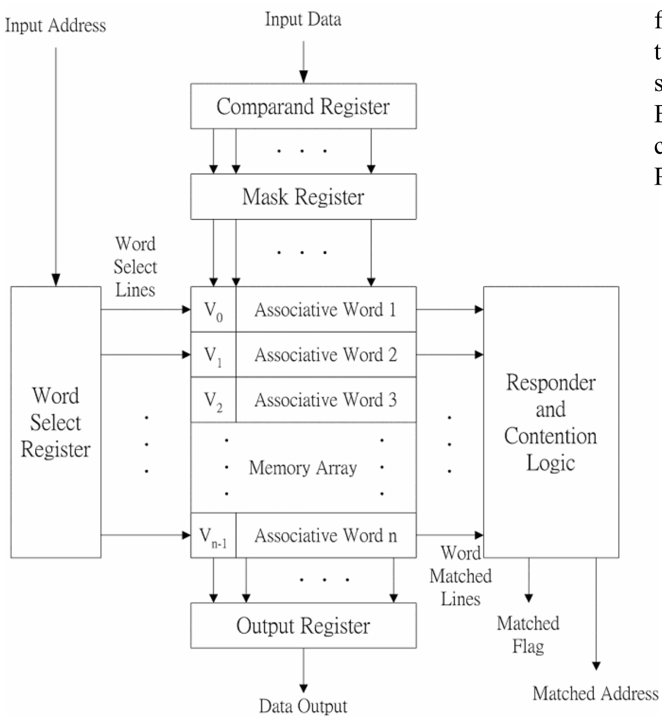


Fig. 7.: The typical architecture of CAM.

Following the same procedure for the other 15 EPPBs, the matching numbers related to the 16 EPPBs are obtained and put in order as the PAV of the test character. Compare the PAV of the test character with all the PAVs of the extended templates, G1, G2 and G3, to find the best one with shortest distance. Then the test character is recognized as the best one in the extended templates. This finishes the recognition process. Compare the PAV of the test character with all the PAVs of the extended templates, G1, G2 and G3, to find the best one with shortest distance. Then the test character is recognized as the best one in the extended templates. This finishes the recognition process.

IV. Experimental Results

Besides the proposed CAM-based architecture, a general RAM-based architecture is also made for comparing the difference of processing time. To keep the same experimental environments, both architectures are established and tested on the same development board (EP20K1500EBC652-1X) at the same clock frequency 25MHz. Besides, both architectures use the same database of characters from license plates for the experiment.

The database of license plates contains 1800 test characters and the experimental results are given in Table VI. From the table, the average processing time for the CAM-based system is 77.998s and for the RAM-based system is 128.367s. Evidently, the CAM-based system is faster about 50.4s, almost 40% improved in processing time. Besides, it is clear that both architectures have the same recognition rate. Moreover, the PAV method with EPPBs has a better recognition rate 99.56% when comparing to the recognition rate 93.83% of the case with PPBs.

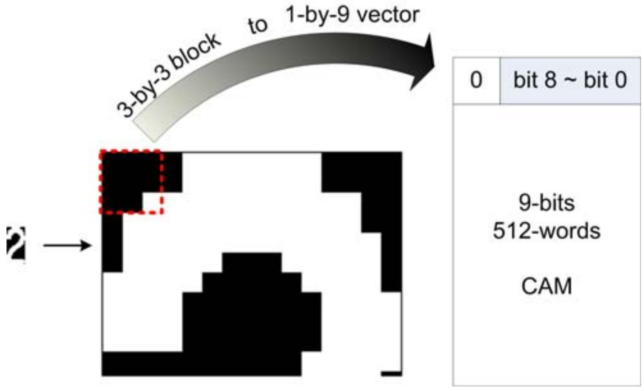


Fig. 8.: The pattern blocks stored in CAM.

TABLE VI
EXPERIMENTAL RESULTS OF CAM-BASED AND RAM-BASED SYSTEMS.

Architecture	Average Processing Time	Recognition rate	
		Using Templates by PPBs	Using Templates by EPPBs
CAM-based	77.998μs	93.83%	99.56%
RAM-based	128.367μs	93.83%	99.56%

The reason for the EPPBs to have a better recognition rate is because they are extracted from not only the standard templates but also the additional 2 groups of inclined templates, G2 and G3. But the PPBs are extracted from the standard templates only. It can be concluded that the PAV method with EPPBs implemented by CAM-based architecture is the best choice for both recognition rate and processing time.

V. Conclusion

This paper presents a novel CAM-based PAV method for character recognition in LPR systems. After choosing the principal pattern blocks by the recursive selection method, the computation complexity of the PAV method can be highly simplified since the dimension of the feature vectors is reduced. Besides, in order to deal with inclined character in physical license plates, the standard templates are replaced by the extended templates. From the experiment result, the success of the PAV method with the extended templates is verified from the high character recognition rate up to 99.56%. Most significantly, applying the CAM architecture to the PAV method indeed saves a large amount of processing time, which makes the PAV method more efficient for real-time LPR systems.

REFERENCE

[1] Y. Amit, D. Geman, and X. Fan, "A Coarse-to-Fine Strategy for Multiclass Shape Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1606-1621, Dec. 2004.

[2] H. Anderson, "Neural network machines," *Potentials, IEEE*, vol. 8, no. 1, pp. 13-16, Feb. 1989.

[3] R. F. Chang and W. M. Chen, "Adaptive Edge-based Side-match Finite-state Classified Vector Quantization with Quadtree Map," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 378-383, Feb. 1996.

[4] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic License Plate Recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 1, pp. 42-53, Mar. 2004.

[5] L. Chisvin and R. Duckworth, "Content-Addressable and Associative Memory: Alternatives to the ubiquitous RAM," *IEEE Transactions on Computer*, vol. 22, no. 7, pp. 51-64, Jul. 1989.

[6] C. Foster, *Content Addressable Parallel Processors*, John Wiley & Sons, Inc., New York. USA, 1976.

[7] T. Ikenaa and T. Ogura, "A Fully Parallel 1-MB CAM LSI for Real-Time Pixel-Parallel Image Processing," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 4, pp. 536-544, Apr. 2000.

[8] T. Jamil, "RAM versus CAM," *Potentials, IEEE*, vol. 16, no. 2, pp. 26-29, 1997.

[9] S. Impedovo, L. Ottaviano, S. cchinegro, "Optical Character Recognition - A Survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 5, no. 1-2, pp. 1-24, 1991.

[10] S. Kim, D. Kim, Y. Ryu, and G. Kim, "A Robust License Plate Extraction Method Under Complex Image Conditions," in *Proc. 16th International Conference on Pattern Recognition (ICPR'02)*, Quebec City, Canada, vol. 3, pp. 216-219, Aug. 2002.

[11] S. Kornachuk, L. McNaughton, R. Ghibbins, and N. D. Benoit, "A High Speed Embedded Cache Design With Non-intrusive BIST," in *Proc. IEEE 1994 MTDT Conference*, San Jose, Canada, pp. 40-45, Aug. 1994.

[12] A. Krikelis and C. Weems, "Associative Processing and Processors," *IEEE Transactions on Computers*, vol. 27, no. 11, pp. 12-17, Nov. 1994.

[13] C. S. Lin, J. C. Chang, and B. D. Liu, "A Low-power Pre-computation Based Fully Parallel Content Addressable Memory," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 654-662, Apr. 2003.

[14] K. J. Lin, C. W. Wu, and S. Member, "A Low-power CAM Design for LZ Data Compression," *IEEE Transactions on Computers*, vol. 49, no. 10, pp. 1139-1145, Oct. 2000.

[15] V. Madisetti, *VLSI Digital Signal Processors: An Introduction to Rapid Prototyping and Design Synthesis*, IEEE Press, 1995.

[16] I. Masaki, "Machine-Vision Systems for Intelligent Transportation Systems," *IEEE Intelligent Systems*, vol. 13, no. 6, pp. 24-31, Nov. 1998.

[17] B. McQueen and J. McQueen, *Intelligent transportation systems architectures*, Artech House, Boston. USA, 1999.

[18] M. Milward, N. Nunez, and D. Mulvaney, "Design and Implementation of a Lossless Parallel High-Speed Data Compression System," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 6, pp. 481-490, Jun. 2004.

[19] M. Motomura, J. Toyoura, K. Hirata, H. Ooka, H. Yamada, and T. Enomoto, "A 1.2-million transistor 33-MHz 20-b dictionary search processor (DISP) ULSI with a 160-kb CAM," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 5, pp. 1158-1165, Oct. 1990.

[20] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust License-plate Recognition Method for Passing Vehicles under Outside Environment," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2309-2319, Nov. 2000.

[21] K. Nogami, T. Sakurai, K. Sawada, K. Sakaue, Y. Miyazawa, S. Tanaka, Y. Hiruta, K. Katoh, T. Takayanagi, T. Shirotori, Y. Itoh, M. Uchida, and T. Iizuka, "A 9-ns Hit-delay 32-kbyte Cache Macro for High-speed RISC," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 1, pp. 100-106, Feb. 1990.

[22] B. Ramamurthi and A. Gersho, "Clasified vector quantization of images," *IEEE Transactions on Communications*, vol. COM-34, no. 11, pp. 1105–1115, Nov. 1986.

[23] S. H. Seo and M. R. Azimi-Sadjadi, "A 2-D Filtering Scheme for Stereo Image Compression Using Sequential Orthogonal Subspace Updating," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 1, pp. 52-66, Jan. 2001.

[24] V. Shapiro and G. Gluhchev, "Multinational License Plate Recognition System: Segmentation and Classification," in *Proc. 17th International Conference on Pattern Recognition (ICPR'04)*, Cambridge, UK, pp. 352-355, Aug. 2004.

[25] K. W. Wong, K. M. Lam, and W. C. Siu, "An Efficient Low Bit-Rate Video-Coding Algorithm Focusing on Moving Regions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 10, pp. 1128-1134, Oct. 2001.

[26] H. Yamada, Y. Murata, T. Maeda, R. Ikeda, K. Motohashi, and K. Takahashi, "Real-time String Search Engine LSI for 800-Mbit/s LANs," in *Proc. IEEE 1988 Custom Integrated Circuits Conference.*, Rochester, USA, pp. 21.6.1-21.6.4, May 1988.

[27] B. Yan and K. W. Ng, "A Novel Selective Motion Vector Matching Algorithm for Error Concealment in MPEG-4 Video Transmission over Error-Prone Channels," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1416-1423, Nov. 2003.

[28] Choudhury A. Rahman, Wael Badawy and Ahmad Radmanesh, "A Real Time Vehicle's License Plate Recognition System," *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 163 – 166, July 2003