Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Implementing VLPR systems based on TMS320DM642*

ZHU Le-qing†, ZHANG San-yuan, YE Xiu-zi
(*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)
†E-mail: zlq_26@163.com

Received Jan. 31, 2007; revision accepted Apr. 6, 2007

Abstract: This paper gives a practical schema for using DSP boards to construct Vehicle License Plate Recognition (VLPR) modules that could be embedded in any Intelligent Transportation System (ITS). Using DSP can avoid the heavy investment in dedicated VLPR system and improve the computational power compared to PC software environment. Low cost, high computational power, and high flexibility of DSP provide the License Plate Recognition System (LPRS) an excellent cost-effective solution to execute the major part of the recognition tasks. This paper describes a successful implementation of VLPR system based on Texas Instruments (TI)'s TMS320DM642. The DSP board acquires video (which could be output to a monitor for surveillance) from a camera, captures images from the video, locates and recognizes the license plates in images, and then sends the recognized results and related images after compression to a host PC through the network. Finally, the overall software is optimized according to the features of DM642 chip. Experiments showed that the DSP VLPR system performs well on the local license plates, and that the processing speed and accuracy can meet the requirement of practical applications.

Key words: License Plate Recognition (LPR), Embedded system, Image processing, DSP, DM642
doi:10.1631/jzus.2007.A2005 **Document code:** A **CLC number:** TP391.41

INTRODUCTION

Vehicle License Plate Recognition (VLPR) is an important technology which utilizes computer vision and pattern recognition in modern Intelligent Transportation Systems (ITS). The basic technologies used in automatic recognition system for vehicle license plates include digital image processing, pattern recognition and embedded system. A video camera takes the pictures of vehicles running on the road and then the system analyzes the pictures and gives out the recognition result. In this way, the system can complete the License Plate Recognition (LPR) automatically while vehicles are not affected. The complexity of transportation management is decreased, since VLPR technology can be used in automatic toll collection, traffic control and surveillance, automatic vehicles recognition, automatic highway accident report, access control, automatic parking control, vehicles location, vehicle thefts, enforcement, and border protection, etc. (Anagnostopoulos *et al*., 2006). VLPR technology is very important in maintaining city and traffic security, avoiding traffic jam, increasing service efficiency of transportation management, and thus relieving traffic tension comparatively.

Reported researches on LPR are mainly concentrated on the processing steps such as car plate detection, extraction of a license plate region, segmentation of characters from the plate, and recognition of each character. An enhanced cascaded tree style learner for car plate detection was proposed by Wu *et al*.(2006). The algorithm adopts the hybrid object features which include two kinds of simple statistical features, gradient density and gradient density variance which are based on vertical gradient of the image, and well-known Harr-like features. As for license plate region location, the methods of color detection (Cho and Jung, 1998), signature analysis (Barroso *et al*., 1997), edge detection (Bai and Liu, 2004) were proposed. In (Choudhury *et al*., 2003), plate character region's location is identified by tak-

* Project supported by the National Natural Science Foundation of China (No. 60473106), the Hi-Tech Research and Development Program (863) of China (Nos. 2007AA01Z311 and 2007AA04ZA5)

ing vertical color concentration in the image. Sliding Concentric Windows (SCWs) were used for faster detection of plate regions in (Anagnostopoulos *et al*., 2006). Computing techniques rooted in fuzzy technique were also proposed in (Chang *et al*., 2004). Any tilt in the captured image is corrected in this step. As for character segmentation, the character's gray level bands were detected to split characters. Character segmentation was realized by taking horizontal concentration of color in the image in (Choudhury *et al*., 2003). By using Laplacian Transformation, region growing and prior knowledge, the segmentation was more accurate and robust than the simple projection (Yang *et al*., 2006a). And for character recognition, there are genetic algorithms (Kim *et al*., 1996), artificial neural networks (Brugge *et al*., 1998; Parisi *et al*., 1998; Kim *et al*., 2000), fuzzy c-means (Nijhuis *et al*., 1995), Support Vector Machine (SVM) (Kim *et al*., 2000), Markov processes (Cui and Huang, 1997), and finite automata (Adorni *et al*., 1998). SCW method was employed for image binarization in conjunction with a probabilistic neural network (PNN) for character recognition in (Anagnostopoulos *et al*., 2006). In (Chil *et al*., 2006), a simple method that employs SVM for Chinese LPR was proposed. Character recognition was accomplished by matching patterns of horizontal and vertical histograms in (Choudhury *et al*., 2003). As for adaptation to changing environment, Naito *et al*.(2000) presented a sensing system utilizing two CCDs and a prism to split an incident ray into two lights with different intensities to cover wide illumination conditions from twilight to noon under sunshine.

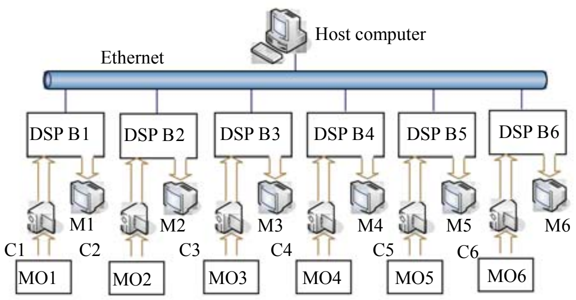

Fig.1 General architecture of the Vehicle License Plate Recognition (VLPR) system based on DSP board
B=Board; M=Monitor; C=Camera; MO=Monitor Object

This paper introduces a VLPR scheme based on TMS320DM642 DSP and is organized as follows. Section 2 gives a general overview of the system structure. Section 3 introduces the hardware design of the system. Software design of DSP system and host computer are given in Section 4 and Section 5, respectively. Experimental results are presented in Section 6, and finally, conclusions and future extensions are presented in Section 7.

SYSTEM OVERVIEW

The system includes a video camera, a DSP circuit board, a monitor and a personal computer serving as the host. The camera takes the video of vehicles running on the road. An analog composite video signal is received through S-port from the camera and decoded into a digital parallel BT656 stream; then, the stream flows into DM642's video port which gets image by decoding the BT656 stream, and the resulting image is stored into an SDRAM through Enhanced Direct-Memory-Access (EDMA). After the image is captured, the DSP processes the images, recognizes vehicle license plate from the images, and finally sends the recognition results and related images after compression to the host computer through Ethernet port. The host computer receives the information from the DSP boards, displays them on the screen and saves the information into the database. A user can review the results and related images by accessing the database. The DSP board also has an RCA composite video output port, the user can connect the port to a video monitor and see what is happening on the road.

Fig.1 is an example of VLPR application based on DSP board. There are 6 DSP boards in this system. We name every DSP board together with the camera and the monitor connected to it as a subsystem. Each subsystem handles one video stream that the camera records continuously of some monitored objects such as the status of a highway lane. There may be 6 or more lanes on a highway. It is assumed that there are 6 lanes on the highway here. If there are more or less lanes, just add on or cut off some DSP subsystems. It would be OK since the Ethernet could be extended flexibly within certain range. The DM642 chip on the board processes the image frames, recognizes the license plate's type, color and number, and uses JPEG algorithm to compress the image. Then, if the PC sends communication requirement to the DSP board, the result and the image in the buffer would be transmitted to the PC through the Ethernet immediately. The captured video stream is encoded and output to the monitor simultaneously.

HARDWARE DESIGN

The camera has an S-video port to output video signal to DSP board and the monitor has an RCA port to input video signal from DSP board. The key part of this system is the DSP board. In this section we mainly introduce the composition of DSP circuit board whose architecture is shown in Fig.2.

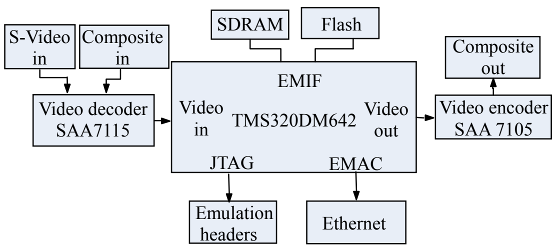

Fig.2 General architecture of DSP

The video signal input from the camera is sampled, A/D-converted, reorganized and then sent to the DSP, which processes the image frames of the video, trying to determine if there is a license plate in the image; if so, the DSP uses VLPR algorithm to recognize the license plate's type, color and number, and then uses JPEG algorithm to compress the image. DSP can transmit the recognized result and related compressed image to the host computer through the Ethernet. The host computer can communicate with several DSPs concurrently, require and accept recognized result and related images from different DSP boards. At the same time, the captured video is output to an encoder to restore the digital video into analog data which are sent to the monitor for display. Since DSP's on-chip memory is not enough for the software to run, and the whole software should be stored in a nonvolatile memory device, we installed SDRAMs and Flash for the DSP through the External Memory InterFace (EMIF).

Core chip DM642

The TMS320DM642 (DM642 for short) developed by Texas Instrument Incorporated is an excellent option for digital multimedia applications. It offers cost-effective solutions to high-performance DSP programming challenges with up to 5760 MIPS at a clock rate of 720 MHz. Its core processor has 64 general-purpose registers of 32-bit word length and 8 highly independent functional units—2 multipliers for a 32-bit result and 6 ALUs. The DM642 uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. It also has a 64-bit seamless EMIF, which can interface to synchronous and asynchronous memories and peripherals. The TMS320DM642 DSP is IEEE-1149.1 JTAG Boundary-Scan-Compatible. In order to support multimedia applications, the DM642 device has been integrated with 3 configurable Video Port (VP0, VP1, and VP2) peripherals and Multi-Channel Audio Serial Port (MCASP). The Ethernet Media Access Controller (EMAC) provides an efficient interface between the DM642 DSP core processor and the network. The DM642 EMAC support both 10Base-T and 100Base-TX, or 10 Mbps and 100 Mbps in either half- or full-duplex, with hardware flow control and QoS support (TI, 2005). The DM642 EMAC uses a custom interface to the DSP core that allows efficient data transmission and reception. Since DM642 has the above advantages, our VLPR system is cored with DM642 to accomplish real time video capturing, image processing, compressing, information transmission, etc.

Video input and output circuit

We adopted Philips Semiconductors' SAA7115H as our video decoder device. The analog video signal input from S-video port or composite port is clamped, anti-aliasing filtered, A/D-converted, YUV separated (Lambers, 2001), and then changed into digital parallel BT.656 video stream which is in turn sent to core chip DM642, which has 3 video ports VP0, VP1, and VP2. Here we use VP0 as the video input port to connect with the decoder SAA7115H and VP2 as the video output port to connect with the encoder. VP0 channel is configured into 8-bit BT.656 input video port. The line or field synchronization signals are included in the End of Active Video (EAV) and Start of Active Video (SAV) time based signals of BT.656 digital video flow. Setting up the parameters and reading the statutes of internal registers in

SAA7115H are carried on through I2C bus. DM642 decodes the BT.656 stream and gets the images, which are stored into the SDRAM automatically through the EDMA channel.

DM642 video output port gets the data from the SDRAM output buffer through EDMA and forms BT656 stream. We use Philips Semiconductors' SAA7105H to encode the BT656 stream and turn it into standard TV signal. The signal output from the SAA7105H is sent to the monitor through composite port.

Extended memory circuit

DM642 has 16 kB Level 1 (L1) program cache, 16 kB L1 data cache and 256 kB Level 2 (L2) memory/cache that is shared between program and data space and could be configured as mapped memory, cache, or combination of both. That is not enough for processing image. We extended two 32 MB SDRAM (MT48LC4M32B2) chips to store the original image data and an 8 MB Flash (AM29LV640MT) to store application program, which are mapped to the external memory space CE0 and CE1 of DM642, respectively.

Ethernet interface circuit

The system uses Intel's LXT971 as the high speed Ethernet physical layer self-adaptive transceiver. LXT971 supports the IEEE802.3 standard and provides Media Independent Interface (MII). It also supports MAC. DM642 integrates Ethernet MAC on chip, so LXT971 can realize seamless connection with DM642.

The circuit is shown in Fig.3, where PM44-11BG is a network transformer which also isolates the board circuit from the network circuit. The data transmitted from DM642 is transformed into data format that could be received by Ethernet physical layer in LXT971 and then transferred on the Ethernet through the RJ-45 port.

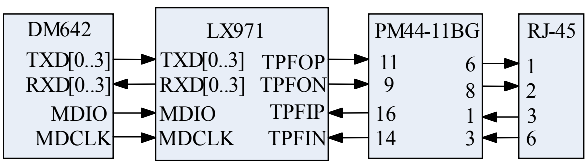

Fig.3 The Ethernet interface circuit

Power supply circuit

The whole DSP board is powered by a 5 V DC voltage. This 5 V DC voltage is transformed into a 1.4 V DC and a 3.3 V DC voltage to power DM642 core and I/O, respectively. It is also transformed into another 3.3 V DC to give power supply for video decoder, video encoder and other devices on the board. Since DSP needs two different power supplies, the power sequence must be considered. When the board is powering on, it must be guaranteed that the DSP core is powered on first, and then the I/O; when the board is powered off, the sequence should be the opposite, i.e., I/O is first powered off and then the DSP core. The interval between them should be no longer than 1 s. To ensure this requirement, we use DC/DC controller with integrated switch TPS54310PWP devices, and connect the Power Good (PG) pin of 1.4 V regulator to the ENable (EN) pin of 3.3 V regulator. By doing so, the 3.3 V regulator could be only powered on when 1.4 V is valid, therefore the DSP core is powered up before the I/O (Ren *et al*., 2000).

SOFTWARE DESIGN

We use TI's Code Composer Studio (CCS) 2.21 (TI, 2005) as the Integrated Developing Environment (IDE) of DSP software. The DSP VLPR software is composed of 5 task modules: video input task, image processing task, video output task, network task and network initialization task. The software adopted the Reference Framework 5 (RF-5) (Mullanix *et al*., 2003) to integrate these tasks.

After the board initialization is completed, the DSP system enters the DSP/BIOS and let the schedule program of DSP/BIOS manage the 5 tasks (TI, 2002a). The 5 tasks (including an idle task) communicate with each other through the Synchronized COMmunication (SCOM) module of RF-5 as shown in Fig.4. The schedule algorithm is based on tasks' priority. The higher priority the process has, the earlier it will be served. If two or more tasks have the same priority, then the schedule program will adopt First Come First Served (FCFS) algorithm. Here the video input task, video output task and image processing task have the highest priority (=5), which are run in FCFS manner until they all wait for appropriate SCOM messages. Then the DSP core runs network task (priority: 1). If all the 4 tasks above are waiting, the idle process (lowest priority: 0) will be scheduled to run.

Video input

Video input is the first step of the VLPR. This step accomplishes the capture of video and the transformation of image format in the color space.

Video capture

As mentioned in Section 3, SAA7115H transforms the analog video into digital. Here we use the TMS320DM642 Video Port Mini-Driver (TI, 2003a) to control the SAA7115H device. The function *FVID_exchange* of the driver gets the new frame of video from the input device. Since its precision is good enough, NTSC640 mode is chosen to facilitate the following processes. Now we get the image frames through the video capture driver.

Transformation in color space

The image obtained from the video capture procedure is formatted in YUV422 while the recognition algorithm needs the RGB format image. Below is the formula to transform image from YUV to RGB space (Shao and Han, 2004).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0 \end{bmatrix} \cdot \left(\begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right). \quad (1)$$

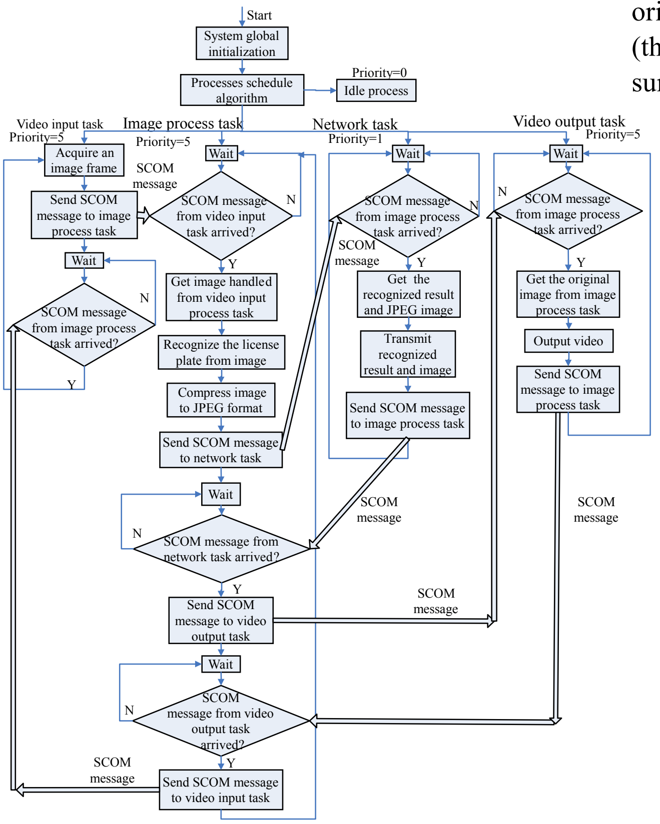

Fig.4 The tasks system under the management of DSP/BIOS schedule program

Image process

After the RGB format image is obtained, we could go on to recognize the possibly existing license plate. The DSP VLPR algorithm is based on the VLPR algorithm realized on PC.

Character location

The image (640×480) is first turned into a gray scale image and shrunk to a small one (320×240) to speed up the computation. The horizontal gradient image is obtained by calculating the luminance difference between horizontal neighbor pixels. Then the gray scale histogram of the resulting image is computed to get the threshold of the binary procedure. The binarization techniques can be roughly categorized as global thresholding and local thresholding. For LPR, where non-uniform illumination is usually not an issue, global thresholding is commonly used for its simplicity and high speed (Yang *et al.*, 2006b). The original threshold value is the first gray scale value (the searching direction is from 255 to 0) of which the summed pixels are over 2500. The pixels whose gray

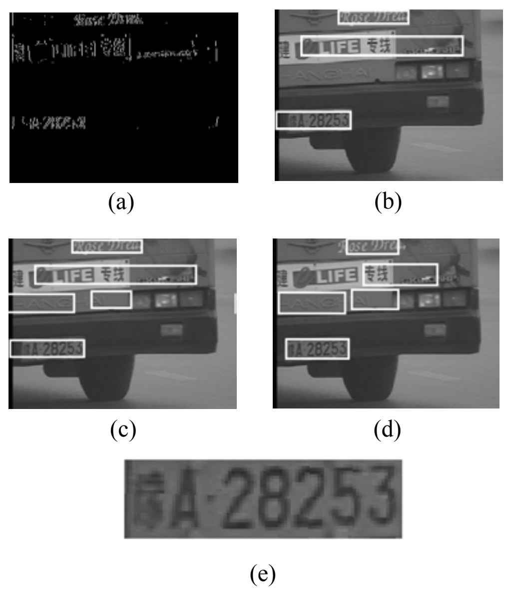

Fig.5 Plate location. (a) The binary horizontal gradient image; (b) Coarse plate location based on region gradient variance; (c) Locations based on saturation; (d) Accurate locations; (e) Final plate region after eliminating fake locations

scale values above the threshold are edge pixels, others are background pixels. Thus we got the binary image as shown in Fig.5a. Scanning the binary image in row, we assume that the rows containing moderate numbers (more than 7 and less than 25) of closely located (less than 20 pixels apart) edges are possible plate rows. Fill the plate rows with the foreground color (white) from the first edge column to the last edge column. Fill other rows with the background color (black). Then after 4-neighborhood traversals of this image, we got the candidate plate regions. These regions are filtered by the constraints that point number must be larger than 50, row number more than 8, and column number more than 20. If no plate region left, decrease the threshold to 70% and repeat the above procedure until the plate region is found or threshold is as little as 5. The coarse plate location of our example is shown in Fig.5b.

The additional candidate regions are located according to the saturation and connection features of the image. The input RGB color image is shrunk to 160×120 and transformed to HIS (Hue, Intensity, Saturation) space (Chang *et al*., 2004). We extract the saturation image from the result and calculate the gray scale histogram of it. From this histogram, we calculate the threshold from the average gray value of maximum point with the last point appearing before zero point. If the average is larger than 30, we take 30 as the threshold, and 10 if the average value is less than 10. With this threshold, we got the binary image from the saturation image. From this binary image we make 4-neighborhood traversals and get some candidate plate regions. The regions out of range (width∈[12, 62], height∈[10, 20], and width height ratio∈[1, 4.5], unit of height and width: pixel) are discarded.

Both above results are merged (shown in Fig.5c). Finally, the accurate borders of candidate regions are extracted by using the saturation projection on vertical and horizontal sides together with the gradient feature (Fig.5d). Fake plate regions are eliminated according to the features of width-height ratio, the noise point numbers, the horizontal and vertical scan line characteristics and connections, etc. If the width-height ratio is out of the range [1.4, 2.1], we think the plate region is fake and throw it away. Scan the plate binary image horizontally, calculate the white segment number, if there are more than 6 lines of which the foreground segments' account is out of range [7, 20], we decide it is a fake plate, too; if the distance between the first legal (with white segments number inside [7, 20]) line and the last legal line are less than 8 lines, it is also a fake plate; if there are more than 3 lines which include long noise segment (more than 30 points) around centre position (inside 4 lines to central line) of plate, it is a fake plate; if the appearance of neighbor with white segments number differ by more than 5 for more than 6 times, it is a fake plate. Furthermore, we have an 8-neighborhood traversed binary plate image to find all the connection areas. Too small areas (less than half plate height) are filtered as noise. Finally, if the overall connection areas are accounted less than 4, the plate is fake; if outside rectangles of connection areas overlap with each other, the plate is also a fake one. After all these processes, we finally get a real plate region (Fig.5e).

We verified that the background was black and foreground was white in the binary image of plate region (if the situation is opposite, just reverse it). Then the plate color can be determined by calculating the background RGB, HIS values of plate region in the original color image. First we transform the RGB color image to HIS space. If we count the average hue, saturation, intensity, red, blue and green values of the plate background respectively, we can classify the plate color. For example, when the average hue value of plate background is between 70 and 270, the plate is blue or black. Further information such as average red, blue, green and intensity values of plate background would be used to distinguish between the above two plate colors. We divide the plate color into 5 classes: blue, white, black, yellow and unknown color. The above process is based on experience.

Character segmentation

The number of characters on the normal license plate in mainland China is fixed, i.e. 7 characters: 1 Chinese character plus 1 English letter and 5 numerals; 1 Chinese character plus 2 English letters and 4 numerals; 1 Chinese character plus 3 English letters and 3 numerals; etc. (see Fig.6).

The Chinese character indicates province or municipality where the vehicle is registered. For instance, in the wide region of Zhejiang Province, the first word of the plate is originally set to "浙". The width of the characters is the same, which can be used in character segmentation. The plate image is binarized, rotated and denoised before segmentation. This procedure is mainly based on the projection of white pixels sum on horizontal side. The plate is split at the least point (Fig.7a). Adjust the split units into 7 characters. Then denoise the plate again and relocate the border of characters to get ready for recognition. This denoising procedure is just for segmention, so the noise inside the character area still exists (Fig.7b).

Fig.6 Examples of license plate in Mainland China

(a) (b)

Fig.7 Character segmentation results. (a) The rough split character; (b) Denoised segmentation

Segmentation is performed for the classification. At the same time, segmentation should be done in cooperation with the classification, so that some segmentation mistakes could be corrected, and success of segmentation is reinforced.

Chinese character recognition

The multi-resolution property of wavelet is applied to extract the features of Chinese characters, and a method that directly extracts features from gray scale image is proposed by Pan *et al*.(2003).

If $f(x,y)$ represents the gray scale image, the corresponding polar coordinates expression is $f(r,\theta)$, the moment invariant can be expressed as:

$$F_{pq}=\iint f(r,\theta)g_p(r)\mathrm{e}^{\mathrm{j}q\theta}r\mathrm{d}r\mathrm{d}\theta, \tag{2}$$

where F_{pq} represents the pqth moment, $g_p(r)$ is a function on r which controls different moment parameters. It can be proved that the module of F_{pq} is invariant to image rotation. Let

$$S_q(r)=\int f(r,\theta)\mathrm{e}^{\mathrm{j}q\theta}\mathrm{d}\theta, \tag{3}$$

we get

$$F_{pq}=\int\int S_q(r)g_p(r)r\mathrm{d}r\mathrm{d}\theta. \tag{4}$$

Let $g_p(r)$ be the wavelet base function:

$$\psi_{a,b}(r)=\frac{1}{\sqrt{a}}\psi\left(\frac{r-b}{a}\right), \tag{5}$$

then we get the wavelet moments. To simplify the computation, here the parent wavelet is in the form of cubic B-spline Gaussian approximation shown below:

$$\psi(r)=\frac{4\beta^{n+1}}{\sqrt{2\pi(n+1)}}\sigma_w\cos[2\pi f_0(2r-1)]\exp\left(-\frac{(2r-1)^2}{2\sigma_w^2(n+1)}\right), \tag{6}$$

where n is the order of B-spline, f_0 is the modulation coefficient, $\sigma_w^{\ 2}$ is the deviation. If we set n=3, β=0.697066, f_0=0.409177, $\sigma_w^{\ 2}$=0.51145, then the function can approximate the B-spline so well that the approximation error is less than 3%.

The values of a, b are:

$$\begin{cases} a=m/2, & m=0,1,2,3, \\ b=n/2^{m+1}, & n=0,1,...,2^{m-1}. \end{cases} \tag{7}$$

Turn the parent wavelet into a general form:

$$\psi_{m,n}(r)=2^{m/2}\psi(2^m r-n/2), \tag{8}$$

then the wavelet moment invariants can be computed as follows:

$$\left\|F_{m,n,q}\right\|=\left\|\int S_q(r)\psi_{m,n}r\mathrm{d}r\right\|, \tag{9}$$

where m=0,1,2,3; n=0,1,…,2^{m-1}; q=0,1,2,3. We can get 136 dimensions of wavelet moment features.

These features together with the features obtained from OCR (optical character recognition) system are selected according to their contribution to recognition rate. In OCR system, zone density is a kind of global feature in common use. We calculated zone (3×3 in size) density of white pixels on the character binary image and got a 66-dimension feature (since the image size is 32×16, some of the zones are not in size 3×3). Eighty-one dimensions of features are selected out of 136 dimensions of wavelet moment features and 66 dimensions of zone density features according to their contribution to the recognition rate.

These features are input to a back propagation neural network (BPNN) (shown in Fig.8). A neural network (NN) is an artificial network model which emulates the cerebral neural network (Nukano *et al*., 2004). BPNN is based on gradient descent method which minimizes the sum of the squared errors between the actual and the desired output values (Wen *et al*., 2000). The BPNN is a three-layer feed forward neural network. As shown in Fig.8, this BPNN has 81 input units (equal to the input dimension of features), 50 hidden units and 34 output units (equal to the number of provinces and municipalities in China). A sigmoid function is used as the output function of hidden neuron units with the output function of the output neuron units being a linear function. The training error of the BPNN is 0.01 and it learns 10000 times in maximum. The BP learning algorithm is described below: a random value between 1 and 60 and a random value for index of Chinese character are generated and combined to be a file index for a Chinese character image sample. The features are extracted from this sample and input to the neural network for training until the training error is less than the given error (0.01). If this procedure is completed, the neural network is thought to have learned successfully once and turns to learn the next sample; otherwise, if a sample (ignoring the first 100 samples) is learned more than 300 times and the training error is still more than the given error, the network is thought unable to converge and the learning is stopped. Thus the Chinese characters can be classified and recognized. For detailed procedure of BP algorithm, please refer to (Zhang *et al*., 2006).

The plate type can be decided according to the plate color and the recognized Chinese character.

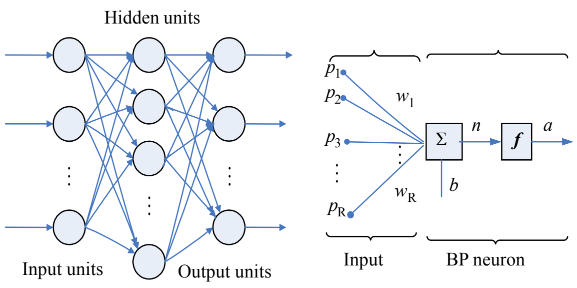

Fig.8 The cell models. (a) Back propagation neural network (BPNN); (b) BPNN cell

Recognition of alphabetic and digital characters

The structure of alphabetic and digital characters is simple compared with that of Chinese characters. Letters and numbers can be recognized from a binary image. The located and segmented image is transformed into a binary image first. A moment vector can be obtained from the binary image through linear moment transformation. A binary image which contains object can be described as:

$$f(i,j)=\begin{cases}1, & (i,j)\in \text{Object},\\ 0, & (i,j)\notin \text{Object}.\end{cases} \tag{10}$$

The referenced origin $C_0=(i_0,j_0)$ on the image is defined as:

$$i_0=\min(i{:}\exists j,\ \text{yields}\ f(i,j)=1,\ j=0,1,\ldots,W-1;\ i=0,1,\ldots,D-1), \tag{11}$$

$$j_0=\min(j{:}\exists i,\ \text{yields}\ f(i,j)=1,\ i=0,1,\ldots,D-1;\ j=0,1,\ldots,W-1), \tag{12}$$

where W, D are the width and height of the image respectively. We can define linear moment vector of row and column as $\boldsymbol{M}_{\text{row}}$ and $\boldsymbol{M}_{\text{col}}$ respectively in the image. Take $\boldsymbol{M}_{\text{row}}$ for example, if the number of elements of $\boldsymbol{M}_{\text{row}}$ is N, the kth element is expressed as $\boldsymbol{M}_{\text{row}}[k]$ ($N\le D$, the scan line of the kth element of $\boldsymbol{M}_{\text{row}}$ is line $[D\cdot k/N+0.5]$ of image, where [] is the operator for conversion to integer). There are black segments B_{kl} and white segments W_{kl}. B_{kl} represents a segment on the object, and W_{kl} represents a segment on the background, where $1\le l\le N_k$, N_k is the sum of black segments on the scan line of the kth element. Let L_{kl} be the length of B_{kl}, C_{kl} be the center of B_{kl}, then we get

$$\boldsymbol{M}_{\text{row}}[k]=\sum_{l=1}^{N_k}(C_{kl}-C_0)L_{kl}. \tag{13}$$

We can see that the moment value is proportional to the sum, length and location distance of black segments. Mark the row scale coefficient as S, the central point after moving as $\tilde{C}_{kl}$, the origin of object on the image as $\tilde{C}_0$, then

$$\tilde{C}_{kl}-\tilde{C}_0=S(C_{kl}-C_0), \tag{14}$$

$$\tilde{L}_{kl}=SL_{kl}. \tag{15}$$

So

$$\tilde{\boldsymbol{M}}_{\text{row}}[k]=\sum_{l=1}^{N_k}(\tilde{C}_{kl}-\tilde{C}_0)\tilde{L}_{kl}=S^2\sum_{l=1}^{N_k}(C_{kl}-C_0)L_{kl}$$
$$=S^2\boldsymbol{M}_{\text{row}}[k]. \quad (16)$$

Define the linear moment as

$$m_{\text{row}}[k]=\boldsymbol{M}_{\text{row}}[k]\Big/\sum_{k=1}^{N_k}\boldsymbol{M}_{\text{row}}[k], \quad (17)$$

then

$$\tilde{m}_{\text{row}}[k]=\frac{\tilde{\boldsymbol{M}}_{\text{row}}[k]}{\sum_{k=1}^{N_k}\tilde{\boldsymbol{M}}_{\text{row}}[k]}=\frac{S^2\boldsymbol{M}_{\text{row}}[k]}{\sum_{k=1}^{N_k}S^2\boldsymbol{M}_{\text{row}}[k]}=m_{\text{row}}[k]. \quad (18)$$

Thus the linear moments $m_{\text{row}}[k]$ are scale-invariable and movement invariable (Shen and Li, 2000). We can extract features by defining statistic variations after the 1D wavelet transformation of the moment vector. At the same time, we got other 4 features that reflect the characters' direction of strokes. Human eyes recognize characters depending on the characters structure, namely the stroke features of characters such as horizontal, vertical and diagonal strokes. In order to extract 4 directions' edge, the binary image is wavelet parsed. Finally, the unitary feature-vector is input into a BPNN to be classified and recognized.

We adopted a three-layer BPNN (shown in Fig.8) to classify the alphabetic and digital characters. The neuron of hidden units is a sigmoid function. There are 8 units in the input layer for accepting different features, 36 units in the output layer for classified results (10 for digital characters and 26 for alphabetic characters), and 22 hidden units in the middle layer. The training error of the BPNN is 0.01 and it learns 10000 times in maximum. The BP learning algorithm is similar to that used in Chinese characters' classification.

Image compression

If we recognize a license plate from a certain image, it is necessary to store this image to be referred to afterward. The resolution of our image is 640×480, for RGB format its size would be as large as 900 kB (640×480×3=900 kB). If we store the image in RGB format, it would occupy considerable memory space. Furthermore, the image would be sent to the host computer through network. So under the condition that the image is clear enough, it is better for the image to be smaller in size. Therefore the image should be compressed as far as possible. JPEG standard is adopted in our system for its high compression ratio for static images. JPEG supports a wide variety of applications and it mainly compresses the image through ignoring the high frequency components that the human eyes are less sensible to.

Communication

The compressed image together with the recognition result is transmitted to host computer through network. There is a 10/100 Mbps EMAC among TMS320DM642's peripheral interfaces that can connect to TCP/IP net freely (TI, 2003b). This network interface should be initialized beforehand. We use API from TI's Network Developer's Kit to initialize the IP address, subnet mask, gateway, host name and DNS server of DSP so that the DSP could connect to the network (TI, 2001). If necessary, the DSP can also use DHCP (dynamic host configuration protocol) to locate and contact the host computer, which returns the appropriate information of network address allocation and parameter configuration.

After the connection is established, every time the DSP VLPR software recognizes the plate, it will pack up information of the recognized result (the type, color and number of the plate) and the size of compressed image, and send the packet to the host computer. The compressed image is sent subsequently. The packet size is 32 bytes. Fig.9 shows the format of message packet.

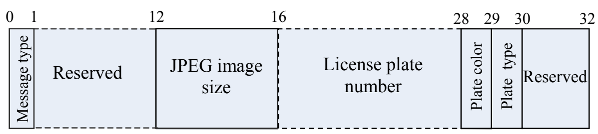

Fig.9 Format of the message packet

Video output

In order to realize real-time surveillance, the video acquired from the camera is played back on the monitor. The digital signal captured from the video is restored into analog signal so that it could be played on the monitor. We use SAA7105H to encode the digital signal and the display driver program to control the device.

Software optimization (TI, 2002b)

There are considerable distinctions between PC and DSP. Lower clock rate and less memory space make it necessary for DSP to optimize the software. After optimization, the DSP software would run faster and be more efficient so as to meet the requirement of applications. The optimization techniques include: to increase parallelity by eliminating memory dependency and loop unrolling (Li *et al*., 2003); to decrease unnecessary floating-point, multiplication and division operations; to use TI's intrinsic function library and IMGLIB (image process library).

Table 1 gives a comparison between the execution cycles before and after optimization. The execution cycles of all the modules are obviously decreased after optimization and surely the execution rate of DSP is sped up considerably.

Table 1 Execution cycles before and after optimization

Modules	Before optimization	After optimization
Video input	43 550 224.94	16 035 467.41
Recognition	559 539 196.28	67 935 995.67
Image compression	1 292 423.50	1 182 613.55
Network communication	13 163 033.48	8 340 333.57
Video output	1 487 646.86	1 469 164.26

SOFTWARE DESIGN OF HOST COMPUTER

There are no display or storage devices for the recognition results and images on the DSP board; results and images are transmitted to the host computer through Ethernet. We designed a PC software to communicate with DSPs, to browse and manage the information from several DSP boards side by side. This software is realized with C# based on Microsoft Net Framework 2.0 which can debug and compile with VS2005 (Jeffrey and Chris, 2002).

This PC software communicates with DSP asynchronously to achieve instant response, i.e., when the socket receives data from DSP VLPR software, it calls the callback function automatically instead of asking the socket continually to find out if there are any data received. The recognition results organized in an XML file is structured below:


```
<RecognizeCamera IPAddress="10.214.50.134" Result-
    sCount="1346">
    ...
    <RecognizeResult>
      <Result>粤 A13302</Result>
      <Color>Black font yellow plate</Color>
      <Type>Normal type</Type>
      <DateTime>2006-9-18 16:39:58</DateTime>
        <ImagePath>D:\test\10.214.50.134\2006-9-18\
          [2006-9-18 16.39.58.375]粤 A13302.jpg
        </ImagePath>
    </RecognizeResult>
    ...
</RecognizeCamera>
```


EXPERIMENTAL RESULTS

The proposed system was tested repeatedly. Some experimental results are shown in Table 2.

Table 2 Experimental results of DSP VLPR system

Tests	Plates	Correct Recog.	Recog. ratio (%)	Total time (s)	Mean time (s)
1	171	167	97.7	64.232	0.376
2	226	224	99.1	76.031	0.336
3	249	236	94.8	91.594	0.368

According to this statistics, the system can recognize 160 vehicles in 1 min and has an accuracy of around 95%, which can satisfy the requirement of applications very well.

Fig.10 is the software GUI on the host computer. The left column is the list of IP addresses of DSPs connected to the host computer. The upper part of the right column lists the selected DSP's recognized results which include plate number, plate type, plate color, recognized time, etc. The lower region of the right column displays the compressed image of selected record of the recognized results.

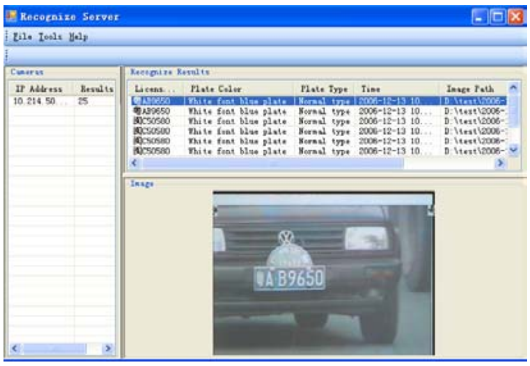

Fig.10 The GUI of the host computer software

CONCLUSION

This paper introduces a novel scheme of VLPR system based on DM642. This system can be embedded into ITS feasibly and freely. Experiments showed that our DSP VLPR system performs efficiently on local license plates. It can recognize a vehicle license plate in less than 0.4 s, which is very suitable for real-time applications. Compared to VLPR realized in PCs, this system has the advantages of smaller size, faster processing, lighter weight, lower cost, easier operation, etc.

However, there is still some further research to do. For example, the system cannot work well when illumination is not very good. So we plan to equip our system with automatic illumination-compensation device to make the system adapt to all kinds of environments. Furthermore, we are thinking about realizing more additional functions such as velocity measurement, traffic statistics, etc. on the DSP board through video processing.

References

Adorni, G., Bergenti, F., Cagnoni, S., 1998. Vehicle License Plate Recognition by Means of Cellular Automata. Proc. IEEE Int. Conf. Intelligent Vehicles, p.689-693.

Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Loumos, V., Kayafas, E., 2006. A License plate-recognition algorithm for intelligent transportation system applications. *IEEE Trans. on Intell. Transport. Syst.*, **7**(3):377-392. [doi:10.1109/TITS.2006.880641]

Bai, H.L., Liu, C.P., 2004. A Hybrid License Plate Extraction Method Based on Edge Statistics and Morphology. Proc. ICPR, p.831-834. [doi:10.1109/ICPR.2004.1334387]

Barroso, J., Bulas, C., Dagless, E.L., 1997. Real Time Number Plate Reading. 4th IFAC Workshop on Algorithms and Architectures for Real-time Control. Portugal.

Brugge, M.H.T., Stevens, J.H., Nijhuis, J.A.G., Spaanenburg, L., 1998. License Plate Recognition Using DTCNNs. Proc. 5th IEEE Int. Workshop on Cellular Neural Networks and Their Applications, p.212-217. [doi:10.1109/CNNA.1998.685366]

Chang, S.L., Chen, L.S., Chung, Y.C., Chen, S.W., 2004. Automatic license plate recognition. *IEEE Trans. on Intell. Transport. Syst.*, **5**(1):42-53. [doi:10.1109/TITS.2004.825086]

Chil, X.J., Dong, J.Y., Liu, A.H., Zhou, H.Y., 2006. A Simple Method for Chinese License Plate Recognition Based on Support Vector Machine. Int. Conf. on Communications, Circuits and Systems Processing, **3**:2141-2145. [doi:10.1109/ICCCAS.2006.284922]

Cho, B.H., Jung, S.H., 1998. Non Feature-based Vehicle Plate Recognition System Using Neural Network. Proc. ITC-CSCC Int. Conf., **2**:1065-1068.

Choudhury, A.R., Wael, B., Ahmad, R., 2003. A Real Time Vehicle's License Plate Recognition System. Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance, p.163-166. [doi:10.1109/AVSS.2003.1217917]

Cui, Y.T., Huang, Q., 1997. Character Extraction of License Plates from Video. Proc. IEEE Conf. Computer Vision and Pattern Recognition, p.502-507. [doi:10.1109/CVPR.1997.609372]

Jeffrey, P.M., Chris, K., 2002. C# Developer's Guide to ASP.NET, XML, and ADO.NET. Addison Wesley.

Kim, S.K., Kim, D.W., Kim, H.J., 1996. A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation. Proc. Int. Conf. Image Processing, **2**:661-664. [doi:10.1109/ICIP.1996.560964]

Kim, K.K., Kim, K.I., Kim, J.B., Kim, H.J., 2000. Learning-based Approach for License Plate Recognition. Proc. IEEE Signal Processing Society Workshop, **2**:614-623. [doi:10.1109/NNSP.2000.890140]

Lambers, H., 2001. SAA7115_datasheet. Philips Semiconductors.

Li, F.H., Wang, F., He, P.K., 2003. The Principle and Applications of TMS320C6000 Series DSPs. Publishing House of Electronics Industry, Beijing (in Chinese).

Mullanix, T., Magdic, D., Wan, V., Lee, B., Cruickshank, B., Campbell, A., DeGraw, Y., 2003. Reference Frameworks for eXpress DSP Software: RF5, an Extensive, High-density System. Texas Instruments.

Naito, T., Tsukada, T., Yamada, K., Kozuka, K., Yamamoto, S., 2000. Robust license-plate recognition method for passing vehicles under outside environment. *IEEE Trans. on Vehicular Technol.*, **49**(6):2309-2319. [doi:10.1109/25.901900]

Nijhuis, J.A.G., Brugge, M.H.T., Helmholt, K.A., Pluim, J.P.W., Spaanenburg, L., Venema, R.S., Westenberg, M.A., 1995. Car License Plate Recognition with Neural Networks and Fuzzy Logic. Proc. IEEE Int. Conf. on Neural Networks, **5**:2232-2236. [doi:10.1109/ICNN.1995.487708]

Nukano, T., Fukumi, M., Khalid, M., 2004. Vehicle License Plate Character Recognition by Neural Networks. Proc. Int. Symp. on Intelligent Signal Processing and Communication Systems, p.771-775. [doi:10.1109/ISPACS.2004.1439164]

Pan, X., Ye, X.Z., Zhang, S.Y., 2003. The car plate Chinese character feature extraction based on wavelet. *J. Image Graph.*, **8**(10):1218-1222 (in Chinese).

Parisi, R., Claudio, E.D.D., Lucarelli, G., Orlandi, G., 1998. Car Plate Recognition by Neural Networks and Image Processing. Proc. IEEE Int. Symp. Circuits and Systems, **3**:192-198. [doi:10.1109/ISCAS.1998.703970]

Ren, L.X., Ma, S.F., Li, F.H., 2000. The Principle and Applications of TMS320C6000 Series DSPs. Publishing House of Electronics Industry, Beijing, p.239-245 (in Chinese).

Shao, D., Han, J.W., 2004. Inter-transformation between YUV and RGB. *J. Changchun Univ.*, (04):53-55 (in Chinese).

Shen, H.L., Li, Z.N., 2000. A Study of Number and Letter Character Recognition Based on Moments and Wavelet Transform. *J. Image Graph.*, **5(A)**(03):249-252 (in Chinese).

TI, 2001. TMS320C6000 TCP/IP Network Developer's Kit (NDK) Programmer's Reference Guide. SPRU524A.

TI, 2002a. TMS320C6000 Optimizing C Compiler Tutorial. Texas Instruments Incorporated.

TI, 2002b. TMS320 DSP/BIOS User's Guide. Literature Number: SPRU423B.

TI, 2003a. The TMS320DM642 Video Port Mini-Driver. Literature Number: SPRA918A, p.1-34.

TI, 2003b. TMS320C6000 DSP Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module Reference Guide.

TI, 2005. TMS320DM642 Video/Imaging Fixed-point Digital Signal Processor. SPRS200J.

Wen, J.W., Zhao, J.L., Luo, S.W., Han, Z., 2000. The Improvements of BP Neural Network Learning Algorithm Proc. 5th Int. Conf. on Signal Processing, **3**:1647-1649. [doi:10.1109/ICOSP.2000.893417]

Wu, Q., Zhang, H.F., Jia, W.J., He, X.J., Yang, J., Hintz, T., 2006. Car Plate Detection Using Cascaded Tree-style Learner Based on Hybrid Object Features. Proc. IEEE Int. Conf. on Video and Signal Based Surveillance, p.15. [doi:10.1109/AVSS.2006.30]

Yang, F., Ma, Z., Xie, M., 2006a. A Novel Binarization Approach for License Plate. 1st IEEE Conf. on Industrial Electronics and Applications, p.1-4. [doi:10.1109/ICIEA.2006.257232]

Yang, F., Ma, Z., Xie, M., 2006b. A Novel Approach for License Plate Character Segmentation. 1st IEEE Conf. on Industrial Electronics and Applications, p.1-6. [doi:10.1109/ICIEA.2006.257234]

Zhang, J., Fan, X.P., Huang, C.L., 2006. Research on Characters Segmentation and Characters Recognition in Intelligent License Plate Recognition System. Proc. 25th Chinese Control Conf., p.1753-1755. [doi:10.1109/CHICC.2006.280847]