A new design on image processing scheme for smart car

Yixu Chen
School of Control Science and Engineering, Shandong University, Shandong, Jinan 250061, China
cyx709671676@gmail.com

Abstract. As a product of comprehensive discipline, smart car involve artificial intelligence, computer vision, mechanical design, control science and vehicle engineering, etc. In this study, we take the NXP smart car competition as the background and put forward a new image processing scheme for smart car. The purpose of the image processing is to identify the different types of track from continuously transformed video images, to filter out noise and interference in the images, to identify the obstacles and extract the available left and right edges of the track. The proposed new scheme can effectively extract and classify the image features of different road types and give corresponding image processing algorithms. Experiments show that this scheme can efficiently deal with the probe images and provide accurate image information for smart car. Thus, the proposed scheme has a high practical value in applications, such as smart car contests and self-driving car industry.

1. Introduction

With the development of the automobile industry, there are more and more vehicles on the road, environmental pollution, traffic safety and other issues are becoming increasingly serious. In order to solve these problems, people began to research the safety and intelligent of cars. Nowadays, control of smart car has become a hot research area [1]. The smart car acquires real-time information of the external environment through sensors, uses prior knowledge to carry out the path planning and moves to the destination without any intervention. Intelligent car system is a complex system, which integrates the information collection and processing, intelligent control, mechanical structure and transmission as a whole.

Path identification is the basis for the realization of autonomous running of smart car along the track, accessing a wide range and accurate track information is the key to improve the speed of smart car [2-4]. Based on this, we analyze the image processing method for smart car. The purpose of intelligent car image processing is to identify the different types of track from the continuously transformed video images, to filter out the noise and interference in the image, to identify the obstacles, and to extract the available left and right edges of the track. There are several possible types of track, including straight, corner, cross, roundabout, obstacle, etc. Therefore, the smart car image processing needs to solve the following problems: extract the useful edges of the collected black and white track images, identify the different track types, extract the front obstacle position, and search for the viable main road edge [5].

Due to the limitation of the computational intelligence of the embedded system, the algorithm should have the characteristics of real-time and lightweight, and many existing bitmap processing algorithms cannot meet the demand. Therefore, in the exploration of different algorithms, we also design a number of targeted algorithms. In order to simplify the image preprocess procedure, we directly use a digital camera which acquires gray-scale image and then turn it into binary image according to the average...
brightness of the whole image using hardware.

2. Image preprocessing

2.1. Bitmap preprocessing
Because of various facts, there are different noises and interference in image, and they will affect the follow-up image processing, so we need the "erosion" operation in the pretreatment process to remove them.

The camera will automatically adjust the exposure as the brightness of the entire image changes, resulting in excessive exposure when the car position is relatively partial. Sometimes, the direct sunlight causes ground reflection, making the edge of the original black image intermittent. The erosion operation in the basic morphological operation of binary images can be expressed by the following formula:

$$A \Theta B = \{z \mid (B_z \subseteq A)\}$$  \hspace{1cm} (1)

The meaning of Equation (1) is that the structural element $B$ is moved over the entire $Z^2$ plane, and $B$ is able to be completely contained in $A$ when the origin point of $B$ is translated to point $z$, then the set of all such $z$-points is the eroded image of $B$ to $A$.

Using erosion, dilation, open and close operation with the structural elements of 3*3 squares on the collected images (the black noise points in the image are artificially added), the result is shown in Figure 1. We can see that only the erosion can make the original broken edge be closed, the image quality is improved significantly. The cost of the corrosion operation is that the left and right sides of the white track are narrowed by one pixel width, which has little effect on the subsequent processing and the decision of the car model, so it can be ignored, however, the black noise points are enlarged by erosion which will be eliminated by following filter.

![Figure 1](source.png)

2.2. Search connectivity area, convert bitmap to edge graph
According to the law of the projection perspective, the size of the white area of the track in the image is gradually reduced from near to far, so it is necessary to search for a track with a white connectivity area. We can use the traditional methods of region growing or boundary trace to solve this problem. However, region growing and boundary trace do not meet the requirements of embedded systems both in terms of time and search results. Therefore, we propose a new cross-edge search algorithm that based on row search which means any edge in the next line crossing with the edge in the previous line is treated as a
valid edge. This algorithm can simultaneously complete the search of the connected area and the extraction of the boundary information, and compress the bitmap into edge graph. Because the images obtained by a digital camera are bitmaps, and the embedded system computing intelligence is limited, so the digital bitmap processing speed is decreased obviously. However, the final information we need is the left and right edges of the track, which is relatively simple (the number of black and white edges of a line is generally no more than 3). Thus, we can extract the edge information to reduce the amount of information to facilitate the embedded system processing.

Figure 2 shows an obstacle in the middle of a straight road. We sample the four lines of L1, L2, L3, L4, use x-y to represent a pair of edges, where x is the left edge, y is the right edge.

The steps of the cross-edge search algorithm based on row search are as follows:

1) Assume that the datum point is at the position of 55 in the middle of the first line of the image. Based on a determined search datum, the search algorithm starts from L1 and finds the first edges of the left and right sides(30 and 80), respectively.

2) Each edge obtained from the previous line is used as a benchmark to search for all edges of the current line that intersect with each other and store them into the array. The cross criterion is $R_n > L_{n-1} \& \& L_n < R_{n-1}$, where $L_n$ and $R_n$ are the coordinates of the left and right edges of the current line, $L_{n-1}$ and $R_{n-1}$ are the coordinates of the left and right edges of the previous line.

3) Repeat step 2) until all rows are searched.

The results of this algorithm are shown in Table 1 and Figure 3.

| Table 1 The results of the cross edge search algorithm |
|-----------------|-----------------|
| L4              | 30-80           |
| L3              | 30-50 60-80     |
| L2              | 30-50 60-80     |
| L1              | 30-80           |

Figure 2 Image example (there is an obstruction in straight road)
In this way, the 120*160 bitmap image taking 19.2KB can be transformed into an edge graph (which is calculated according to the reserved edge number of 5) which approximately taking120*5*2 bytes, that is 1.2Kb bytes, so the edge graph greatly reduces the amount of data.

2.3 Fixed-scale filtering

Fixed-scale filtering uses a filtering method that based on the ratio of the noise area, filtering out all black noise with a noise area ratio that less than the threshold, so that it can effectively remove the interference and don’t affect the edge of the original image.

The area ratio is calculated according to the following formula:

$$\text{area ratio} = \frac{\text{track area} - \text{white valid area}}{\text{track area}} = \frac{\text{black stain area}}{\text{track area}}$$

(2)

In equation (2), the track area is the number of pixels of the track line (from the first left edge to the last right edge), similarly, the white valid area is the number of white pixels. The area threshold is usually set to 7%, so that most of the possible stains can be filtered without affecting the original image.

As can be seen from Figure 4, the effect of this targeted filtering is better. This operation should be performed after the cross edge algorithm because the process of edge graph is faster than bitmap.

2.4. The continuity search of the edge

The image data obtained after the cross-edge search and fixed-scale filtering is the track data in Table 1, and the data of the left and right edges of the image are stored in two arrays. The main purpose of the edge continuity search is to rearrange the two arrays in terms of the logic of the edge continuity and delete the edges of all image boundaries, which are limited by the camera's viewing angle. As the last step of the preprocessing, the data format obtained by this algorithm is continuous segment, which is very convenient for subsequent feature extraction.

The $n$-th column in the array obtained by the cross edge search represents the abscissa of the $n$-th edge of the current row. After a continuous search, you need to distinguish each successive edge, the $n$-th column in the array represents the abscissa of the $n$-th consecutive edge (from the bottom to the top as is shown in Figure 5). Obviously, not every continuous edge has a valid data in each row. If there is no point in this position, filling it with zero (zero means invalid). The coordinates of the image starting from 1. The edge of the image boundary is also filled with zero (zero also means invalid). The numbers in Figure 5 mean abscissa or column of the result array.
3. Feature extraction and classified recognition

The features of images are mainly the number and position of the inflection point, the position of the invalid cross line, the position of the fork road, the position of the obstacle, and so on. Therefore, we need to design different algorithms to extract these features, determine the types of track according to different combined conditions of different features. The accuracy of classification directly affects the results of the image processing. For the fork road, cross bend and roundabout, we should design the interpolation algorithm according to the demand, and give complete edges of the main road.

3.1. The extraction of obstacles

Obstacles can affect the recognition of the track, so we need to remove the obstacles, and we need to interpolate the main road to ensure that the smart car can bypass the obstacles. The feature of obstacle is that there is a fork on the first continuous edge, and then a single branch is restored. Because the cross bends and the roundabout may be similar to the feature of obstacle, the scope of the obstacle identification is limited to the first continuous valid edges (both left and right edges are valid). Only when the forks appear and then merge, it is considered that obstacles exist. When recognizing the obstacle, the position of the obstacle can be recorded by determining the number of rows that the folks appear and merge.

3.2. The search of invalid cross line and the recognition of cross

Invalid line means that the left and right continuity arrays are all zero. The invalid cross line, as the name suggests, refers to that both the left and right sides are invalid lines, and they are crossed. The invalid cross line means that the image has no edge, and the pixels of that line are all white or all black. We record the position and number of invalid line for each segment, and combine with subsequent features to identify different types of track.

As shown in Figure 6, the left side of the pink line represents the left invalid line, and the right side of the blue one represents the right invalid line. The horizontal cross paths can be determined when invalid lines intersect.

Figure 5  continuity search example (the numbers are abscissa in array)

Figure 6  The search of the invalid cross line
3.3. The search of inflection point
The inflection point is the point on which the continuous edge changes direction. In general, the first
derivative of the continuity edge is determined, and then the inflection point is judged by the positive or
negative of the derivative. As shown in Figure 7, there is an inflection point on the left edge. The features
of the inflection point can be combined with an invalid line. For example, when an inflection point is
surrounded by an invalid line on the opposite side, it usually means that there is a fork. Multiple
inflection points may mean the emergence of S-shaped corners.

![Figure 7 The emergence of an inflection point](image1)

3.4. Feature extraction and identification of fork
The fork means that the number of cross-edge in a row suddenly changes from one to more. The number
of lines and location of the fork are important features of the image.

![Figure 8 The emergence of fork](image2)

4. Image processing
On the basis of correct image recognition, we can deal with different classification images, search
upwards from the first edge of the continuity array, and then interpolate the edges according to the needs
and different types of track. The methods of interpolation include the least squares method and the two-
point method. For the simple interpolation using the two-point method, as for the complex ones using
the least squares method. Table 2 summarizes the image processing methods.

| Image types       | The results of image processing | Processing methods                                      |
|-------------------|--------------------------------|--------------------------------------------------------|
| Single branch     | ![Image](image3)              | Use of the first proximal continuous edge directly     |
### 5. Conclusions
In this study, we take the NXP smart car competition as the background and put forward a new image processing scheme for smart car. The proposed new scheme can effectively extract and classify the image features of different road types and give corresponding image processing algorithms. Experiments show that this scheme can efficiently deal with the probe images and provide accurate image information for smart car. It is very effective in the fast conversion of image data, information extraction, noise cancellation and image enhancement. Thus, the proposed scheme has a high practical value in applications, such as smart car contests and self-driving car industry.
References

[1] Thorpe, C., Hebert, M. H., Kanade, T., & Shafer, S. A. (1988). Vision and navigation for the Carnegie-Mellon Navlab. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(3), 362-373.

[2] ZHUO, Q., WANG, J., & WANG, L. (2006). The detection of path parameters based on matrix CCD. EEPW, 4, 27-35.

[3] Wang, Z., & Liu, Y. (2010, April). Design of road tracing navigation control for smart car use CCD sensor. In E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on (Vol. 1, pp. 345-348). IEEE.

[4] LIU, M., WANG, H. J., & LI, Y. K. (2012). Research of image collection of CCD based on smart car. Electronic Design Engineering, 17, 057.

[5] Arshad, N. M., & Razak, N. A. (2012, March). Vision-based detection technique for effective line-tracking autonomous vehicle. In Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on (pp. 441-445). IEEE.