Acta Mathematica Academiae Paedagogicae Nyíregyháziensis
Tomus 14 (1998), 63-69

NEW OPTICAL TRACKING METHODS FOR ROBOT CARS

Attila Fazekas

Debrecen

Abstract. In this paper new methods are proposed for intelligent optical tracking of robot cars, the important tools of CIM (Computer Integrated Manufacturing), which can be an alternative of former tracking methods. These methods, are partly based on the new adaptation of some basic picture processing methods known in the literature (Hough transformation, Walsh transformation), and the method described last is based on a special 2+1D thinning algorithm.

Introduction

The robot car is one of the modern tools of flexible manufacturing. For controlling robot cars two basic methods are applied:

- In one method the robot car can determine its position and orientation from signals provided by transmitters that are placed at some characteristic points of the manufacturing hall, then comparing its actual position to the ground-plan of the hall and the indicated route, it determines the required route. The advantage of this method is its flexibility, the disadvantages are that a complicated optical system is needed for detection, the determination of the position and orientation, as well as the indication of the route is time consuming, and the ground plan of the hall has to be stored in the memory of the car.
- The other, more often used method is that the route of the car is determined by electric wires laid down under the floor, and the car is able to follow it with the help of induction sensors. The advantage of this method is the relatively simple application. The disadvantages are that it is not flexible enough, since modifications of the routes or developing new ones are difficult and expensive, and it is complicated to install it to the existing manufacturing systems.

The authors in [1] proposed a new, optical route controlling method, which means a flexible, simple, and cheap tracking method for robot cars by using neural networks. Applying this, the robot car provided with sensors is able to recognize and follow signs painted on the floor. The most important advantages of the method are the followings: fault tolerance, relative flexibility and that it can easily be established in any working place. In this paper the authors, using the principles of the controlling method of robot cars described in [1], introduce three new and simple position determining methods, and the corresponding results of some simulation experiments.

The tracking model of the robot car

The schematic picture of the robot car can be seen on Fig .1.

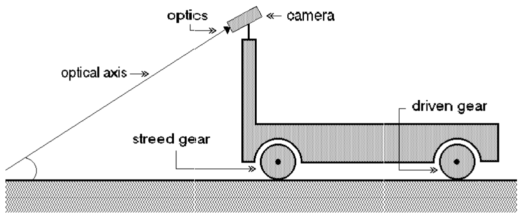

Figure 1.

The main controlling task is to determine "the proper rotation angle" in order to keep the robot car on its route. To do this we can use a sequence of pictures taken by the camera, providing that the robot car proceeds with a constant speed, except for the acceleration and slowing phase. At these phases other methods are used to keep the robot car on the route.

When planning the route, we have to take into consideration the manoeuvrability of the robot car, and the resolution of the optical system. The manoeuvrability — among others is determined by the the maximal rotation angle of the steered gear and the distance between the steered and driven axle — restricts the maximal curvature of the route. In order to determine the line width of the route painted on the floor, we have to consider the resolution of the optical system. In our experiment the line width is determined in such a way that the width of the picture of the route should be approximately the quarter of the width of the picture taken by the camera.

In the following we suppose that the controlling system of the robot car obtains a binary picture of size $n \times m$, where a binary picture is a two-dimensional brightness function $f(x, y)$, where x and y are the coordinates, and the value of f indicates the brightness code of (x, y). In case of a binary picture the two possible brightness code can be either 1 (when the point is called an object point), or 0 (when the point is called a background point). This picture, however, can be represented by a matrix of size $m \times n$, where the value of an element (x, y) is equal to the value of the function at (x, y).

The picture processing part of the controlling task can be easily understood with the help of Fig. 2. This figure shows a part of the route recorded at a given moment. The axis x is the common median axis of the robot car and the camera. To control the robot car we have to know the distance d between the axis x and the line e, and the angle β included by them. Knowing these data, the controlling system gives commands which keep the values of β and d, respectively, below a given limit.

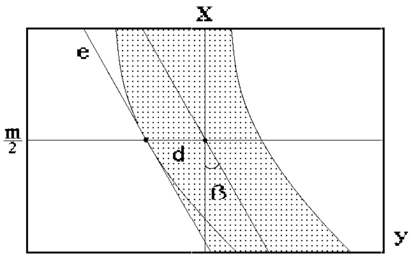

Figure 2.

In the following three chapters methods for determining β and d will be presented.

A method based on the Hough transformation

The Hough transformation, which is a widely used method in digital picture processing, determines the subsets of object points of a given picture on which a common line can be fitted. In the following we are going to give a brief outline of the Hough transformation in the continuous case, for details see [2]. Then we describe its discrete realization used in our method as well.

Let F be a set of object points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ $(n \in \mathbb{N})$. For every point (x_i, y_i), the equation $y_i = ax_i + b$ determines an infinite system of lines, depending on the value of the parameters (a, b). The basic idea is that instead of investigating the above mentioned equation in plane xy, we can investigate the equation $b = -x_i a + y_i$ in plane ab. According to this, let us assign the lines in plane ab determined by the above mentioned relation to the points (x_i, y_i) of F. Thus, if the points (x_i, y_i) belong to a line with parameters (a, b) then the lines assigned to the aforementioned two points intersect each other at a point with coordinates (a, b). Applying these principles, the fitting of k points to a common line is equivalent to the intersecting of k corresponding lines in plane ab, in one point.

In the discrete case we divide the finite part of the ab parameter plane determined by the values of $a_{min}, a_{max}, b_{min}$ and b_{max}, into $n \times m$ pieces — called cells — by using equidistant scale. The values of $a_{min}, a_{max}, b_{min}$ and b_{max} obtained by experiments. Correspond to all cells a non-negative integer which indicates the number of the lines crossing the given cell. Obviously, at the beginning for any cell this value will be 0. After this initializing step, determine the value of b for every element of F and for all dividing points $a_{min} \leq a \leq a_{max}$, then increase the number assigned to the cell with coordinates (a, b), by one.

Having executed the Hough transformation, if the content of the cell with coordinate (a, b) is k, then k points of the set F fits on the line with parameters (a, b).

In our case F is the set of the object points of the binary picture recorded at a given time. We apply the transformation

$$A'(a,b) = \begin{cases} 0 & \text{, if } A(a,b) \leq m \\ A(a,b) & \text{, otherwise.} \end{cases}$$

to the system of cells $A(a, b)$ of size $n \times m$ before further processing. This transformation deletes those cells, the lines represented by which contains too few object points. After this we can determine the smallest and largest local maximum belonging to a in $A'(a, b)$. The coordinates of these cells approximately determine the parameters of the lines e', e'', (a', b') and (a'', b'') in Fig. 3. Relations $a = \frac{a'+a''}{2}$ and $b = \frac{b'+b''}{2}$ then give the parameters of line e. From these, the values of β and d can be easily be calculated.

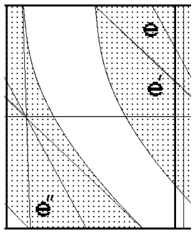

Figure 3.

An application of the Walsh transformation

In this chapter we use the following model. Let a vector, called feature vector, correspond to the picture being processed at the given time. This vector will be compared with to the etalons stored in the database. As a result of the comparison we get an etalon vector which is the closest one among the given vectors. The controlling system uses the values of β and d belonging to this etalon vector as inputs. The calculation of the feature vector is based on the Walsh transformation of the picture, which can successfully be applied in character recognition as well ([3]).

Definition. *When $N = 2^n$ $(n \in \mathbb{N})$, the discrete Walsh transformation of a function $f(x, y)$, denoted by $W(u, v)$ is given by the relation*

$$W(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \prod_{i=0}^{n-1} (-1)^{(b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v))},$$

where $b_k(z)$ is the kth bit in the binary representation of z.

To use the Walsh transformation the picture being processed must be of size $2^n \times 2^n$ $(n \in \mathbb{N})$, which can be obtained by reducing the size of the picture; in our case this reduction yields the size 32×32. After this we calculate the $(W(0,0), W(1,0), \ldots, W(3,0), W(3,1), \ldots, W(3,3))$ feature vector.

2+1D skeletonization

The basis of the third method is the thinning algorithm, which is widely used in picture processing. Thinning is an iterative method in which the original object is substituted by a one pixel wide "median axis", which is a topological equivalent to the original object. This is called the skeleton of the object.

Our method is the following: using the method in [4] we determine the skeleton of a binary picture under processing at the given moment. Then we code this skeleton in the following form:

$$(x_0, y_0), (c_{1,1}, c_{1,2}), \ldots, (c_{n,1}, c_{n,2}),$$

where $n+1$ is the number of the object points forming the skeleton, (x_0, y_0) is the coordinates of that point of the skeleton which can be found in the row with the smallest x_0 index, furthermore $(c_{i,1}, c_{i,2})$ is a code which determines — according to the table mentioned below — the location of the ith point compared to the $(i-1)$th point. Regarding the features of the route, we have to consider neither branches nor loops, which assure us of finding only one point with such feature while tracking the line from point (x_0, y_0) at the ith step. Futhermore, it is also sufficient to deal with the angle range $-90°, \dots, +90°$.

←	↖	↑	↗	→
$(-1,0)$	$(-1,1)$	$(0,1)$	$(1,1)$	$(1,0)$

With the help of the above mentioned set of codes corresponding to the skeleton, the parameters of the line indicated by e in Fig. 3 can be determined in the following form:

$$a = \frac{\sum_{i=1}^{n} c_{i,1}}{\sum_{i=1}^{n} c_{i,2}} \quad , \quad b = -y_0 a.$$

From this, as we could see before, the values of β and d can be easily calculated.

The calculation time can be considerably decreased, if while processing a picture recorded at a given time, we also use the previous one which is already processed. In order to do this, we have to consider the following: If we know the speed of the robot car and the angle of the driven gear while recording the picture, then the projection on axis x and y of the route covered during two consecutive picture shots can be determined. From these the distances indicated by Δx and Δy on Fig. 4 can be obtained.

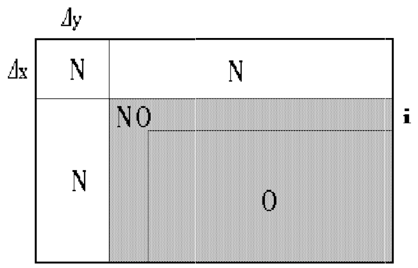

Figure 4.

This means, that we have already processed the part of the picture shown in grey on the figure. The method is then the following: we only apply the thinning algorithm on the part of the picture marked with N on the figure. Using the result of this and part O of the picture obtained by a previous process we get the skeleton recorded at the investigated moment. To the skeleton obtained this way the code

$$(x_0 + \sum_{i=1}^{\Delta x} c_{i,2}, y_0 + \sum_{i=1}^{\Delta x} c_{i,1}), (c_{\Delta x+1,1}, c_{\Delta x+1,2}), \dots, (c_{\Delta x+n,1}, c_{\Delta x+n,2})$$

has to be assigned, where $c_{n+1}, \dots, c_{n+\Delta x}$ is the code of the new subpicture. According to these, the parameters of line e can be determined in the following way:

$$a = \frac{\sum_{i=1}^{n} c_{i,1} - \sum_{i=1}^{\Delta x} c_{i,1} + \sum_{i=n}^{\Delta x+n} c_{i,1}}{\sum_{i=1}^{n} c_{i,2} - \sum_{i=1}^{\Delta x} c_{i,2} + \sum_{i=n}^{\Delta x+n} c_{i,2}} \quad , \quad b = -(y_0 + \sum_{i=1}^{\Delta x} c_{i,1})a.$$

After this, we only have to prove that knowing the skeletons of the binary pictures N and O, respectively, the skeleton of the binary picture $N \cup O$ can be determined. This can be proved — assuring overlap — with the following theorem:

Theorem. *Let* $A = \begin{pmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{pmatrix}$ *be a binary matrix of size* $m \times n$*, where* $A_1, A_2, \ldots, A_9$ *are binary matrices of size* $\Delta x \times \Delta y, \Delta x \times i, \Delta x \times n', i \times \Delta y, i \times i, i \times n', m' \times \Delta y, m' \times i$*, and* $m' \times n'$*, respectively. Let* $N = \begin{pmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & 0 \end{pmatrix}$*, and* $O = \begin{pmatrix} 0 & 0 & 0 \\ 0 & A_5 & A_6 \\ 0 & A_8 & A_9 \end{pmatrix}$*. Let* Ψ *be an iterative thinning algorithm using patterns of size* 3×3 *interpreted on binary matrices of size* $n \times m$*, which in case of binary matrix* A *gives the skeleton of the binary picture represented by the matrix in the* i*th iterative step. Then the following equation holds*

$$(\Psi(N) + \Psi(O)) \wedge T = \Psi(A) \wedge T,$$

where $T = \begin{pmatrix} A_1 & A_2 & A_3 \\ A_4 & 0 & 0 \\ A_7 & 0 & A_9 \end{pmatrix}$ *and* $\wedge$ *is the "logical and" operation between the two binary matrices.*

Proof. The thinning algorithm Ψ deletes the point with coordinates (x, y) on the basis of investigation of points with coordinates (x', y') wich satisfy the inequalities $|x - x'| \leq 1$ and $|y - y'| \leq 1$. (This is the result of using 3×3-size patterns, which means that the deleting is determined by the configuration of the points in a 3×3-size environment.) Obviously, the algorithm makes its decision about deleting the point with coordinates (x, y) in the ith iterative step, using the configuration of the points (x', y'), for which the inequalities $|x - x'| \leq i$ and $|y - y'| \leq i$ hold. As a consequence of this the result of the algorithm Ψ on the submatrix A_9 is independent of the elements of the submatrices A_1, A_2, A_3, A_4, and A_7 and vice versa. The statement of the theorem follows from this.

Experimental results, conclusions

In the executions time of these algorithms in the worst case supposing sequential execution, are the following: in the case of 2+1D thinning, it is $n^4/2$, in the case of Walsh transformation $n^2 \log n$, and in the case of Hough transformation $2n^2$, where n is the size of the picture in pixels. For all three methods there is a chance for paralellization, which considerably can increase the speed of processing. In that case the following time complexities are obtained keeping the order above: $n/2$, $\log n$, and n^2.

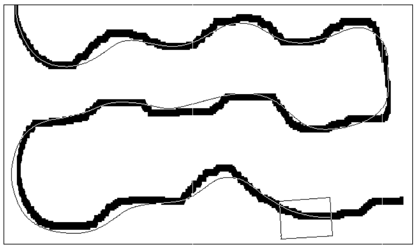

Figure 5.

In Fig. 5 an example for computer simulation can be seen. The degree of "keeping the route" can be given by investigating the quantity

$$E_{\text{tracking}} = \frac{1}{n} \sum_{i=1}^{n} |d(i)|,$$

where n is the number of the pictures processed (the length of the route), and $d(i)$ is the value d belonging to the ith picture. On the basis of the investigations we have done on 50 different ways, we can set up the following order on the basis of tracking e: 2+1D skeletonization, Walsh transformation, and Hough transformation.

References

1. Gy. Erőss, T. Boros, A. Radványi, J. Bitó and T. Roska, *A new optical tracking method for robot cars, using cellular neural networks*, MTA SzTAKI REPORT **DNS-6** (1991), 1–16.
2. R. C. Gonzalez and P. Wintz, *Digital image processing*, Addison-Wesley Publishing Company, Singapore, 1987.
3. A. Fazekas, *Methods and applications of digital image processing*, Bulletins for applied mathematics **778/91** (1991), 369–378.
4. E. S. Deutsch, *Thinning algorithms on rectangular, hexagonal and triangular arrays*, 9, Communications of the ACM **15** (1972), 827–836.

H-4010 Debrecen P.O.Box.12

(Received October 29, 1997)