Two-Dimensional Position Recovery for a Free-Ranging Automated Guided Vehicle

Emil M. Petriu, *Senior Member, IEEE*, William S. McMath, Stephen K. Yeung, Niculaie Trif, and Taco Bieseman

Abstract—**This paper presents a new absolute position recovery method for an automated guided vehicle (AGV) which can move in any direction (free-ranging) on the floor using vision. The floor is permanently encoded with the terms of a pseudo random binary array requiring only one bit of code per quantization interval, independent of the desired resolution.**

I. Introduction

MOST Automated Guided Vehicles (AGV's) currently used in industry follow buried cables or painted guide paths on the floor. While such a guiding technique is robust, it also constrains the AGV motion to the guide path. More advanced free-ranging AGV's which can move in any direction on the floor are in an experimental stage [1] and [2]. Two-dimensional (2-D) position recovery is an essential feature for free-ranging navigation. Vision is the most powerful method to implement this feature, but it still suffers from problems which limit its practical value [1]. A number of machine vision methods use position-referencing patterns [3] and [4], but their applicability is restricted to a limited number of *a priori* defined locations which finally does not allow for the AGV position recovery at any point on the floor.

Based on previous results in 1-D AGV position recovery, [5], this paper discusses a new machine vision method which allows a free-ranging AGV to recover its absolute position and orientation anywhere on the floor. The floor is permanently encoded with the terms (marked by distinct graphical symbols) of a "pseudorandom binary array" (PRBA). This results in a very compact encoding which requires only one bit of code per quantization interval, independent of the desired resolution. As the required image processing is limited to recognition of the binary symbols, the proposed solution is considerably faster and more cost efficient than other vision techniques employed for AGV position recovery.

Manuscript received July 7, 1992; revised November 23, 1992. This work was supported in part by the Canadian Space Agency under DSS Contract 9F009-0-4097/00 and by the Natural Sciences and Engineering Research Council of Canada under Operating Grant OGP0006726.

E. M. Petriu and N. Trif are with the Department of Electrical Engineering, University of Ottawa, Ottawa, Ont. K1N 6N5, Canada.

W. S. McMath and S. K. Yeung are with the Canadian Space Agency, Ottawa, Ont., Canada.

T. Bieseman was with the Department of Electrical Engineering, University of Ottawa, Ottawa, Ont. K1N 6N5, Canada. He is now with Shell Exploration and Production Research Laboratory, Rijswijk, The Netherlands.

IEEE Log Number 9208275.

II. Two-Dimensional Pseudorandom Encoding

The proposed position measurement paradigm is illustrated in Fig. 1. It can be formally stated as follows. "Given a PRBA encoded floor, whose grid node coordinates are defined in a 2-D floor reference frame $x_u O_u y_u$, and given a perspective image showing only a part of the PRBA, determine the 2-D position and orientation of the AGV's frame $x_r O_r y_r$ in the floor's frame $x_u O_u y_u$."

A. Pseudorandom Array Encoding

The floor is permanently encoded with the elements of a PRBA. Such an $n1$-by-$n2$ array can be obtained as shown in Fig. 2, by folding a $(2^n - 1)$-term pseudorandom binary sequence $\{S(p) \mid p = 0, 1, \cdots, 2^n - 2\}$ generated by a modulo-two n-bit shift register (Fig. 3 and Table I). The following relations hold for this PRBA [6]:

$$2^n - 1 = 2^{k1 \cdot k2} - 1 \tag{1}$$

$$n1 = 2^{k1} - 1 \tag{2}$$

$$n2 = (2^n - 1)/n1 \tag{3}$$

where $n1$ and $n2$ must be relatively prime.

The 2-D coordinate recovery is based on the PRBA window property. According to this any $k1$-by-$k2$ nonzero binary pattern seen through a $k1$-by-$k2$ window sliding over the array is unique and may fully identify the window's absolute coordinates (i, j) within the PRBA. The "pseudorandom/natural" code conversion is implemented as a memory-stored table (with $2^n - 1$ addresses).

For instance, as shown in Fig. 4, a 63-bit sequence (generated by an $n = 6$ bit shift register) will produce a PRBA with $n1 = 7$ lines (since $k1 = 3$, and $k2 = 2$) and $n2 = 63/7 = 9$ columns. The pseudorandom binary pattern seen through a 3-by-2 window sliding over this array is unique and may be used to identify the window's absolute coordinates (i, j).

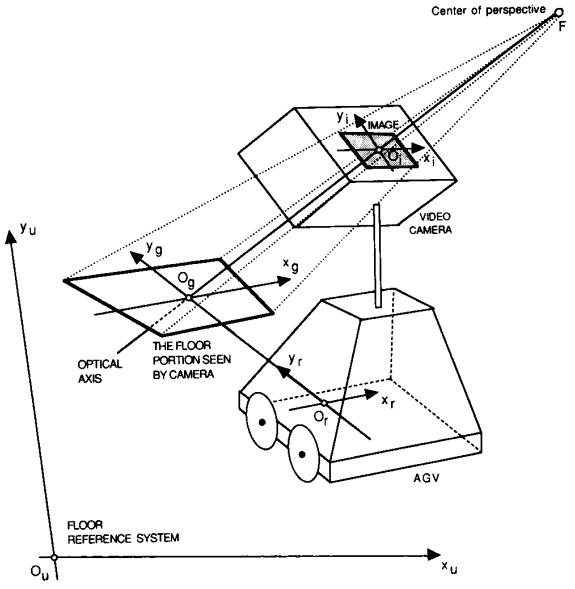

Fig. 1. The free-ranging AGV position recovery paradigm.

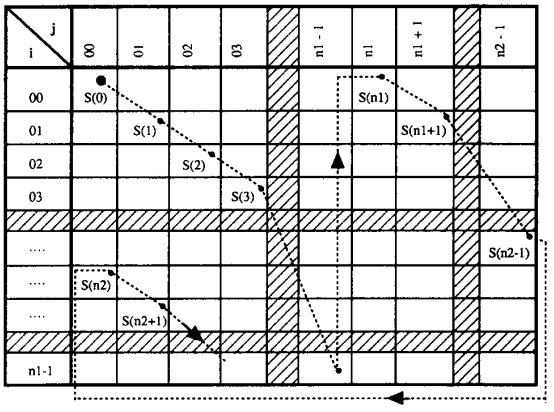

Fig. 2. A $(2^n - 1)$-bit PRBS $\{S(p) | p = 0, 1, \cdots, 2^n - 2\}$ is folded to produce an $n1$-by-$n2$ PRBA.

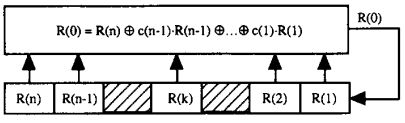

Fig. 3. Moldulo-two direct-feedback shift register.

B. Visual Recognition of the Binary Symbol

Two distinct graphical symbols, shown in Fig. 5, have been used to mark the binary values "1" and "0" within the PRBA. The particular shapes of these binary symbols are selected in such a way to meet the following requirements:

1) there is enough information at the symbol level to provide an immediate indication of the array's orientation and about where neighboring symbols can be found;

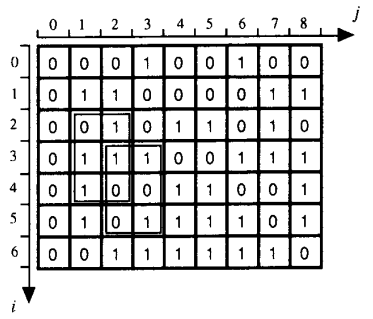

Fig. 4. Example of the window property in a PRBA.

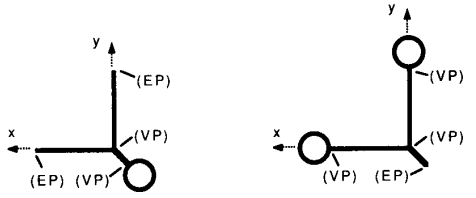

Fig. 5. The two graphical symbols used to mark "0" and "1" within the PRBA. The EP notation identifies an end point, and VP identifies a vertex point in the skeleton of these symbols.

TABLE I
FEEDBACK EQUATIONS FOR P.R.B.S. GENERATION

Shift Register Length n	Feedback for Direct PRBS $R(0) = R(n) \oplus c(n-1) \cdot R(n-1) \oplus \cdots \oplus c(1) \cdot R(1)$	Feedback for Reverse PRBS $R(n+1) = R(1) \oplus b(2) \cdot R(2) \oplus \cdots \oplus b(n) \cdot R(n)$
4	$R(0) = R(4) \oplus R(1)$	$R(5) = R(1) \oplus R(2)$
5	$R(0) = R(5) \oplus R(2)$	$R(6) = R(1) \oplus R(3)$
6	$R(0) = R(6) \oplus R(1)$	$R(7) = R(1) \oplus R(2)$
7	$R(0) = R(7) \oplus R(3)$	$R(8) = R(1) \oplus R(4)$
8	$R(0) = R(8) \oplus R(4) \oplus R(3) \oplus R(2)$	$R(9) = R(1) \oplus R(3) \oplus R(4) \oplus R(5)$
9	$R(0) = R(9) \oplus R(4)$	$R(10) = R(1) \oplus R(5)$
10	$R(0) = R(10) \oplus R(3)$	$R(11) = R(1) \oplus R(4)$

2) the symbol recognition procedure is invariant to position, orientation, scaling, and perspective transformations; and
3) the symbols should have sufficient uniqueness so that other objects in the scene will not be mistaken for encoding symbols.

The initial gray-scale (8-bit) digital image from the image acquisition board is preprocessed by the following steps: local maximum filtering, global thresholding and binary closure. The local maximum filter produces intermediate image in which the encoding symbols have disappeared, but the general illumination effects (shadows for instance) on the background are still present. By extracting this intermediate image from the original gray-scale image, an enhanced image is produced in which the encoding symbols (represented by dark pixels) appear on a uniformly bright background which is not affected by negative illumination effects. The global thresholding ap-

plied to this image produces a binary image. The closure operation, consisting of a dilation followed by an erosion, is used to close little cracks and breaks in the symbols from the binary image.

A skeleton transform is applied to the resulting binary image. The 1-bit thick skeleton is an information-preserving representation of the symbol shapes [7]. Despite its noise sensitivity, the skeleton method was used for this symbol recognition application since vertex points and end points can be easily recognized in a 3-by-3 pixel neighborhood. The encoding symbols are recognized by their numbers of end points and vertices in the skeletonized binary image. The symbol representing "0" has 2 end points and 2 vertex points, and the symbol for "1" has 1 end point and 3 vertex points.

The algorithm used to produce the skeleton is based in the Hilditch crossing number [8], defined as

$$X(p) = \sum_{i=0}^{3} \overline{n_{2i}} \wedge (n_{2i+1} \vee n_{2i+1}) \qquad (4)$$

where Σ is a modulo-8 arithmetic sum, $\wedge$ is the logical operation AND, $\vee$ is the logical operation OR, and "n_0" to "n_8" are the binary values of the eight neighbors of the current pixel "p" as shown in Fig. 6.

In function of their crossing number, the binary image pixels can be classified as:

- isolated pixels characterized by $X(p) = 0$;
- break pixels characterized by $X(p) > 1$;
- contour pixels other than break pixels characterized by $X(p) = 1$.

To find the skeleton, the image is scanned from left to right and top to bottom, and the crossing number is calculated for each pixel. All the pixels that have a crossing number equal to one and the number of the neighbors greater than one are removed.

Each scan of the image results in a thinned version of the objects. In this way all the contour pixels of the object are removed with the exception of the end pixels and break pixels. This scanning is repeated until no more pixels are removed. In this way a skeleton representing the 1-bit medial axis of each object in the binary image is obtained. The skeleton consists only of break pixels and end pixels (an end pixel is a pixel with just one neighbor).

The symbol recognition algorithm starts with linear scan of the *skeleton* image for vertices. This is an efficient approach, as skeletons that do not contain vertices are rejected immediately. This excludes the vast majority of noise from being evaluated. When a vertex is found, a sequential search for vertices and end points in the rest of the connected skeleton is started, and a connectivity table is filled out. The features extracted from the connectivity table to describe a symbol are (Fig. 7):

- the binary value ("0" or "1")
- the x- and y-coordinates of the symbol's position;
- the directions of the symbol's x- and y-axes; and
- the expected distances along the symbol's axes to the neighboring symbols.

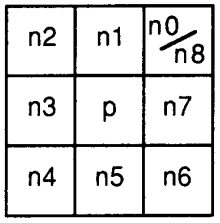

Fig. 6. The eight neighbors of a pixel p.

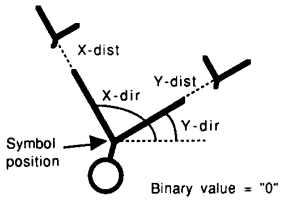

Fig. 7. Features describing a binary symbol.

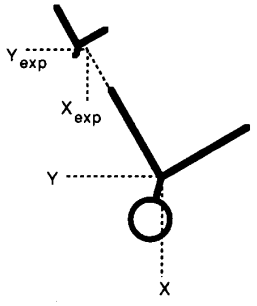

Fig. 8. Estimation of the expected top-neighbor's relative position.

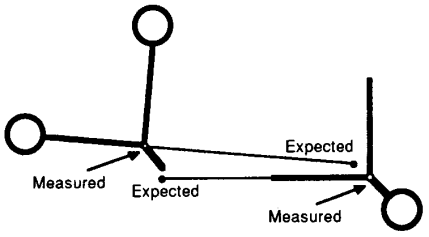

Fig. 9. The double-fitting test.

C. PRBA Reconstruction

The PRBA reconstruction procedure is independent of the employed symbol shapes and the corresponding symbol recognition technique. Its input consists of sets of extracted symbol features, and it deals with their organization into a list of symbol structures and the recognition of the 2-D grid pattern in which the PRBA elements are arranged on the floor.

When a symbol is discovered, its features are added to the linked list of symbols already recognized. From this list, the expected coordinates of the current neighboring symbol are calculated (Fig. 8) and then compared to actual measured coordinates of these neighbors. If the difference between the calculated coordinate values and the

measured symbol coordinates doesn't exceed a certain threshold, the neighbor candidate is accepted for integration into the PRBA grid (Fig. 9). In this way, erroneously recognized symbols are rejected before being integrated in the PRBA. If more than one candidate is found for a given PRBA grid node, then the "best fit" is chosen.

III. Floor Position Recovery from Perspective Image Measurements

A. Perspective Image Geometry

The geometry of the 3-D perspective imaging process is illustrated in Fig. 10 for the nontrivial case of an image plane which is not parallel to the floor plane. Given 2-D coordinates $\{x(A^i), y(A^i)\}$ of a point A in the image frame $x_i O_i y_i$, and knowing the camera's tilt angle α and its lens distance $O_i O_g$ (along the optical axis) to the floor, the problem is to find the coordinates $\{x(C^g), y(C^g)\}$ of the point C (the point A's projection on the floor) in the reference from $x_g O_g y_g$ (which is the floor projection of the image reference frame).

Let us consider an intermediate reference system $x_g O_g y_p$ which defines a plane parallel to the image plane, as shown in Fig. 10. Since $O_g T$ and $O_i S$ are parallel (as there is assumed to be a zero roll angle between the image frame and the floor frame), the triangles FTO_g and FSO_i are similar, which gives

$$O_g T / O_i S = FO_g / FO_i = m, \tag{5}$$

where "m" is a constant magnification ratio. Using the coordinate notation, we obtain from (5):

$$x(C^g) = m \cdot x(A^i). \tag{6}$$

Since $O_g R$ and $O_i P$ are parallel by definition, the triangles FRO_g and FPO_i are similar, which gives

$$O_g R / O_i P = FO_g / FO_i = m. \tag{7}$$

Applying the law of sines to the triangle RQO_g we obtain

$$O_g Q / \sin(90° + \beta) = O_g R / \sin(\alpha - \beta). \tag{8}$$

Substituting $O_g R$ from (7) in (8) we obtain

$$O_g Q / \sin(90° + \beta) = m \cdot O_i P / \sin(\alpha - \beta), \tag{9}$$

which gives after further development

$$O_g Q = m \cdot O_i P / (\sin \alpha - \cos \alpha \cdot \tan \beta). \tag{10}$$

From the triangle $FO_i P$:

$$\tan \beta = PO_i / FO_i = y(A^i)/f, \tag{11}$$

where f is the constant focal length of the camera lens.

Replacing $\tan \beta$ given by (11) in (10) and using the coordinate notation we finally obtain

$$y(C^g) = y(A^i) \cdot m \cdot f / [f \cdot \sin \alpha - \cos \alpha \cdot y(A^i)]. \tag{12}$$

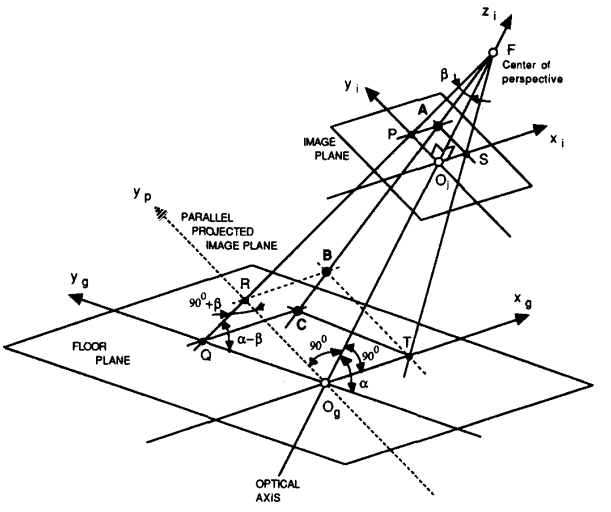

Fig. 10. Perspective distortions due to a camera tilt (angle α).

B. 2-D Position and Orientation Recovery

Based on the perspective transformations (6) and (12), the AGV position measurement paradigm shown in Fig. 1 can be reduced to a 2-D geometric problem, as illustrated in Fig. 11.

The AGV's orientation is defined by the angle θ_z between the x-axes of the robot's, and, respectively, the floor's reference frames. Because of the parallelism which exists (by construction) between the projected image reference frame and the robot's frame on one side, and between the PRBA grid lines and the axes of the floor's reference frame on the other side, this orientation angle θ_z is identical to the angle between any grid line in the reconstructed PRBA and the $O_g x_g$ axis. It is then possible to write from Fig. 11:

$$\tan \theta_z = [y(C^g(i, j)) - y(C^g(i, j+1))] / [x(C^g(i, j)) - x(C^g(i, j+1))]. \tag{13}$$

Using (6) and (12) which relate the 2-D coordinates of the floor-marked grid nodes $C^g(\cdots, \cdots)$ to those of their images $A^i(\cdots, \cdots)$, (13) finally becomes

$$\tan \theta_z = \frac{y(A^i(i, j)) - y(A^i(i, j+1))}{x(A^i(i, j)) - x(A^i(i, j+1))} \cdot \frac{f^2 \cdot \sin \alpha}{[f \cdot \sin \alpha - \cos \alpha \cdot y(A^i(i, j))] \cdot [f \cdot \sin \cdot \alpha - \cos \alpha \cdot y(A^i(i, j+1))]}, \tag{14}$$

which allows calculation of the AGV's orientation from the 2-D coordinates of two adjacent grid nodes, $A^i(i, j)$ and $A^i(i, j+1)$, identified in the PRBA image acquired by the onboard video camera.

The following vectorial equation can be written for the

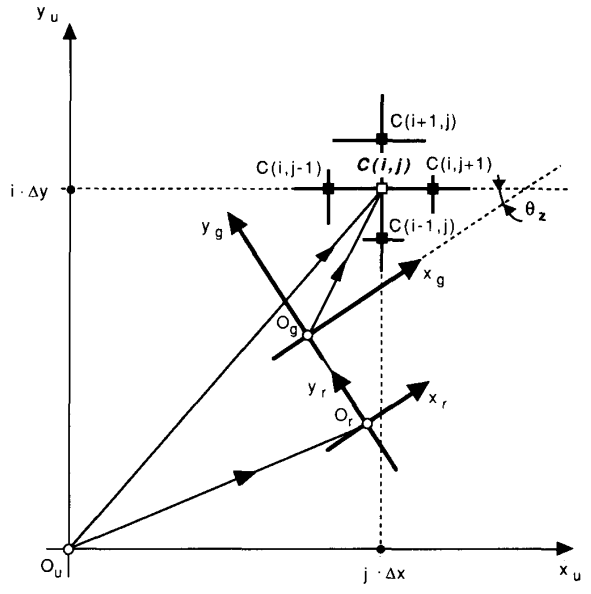

Fig. 11. The position recovery reduced to the floor plane.

relationship which exists (as shown in Fig. 11) between the frames of the AGV, projected image, and floor:

$$\overline{O_u C(i,j)} = \overline{O_u O_r} + \overline{O_r O_g} + \overline{O_g C(i,j)}. \quad (15)$$

Developing (15) in a scalar form, the following two relations result for the 2-D coordinates of the AGV's frame origin O_r in the floor reference frame:

$$x(O_r^u) = j \cdot \Delta x - x(C^g(i,j)) \cdot \cos\theta_z + [d + y(C^g(i,j))] \cdot \sin\theta_z \quad (16)$$

$$y(O_r^u) = i \cdot \Delta y - x(C^g(i,j)) \cdot \sin\theta_z + [d + y(C^g(i,j))] \cdot \cos\theta_z, \quad (17)$$

where $d = O_r O_g$ is the fixed distance between the AGV's frame and the projected image's frame origin, and Δx and Δy are the quantization steps in the x and, respectively, y direction.

Using (6) and (12), (16) and (17) finally become

$$x(O_r^u) = j \cdot \Delta x + d \cdot \sin\theta_z - m \cdot x(A^i(i,j)) \cdot \cos\theta_z + m \cdot f \cdot y(A^i(i,j)) \cdot \sin\theta_z / [f \cdot \sin\alpha - \cos\alpha \cdot y(A^i(i,j))] \quad (18)$$

$$y(O_r^u) = i \cdot \Delta y - d \cdot \cos\theta_z - m \cdot x(A^i(i,j)) \cdot \sin\theta_z - m \cdot f \cdot y(A^i(i,j)) \cdot \cos\theta_z / [f \cdot \sin\alpha - \cos\alpha \cdot y(A^i(i,j))]. \quad (19)$$

The last two relationships allow calculations of the AGV's position from 2-D coordinates of a grid node $A^i(i,j)$ identified in the PRBA image acquired by the onboard video camera. The other parameters to be used are m defined by (7), f which is the constant focal length of the camera lens, α which is the camera's tilt angle, and the AGV's orientation angle θ_z previously recovered using (14).

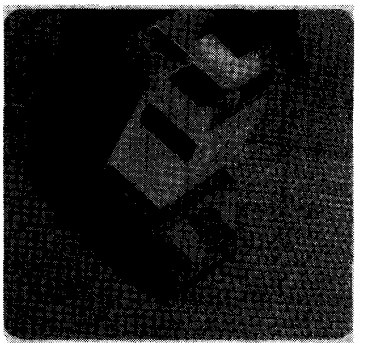

Fig. 12. The experimental vision-guided AGV on the PRBA-encoded floor.

IV. Experimental Results

A free-ranging AGV, shown in Fig. 12, was developed to validate the discussed vision method for 2-D absolute recovery. The floor is permanently marked (with a 55 mm grid step) with the terms of a 63-by-65 PRBA (with $k1 = 6$, $k2 = 2$, $n1 = 63$, $n2 = 65$) which requires a 6-by-2 window for absolute recovery.

The vision system uses a CCD natural camera and a "Matrox PIP 512" image acquisition board resident in a 286 personal computer working at an 8 MHz clock rate. The following representative times have been obtained while processing a 256-by-256 pixel image: 5.93 s for the local maximum filter operation, 0.99 s for the global thresholding, 2.55 s for the closure operation, 3.46 s for the skeleton transform, 0.8 s for break- and end-pixel marking, 3.96 s for symbol recognition and grid reconstruction, 0.06 s for the window's position recovery, and 0.12 s for perspective corrections and AGV position calculation. This results in a total of 17.95 s processing time per image. As expected, the preprocessing and skeleton finding take most of this time. These processing times can be expected to improve by a factor of 12 to 15 by using a 486 personal computer working at a 33 MHz clock rate and a more advanced image acquisition board (e.g. "Matrox 1280") providing onboard filtering and skeleton operations implemented by hardware.

Experimental results shown in Fig. 13 illustrate the performance of this technique. In this case, the recovered grid node coordinates in the floor reference system $x_u O_u y_u$ defined in Fig. 11 are $j = 60$ and $i = 2$, and the AGV's orientation angle is $\theta_z = 5.71^0$. This vision method has an accuracy of 1.5 mm (corresponding to one image pixel) for 2-D AGV position measurement on the floor encoded with a 55 mm grid step.

On the basis of experiments with hundreds of processed images, the described method appears to be robust in the sense that it never produced a grid with false PRBA patterns. Extreme perspective distortion can result in incorrect neighbor distance estimation. Symbols may not be recognized, or a "1" symbol may be recognized as a "0"

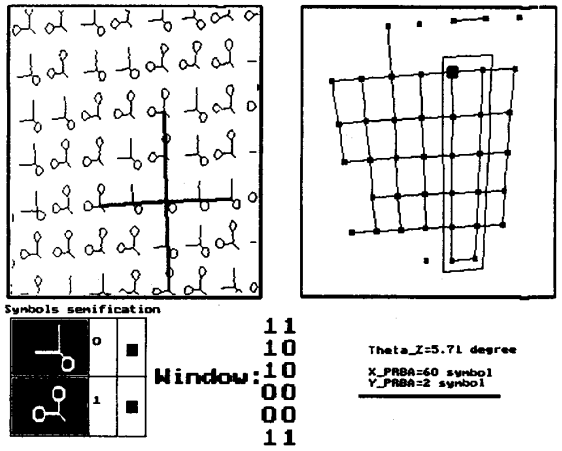

Fig. 13. Floor image acquired by the onboard camera, the recovered PRBA window, and the resulting positional parameters.

symbol due to a break in the skeleton, and sometimes noise may be recognized as a symbol. However, during testing, these falsely recognized symbols never had the correct size and orientation to fit to their neighbors as required for acceptance and integration into the PRBA grid. The reconstructed grid may thus be incomplete; however, the information which it contains is still reliable.

V. Conclusion

The ability to instantaneously recover 2-D absolute position allows the free-ranging AGV to move in any direction on the encoded floor. The PRBA-based encoding technique requires only one bit of code per quantization interval, independent of the desired resolution. The onboard camera also helps the AGV's navigation by providing a larger and deeper perspective of the environment.

When objects resting on the floor cover too large an area on the image it is possible that not enough symbols are recognized to recover at least on complete pseudo-random window. In such case it is necessary to move the vehicle and get another image of the encoded floor.

Further research is pursued in the following directions: 1) development of PRBA code conversion algorithms having a better "conversion time/equipment cost" performance than currently obtained using the ROM-type conversion and 2) experimentation with different binary symbol shapes and symbol recognition techniques.

References

[1] S. K. Premi and C. B. Besant, "A review of various vehicle guidance techniques that can be used by mobile robots or AGV's," in *Proc. 2nd Int. Conf. Automated Guided Vehicle Systems, and 16th IPA Conf.*, Stuttgart, Germany, June 1983, pp. 195–209.

[2] R. K. Miller, *Automated Guided Vehicles and Automated Manufacturing*. Dearborn, MI: Soc. Manuf. Eng., 1987.

[3] M. R. Kabuka and A. E. Arenas, "Position of a Mobile Robot Using Standard Pattern," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 505–516, Dec. 1987.

[4] K. Mandel and N. A. Duffie, "On-line compensation of mobile robot docking errors," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 591–598, 1987.

[5] E. Petriu and J. S. Basran, "On the position measurement of automated guided vehicles using pseudo-random encoding," *IEEE Trans. Instrum. Meas.*, vol. 38, pp. 799–803, June 1989.

[6] F. J. Macwilliams and N. J. A. Sloane, "Pseudorandom sequences and arrays," *Proc. IEEE*, vol. 64, no. 12, pp. 1715–1729, Dec. 1976.

[7] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1. Reading, MA: Addison-Wesley, 1992.

[8] J. Hilditch, "Linear skeletons from square cupboards," in *Machine Intelligence*, vol. 4, B. Meltzer and D. Michie, Eds. Edinburgh, Scotland: Edinburgh University Press, 1969, pp. 403–420.