Positioning by Tree Detection Sensor and Dead Reckoning for Outdoor Navigation of a Mobile Robot

Shoichi MAEYAMA, Akihisa OHYA and Shin'ichi YUTA

Intelligent Robot Laboratory, Institute of Information Science and Electronics,
University of Tsukuba,
Tsukuba, 305 JAPAN
phone:+81-298-53-5155 fax:+81-298-53-5206
Email:maeyama@roboken.is.tsukuba.ac.jp

***Abstract*—We propose a positioning method for outdoor navigation of a mobile robot, by fusing dead reckoning and the tree detection sensor which consists of Sonar and Vision. A street lined with trees is assumed to be the mobile robot's outdoor work space. In this environment, trees are good landmarks for robot's position estimation. This paper describes a method for robot position estimation by fusion of dead reckoning and tree detection sensor based on maximum likelihood estimation at first. Then, the method for the detection of tree using the sensor system with sonar and vision mounted in one body is described. At last, the experimental results of self-guidance with the experimental autonomous mobile robot "YAMABICO" is presented. The result shows the effectiveness of the our method for outdoor navigation of the mobile robot.**

Fig. 1. Example of Outdoor Environment

I. Introduction

We are interested in the long distance navigation of an autonomous mobile robot in an outdoor environment, such as a campus of university. So, one of the goal of this research is to realize the navigation of 2kms in the campus of our university. Many methods for outdoor navigation have been already proposed. These methods includes walking on the terrain[1] [2], tracking a white line[3] or following the road edges[4][5]. We have also reported the experiment of outdoor navigation using walls and hedges[6]. Different from tracking or following a line, our method is principally based on dead reckoning. And the position estimated by dead reckoning is properly corrected more precisely when detecting landmark in map information.

But, by the conventional method, it is difficult to navigate robot in the environment shown in Fig.1, which has no walls, no painted lines nor no obvious road edges. So, we propose autonomous mobile robot navigation using trees as landmarks in the environment. The method to update the estimated position of dead reckoning by measuring the point type landmark have already reported by Komoriya [7], which discussed the selection of landmark to minimize error ellipse when plural landmarks are available in the corridor environment. In outdoor environment, there is no enough landmarks in usual cases. And, the position of natural landmark can not define precisely. Furthermore, the accuracy of dead reckoning are not good because the surface of road has bumpier than in corridor. Therefore, it is more important to correct the estimated position of dead reckoning by external sensor. Our interest is to navigate a mobile robot robustly in such severe environment.

In this paper, we propose a method for estimating the position of a mobile robot using trees, and show the result of experiment on a guidance of a mobile robot itself by the proposed method.

II. Positioning using *Tree*

A. Problem formulation

The mobile robot is assumed to be controlled its motion based on the estimated position obtained by information of the internal and external sensors, and quantitative *route map* information which includes the trajectories to be followed, turning points, turning direction, sensing points of landmarks and landmark information.

In the assumed environments shown as Fig.1, the boundary edges of the road is sometimes difficult to use as landmark, because the road can be covered with fallen leaves. On the other hand, trees are suitable landmarks, because it does not depend on a change of conditions and its location is constant. Therefore, we use trees as landmarks for navigation.

The objectives of this research are as follows,

1. Development of a sensor system to detect a tree and to measure robot's position
2. Developing a position estimation method using the sensor system
3. Achieving autonomous mobile robot navigation using the position estimation system

B. Basic strategy of guidance based on dead reckoning and tree observation

The robot can always observe its current position by dead reckoning. However, dead reckoning has the accumulated errors.

Therefore, to navigate long distances, the robot must be able to modify its own estimated position by external sensors and an environmental map.

The algorithm for positioning using trees is as follows. In this algorithm, trajectories to be followed and the sensing points to correct the position using the landmarks are assumed to be planned prior to navigation and supplied to the robot[8].

1. Guidance of the robot between sensing points is done based on dead reckoning.
2. The robot obtains the *position*, *width* and the *slant* information of the tree as a landmark from an environmental map, and calculates the expected *distance* and *direction* of tree from the robot.
3. Using these parameters, the tree detection sensor tries to the tree and measure the distance and direction.
4. If the *tree* is detected, robot fuses the measured values and the dead reckoning information using the technique described in section III, to correct its estimated position and variance of error.

III. Fusion of the tree detection sensor and the dead reckoning information

In this section, a method how the robot's position and the covariances of errors estimated from dead reckoning is corrected by the measurements made by the tree detection sensor is described. The errors in the measurements and the calculations are assumed to be white noise.

A. Position and Error estimation using Dead reckoning

We represent robot's position using the vector

$$\mathbf{P}[t] = \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T$$

where, $(x(t), y(t))$ is the two dimensional location of robot and $\theta(t)$ is the robot's orientation. The estimated errors are represented by the covariance matrix of $\mathbf{P}[t]$

$$\mathbf{\Sigma}_P[t] = \begin{bmatrix} {\sigma_x}^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & {\sigma_y}^2 & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & {\sigma_\theta}^2 \end{bmatrix} \quad (1)$$

These information are updated every sampling interval τ by accumulating displacement caused by rotation of the robot's wheels, as

$$\mathbf{P}[t+\tau] = \mathbf{P}[t] + \tau \begin{pmatrix} v[t]cos(\theta[t]) \\ v[t]sin(\theta[t]) \\ \omega[t] \end{pmatrix} + \tau\mathbf{n}[t] \quad (2)$$

Where, $v[t]$ is the velocity, $\omega[t]$ is the rotational angular velocity of robot's body, $\mathbf{n}[t]$ includes errors of calculations and sampling.

Let the wheels velocity and the tread be expressed in the vector form $\mathbf{m}[t]$. And, $f[\mathbf{p}[t], \mathbf{m}[t]]$ represents first and sencond terms of equation (2). $\hat{\mathbf{P}}[t]$ is the estimated value of $\mathbf{P}[t]$, $\Delta\mathbf{P}[t]$ is the errors of $\hat{\mathbf{P}}[t]$, $\hat{\mathbf{m}}[t]$ is the measured value of $\mathbf{m}[t]$, and $\Delta\mathbf{m}[t]$ is the errors of $\hat{\mathbf{m}}[t]$. Then,

$$\begin{aligned} \mathbf{P}[t+\tau] &= f[\mathbf{P}[t], \mathbf{m}[t]] + \tau\mathbf{n}[t] \\ &= f[\hat{\mathbf{P}}[t] + \Delta\mathbf{P}[t], \hat{\mathbf{m}}[t] + \Delta\mathbf{m}[t]] + \tau\mathbf{n}[t] \\ &\simeq f[\hat{\mathbf{P}}[t], \hat{\mathbf{m}}[t]] + \mathbf{J}[t]\Delta\mathbf{P}[t] + \mathbf{K}[t]\Delta\mathbf{m}[t] + \tau\mathbf{n}[t] \\ &= \hat{\mathbf{P}}[t+\tau] + \Delta\mathbf{P}[t+\tau] \end{aligned} \quad (3)$$

Therefore, the errors in dead reckoning increase as

$$\Delta\mathbf{P}[t+\tau] = \mathbf{J}[t]\Delta\mathbf{P}[t] + \mathbf{K}[t]\Delta\mathbf{m}[t] + \tau\mathbf{n}[t] \quad (4)$$

Where,

$$\mathbf{J}(t) = \left.\frac{\partial f[\mathbf{P}(t),\mathbf{m}(t)]}{\partial \mathbf{P}(t)}\right|_{\hat{\mathbf{P}}(t),\hat{\mathbf{m}}(t)}$$
$$\mathbf{K}(t) = \left.\frac{\partial f[\mathbf{P}(t),\mathbf{m}(t)]}{\partial \mathbf{m}(t)}\right|_{\hat{\mathbf{P}}(t),\hat{\mathbf{m}}(t)} \tag{5}$$

The covariance matrix $\mathbf{\Sigma}_P[t]$ is represented as

$$\mathbf{\Sigma}_P[t] = E(\Delta\mathbf{P}[t]\Delta\mathbf{P}[t]^T) \tag{6}$$

Therefore, with sampling interval τ, the covariance matrix $\mathbf{\Sigma}_P[t]$ is updated as

$$\mathbf{\Sigma}_P[t+\tau] = \mathbf{J}\mathbf{\Sigma}_P[t]\mathbf{J}^T + \mathbf{K}\mathbf{\Sigma}_m[t]\mathbf{K}^T + \tau^2\mathbf{\Sigma}_N \tag{7}$$

Where,

$$\mathbf{\Sigma}_m = E(\Delta\mathbf{m}[t]\Delta\mathbf{m}[t]^T)$$
$$\mathbf{\Sigma}_N = E(\mathbf{n}[t]\mathbf{n}[t]^T)$$

B. Position Information by the tree Detection Sensor

Principally, the shape of a tree doesn't depend on the direction to look at. Therefore, from the tree shape in the image, it is difficult to know the observing direction. Consequently, the position information obtained from a tree detection sensor is only two dimensional information with the distance r from tree and angle α between tree and robot direction. So, even if the location of a tree (x_t,y_t) is *priori* known, we can not obtain the robot's position directly. However, the robot is expected to be located on the circumference of a circle which has radius r and a location center (x_t,y_t). Then, if the robot's location on the circumference is fixed, the orientation of the robot become unique.

We define the vector of information obtained from the *tree* as

$$\mathbf{s} = \begin{bmatrix} x_t & y_t & \alpha & r \end{bmatrix}^T \tag{8}$$

In reality, this information contains errors. If we suppose the errors doesn't have correlation, then, the covariance matrix $\mathbf{\Sigma}_s$ can be represented as

$$\mathbf{\Sigma}_s = diag\begin{bmatrix} {\sigma_{x_t}}^2 & {\sigma_{y_t}}^2 & {\sigma_\alpha}^2 & {\sigma_r}^2 \end{bmatrix} \tag{9}$$

The relationship

$$\mathbf{g}[\mathbf{P}[t],\mathbf{s}] = \begin{bmatrix} (x-x_t)+r\cos(\theta+\alpha) \\ (y-y_t)+r\sin(\theta+\alpha) \end{bmatrix} = \mathbf{0}_{2\times1} \tag{10}$$

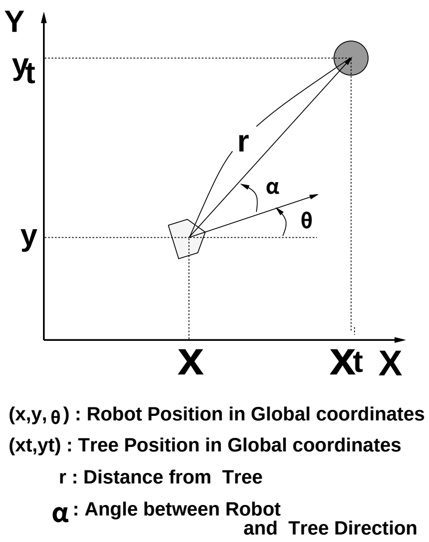

Fig. 2. Configuration of Tree and Robot

can be derived from Fig.2. This equation express the information provided by the sensor.

Supposing that the errors in position $\hat{\mathbf{P}}[t]$ estimated by dead reckoning are small, we can linearize (10) around $\hat{\mathbf{P}}[t]$ and get

$$\mathbf{g}[\hat{\mathbf{P}}[t],\hat{\mathbf{s}}] + \mathbf{j}_p(\mathbf{P}[t]-\hat{\mathbf{P}}[t]) + \mathbf{j}_s\Delta\mathbf{s} = \mathbf{0}_{2\times1} \tag{11}$$

Where, $\hat{s}$ is the measured vector of s, Δs is the error of $\hat{s}$ and, $\mathbf{j}_p$ and $\mathbf{j}_s$ are given as

$$\mathbf{j}_p = \left.\frac{\partial \mathbf{g}[\mathbf{P}(t),\mathbf{s}]}{\partial \mathbf{P}(t)}\right|_{\hat{\mathbf{P}}(t),\hat{\mathbf{s}}}$$
$$\mathbf{j}_s = \left.\frac{\partial \mathbf{g}[\mathbf{P}(t),\mathbf{s}]}{\partial \mathbf{s}}\right|_{\hat{\mathbf{P}}(t),\hat{\mathbf{s}}}$$

By normalizing both sides of (11) by each row vector of $\mathbf{j}_p$, (11) becomes

$$\mathbf{J}_p(\mathbf{P}[t]-\hat{\mathbf{P}}[t]) = -\mathbf{G} - \mathbf{J}_s\Delta\mathbf{s} \tag{12}$$

Where, each parameter is calculated as follows:

$$|\mathbf{j}_{p1}| = \sqrt{1+\hat{r}^2\sin^2(\hat{\theta}+\hat{\alpha})}$$
$$|\mathbf{j}_{p2}| = \sqrt{1+\hat{r}^2\cos^2(\hat{\theta}+\hat{\alpha})}$$

And,

$$\mathbf{J}_p = \begin{bmatrix} \frac{1}{|\mathbf{j}_{p1}|} & 0 & \frac{-\hat{r}\sin(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p1}|} \\ 0 & \frac{1}{|\mathbf{j}_{p2}|} & \frac{\hat{r}\cos(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p2}|} \end{bmatrix} \tag{13}$$

$$\mathbf{J}_s = \begin{bmatrix} \frac{-1}{|\mathbf{j}_{p1}|} & 0 & \frac{-\hat{r}\sin(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p1}|} & \frac{\cos(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p1}|} \\ 0 & \frac{-1}{|\mathbf{j}_{p2}|} & \frac{\hat{r}\cos(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p2}|} & \frac{\sin(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p2}|} \end{bmatrix} \quad (14)$$

$$\mathbf{G} = \begin{pmatrix} \frac{(\hat{x}-\hat{x}_t)+\hat{r}\cos(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p1}|} \\ \frac{(\hat{y}-\hat{y}_t)+\hat{r}\sin(\hat{\theta}+\hat{\alpha})}{|\mathbf{j}_{p2}|} \end{pmatrix} \quad (15)$$

By adding the normalized vector which is independent to the each row vectors of $\mathbf{J}_p$, the 3×3 matrix

$$\mathbf{J'}_p = \begin{bmatrix} & \mathbf{J}_p & \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

is defined. And, the sensor information obtained from tree detection sensor is represented as

$$\hat{\mathbf{P}}_{su} = \mathbf{J'}_p(\hat{\mathbf{P}}_s - \hat{\mathbf{P}}[t]) \quad (17)$$
$$= \begin{bmatrix} -\mathbf{G} \\ a \end{bmatrix} \quad (18)$$

Where, a is an undefinite value. i.e. the variance of a is ∞. But, the inverse matrix of the covariance matrix $\mathbf{\Sigma}_{su}$ is given as

$$\mathbf{\Sigma}_{su}^{-1} = \begin{bmatrix} (\mathbf{J}_s\mathbf{\Sigma}_s\mathbf{J}_s^T)^{-1} & \mathbf{0} \\ \mathbf{0}_{1\times 2} & \mathbf{0} \end{bmatrix} \quad (19)$$

Using this matrix, we can get a formula for fusion of dead reckoning and tree observing.

C. *Position correction by Maximum Likelihood Estimation*

To fuse the information of the tree detection sensor with the estimated dead reckoning vector $\hat{\mathbf{P}}[t]$, the vector $\hat{\mathbf{P}}[t]$ is transformed into the same coordinate as (17).

$\hat{\mathbf{P}}_u[t]$ denotes the vector $\hat{\mathbf{P}}[t]$ in converted coordinates and $\mathbf{\Sigma}_u[t]$ is its covaliance matrix,

$$\hat{\mathbf{P}}_u[t] = 0 \quad (20)$$
$$\mathbf{\Sigma}_u[t] = \mathbf{J}_p^{'}\mathbf{\Sigma}_p[t]\mathbf{J}_p^{'T} \quad (21)$$

Then, Maximum Likelihood Estimation is formulated as

$$\mathbf{\Sigma}_{fu} = (\mathbf{\Sigma}_u[t]^{-1} + \mathbf{\Sigma}_{su}^{-1})^{-1} \quad (22)$$
$$\hat{\mathbf{P}}_{fu} = \mathbf{\Sigma}_{fu}\mathbf{\Sigma}_{su}^{-1}\hat{\mathbf{P}}_{su} \quad (23)$$

Equations (22) (23) are transformed back into the expression of $xy\theta$ axes as,

$$\mathbf{\Sigma}_f = \mathbf{J}_p^{'-1}\mathbf{\Sigma}_{fu}(\mathbf{J}_p^{'T})^{-1}$$
$$= \{\mathbf{\Sigma}_p[t]^{-1} + \mathbf{J}_p^{'T}\mathbf{\Sigma}_{su}^{-1}\mathbf{J}_p^{'}\}^{-1} \quad (24)$$
$$\hat{\mathbf{P}}_f = \hat{\mathbf{P}}[t] + \mathbf{J}_p^{'-1}\hat{\mathbf{P}}_{fu}$$
$$= \hat{\mathbf{P}}[t] + \mathbf{\Sigma}_f\mathbf{J}_p^{'T}\mathbf{\Sigma}_{su}^{-1}\hat{\mathbf{P}}_{su} \quad (25)$$

This formulation fuses tree detection sensor information and dead reckoning position information to obtain a corrected estimated position.

IV. A Sensor to detect a tree and measure the direction and distance

In this section, the constitution of the tree detection sensor and its algorithm are presented, to obtain the distance and the direction of a tree.

The requirements of a sensor which detects and measures the relative position of a tree landmark are as follows:

- No misdetecting of a landmark
- High accuracy
- Fast calculation

A. *Properties of the Sensor*

The tree detection sensor is combined with ultrasonic range sensor and vision system.

Ultrasonic and vision sensors are often used to obtain external information for mobile robots[10][11]. Ultrasonic range sensors can measure the distance from a *tree* using simple electric circuit. However, using only distance information, it is not easy to judge it is reliable or not. Also, the measured distance is not always obtained from the direction pointed by the sensor, since the beam width is wide. A vision sensor can not measure distance directly, and, usually, it takes a long time to process an image. However, when a tree is detected in an image, the direction to the tree can be measured with more high accuracy. Both the ultrasonic and vision sensors can measure only one dimensional information, either distance or direction.

Therefore, the information of 2 sensors must be combined. That is to say, the lack of accuracy in direction which is weakness of ultrasonic range sensor is compensated for by image processing. And, by using the distance from ultrasonic sensor, we can process the image more quickly. When the distance of the ultrasonic sensor shows a suitable value compared with landmark information in the environmental map, no obstacles between camera and landmark are expected. And, the tree can be detected by vision sensor.

B. *Tree detection sensor* SONAVIS

To detect a tree, we propose a sensor system combining sonar and vision(named *SONAVIS*).

As shown in Fig.3, the hardware of *SONAVIS* consists of sonar and vision sensors mounted in one body on a turn table.

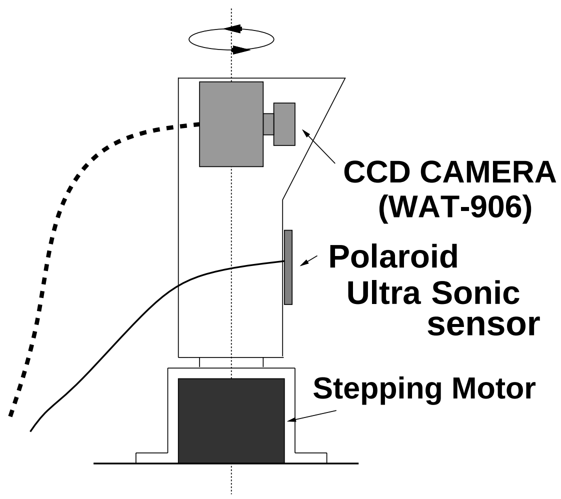

Fig. 3. Configuration of Sensor Parts for SONAVIS

The software to use the sensor module has the interface of

int FindTree(&direction,&distance,diameter,slant)

To request sensing, the function *FindTree()* is called with the estimated *direction*, the estimated *distance*,*diameter* and *slant* of tree. These estimated values are obtained from the estimated position of robot and the landmark information stored in the environmental map. The sensor module detects the tree near the estimated values. As the result, it outputs the presence or absence of a tree and returns the measured *direction* and *distance*.

C. *Algorithm for the detection of Tree and measure*

Algorithm Flow:

1. SONAVIS is pointed to the requested direction.
2. The distance from tree is measured with the ultrasonic sensor. If the measured distance is not reasonable compared with the estimated value, return *No Detection*.
3. If the measured distance is reasonable, obtain the image, detect the tree, calculate the direction by using width of the tree in the image, which is obtained from the measured distance, and so on. If the tree is not detected in the image, return *No Detection*.
4. When the expected tree is detected, return the measured distance and direction, and reply *Detection*.

Algorithm for Image Processing:

To detect a tree, following two assumption of a tree in an image are used.

1. Both sides of a *tree's trunk* in an image are edges in the vertical direction. w denotes the width of the tree in the image. d denotes the range in which edges of the *trunk* can exist. w and d are calculated from the measured distance and the landmark information stored in the environmental map.
2. The tree constitutes the image area in which values are darker than background and uniform in shading. Therefore, the differential values on the left side edges of a *tree's trunk* are negative, differential values on the right side are positive.

We define the horizontal axis as X, the vertical axis as Y, and the top of left corner is defined as $(x, y) = (0, 0)$.

After inputing the image, differentials in the X direction

$$\Delta f[x,y] = -f[x-1,y] + f[x+1,y] \tag{26}$$

are calculated.

Next, the vertical edges

$$s_p[x] = \sum_{y=0}^{n} \begin{cases} \Delta f[x,y] & \text{if } \Delta f[x,y] > 0 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

$$s_m[x] = \sum_{y=0}^{n} \begin{cases} -\Delta f[x,y] & \text{if } \Delta f[x,y] < 0 \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

are calculated. Where, n is y value on the boundary between the ground and the tree in the image, which is calculated from the distance obtained by the ultrasonic sensor. To estimate the central axis of the tree,

$$col[c] = \sum_{m=-w/2-d/2}^{-w/2+d/2} s_m[c+m] + \sum_{m=w/2-d/2}^{w/2+d/2} s_p[c+m] \tag{29}$$

is evaluated and, finds the maximum point of col[c]. Where, $w/2 + d/2 \leq c \leq X_{max} - (w/2 + d/2)$. When we have a tree of width w on the center position c in the image, col[c] has large value. Therefore, 75% of the value of maximum of col[c] is used as the threshold. The portions exceeding of this threshold are considered as candidates for the central axis of the tree. The local maximum is used as the central axis of the tree.

For example, in the image shown in Fig.4, the results of calculations using equation (26) (27) (28) is shown in Fig.5 and the result using equation (29) is shown in Fig.6

If the position of *tree* decided using this algorithm is not unique, the positions are verified again by using the

Fig. 4. Original image with the detected center line(white line)

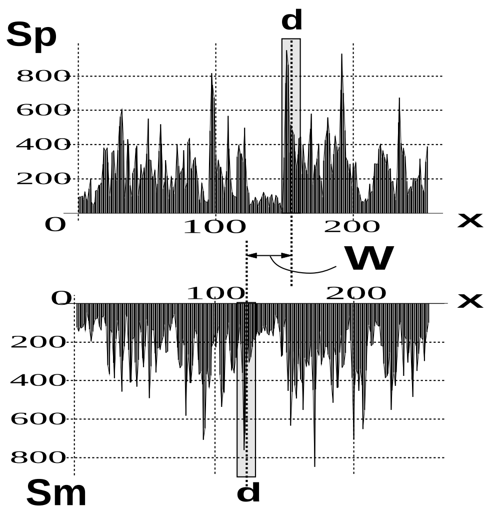

Fig. 5. Weight histogram by summing differential values

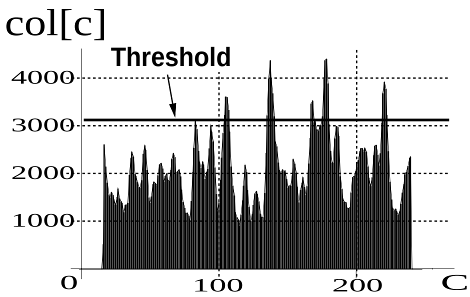

Fig. 6. Evaluated values of the position of tree

characteristics of a *tree's trunk* in the image. Two characteristics decide center line of a *tree*. These center lines are inside a *tree's trunk* and the values are uniform. That is to say, the mean value $\overline{e_i}$ of the center line candidate i is reasonable and that values are uniform.

When the x coordinate value of the center line of candidate i is c, the uniformity of values

$$\sigma[i] = \sum_{y=0}^{n} |f[c, y] - \overline{e_i}| \tag{30}$$

is calculated. By selecting a minimum $\sigma[i]$, the center line of the *tree* is decided uniquely.

The strong point of this algorithm has no multiplications and consists of only additions and subtractions. Furthermore, in only one scan, the candidates for the center line are decided. Therefore, the calculation is very fast.

V. Implementation and Experiment

A. Implementation

The SONAVIS was implemented on the experimental autonomous mobile robot the YAMABICO (Fig.7). Position estimation sub-system, which includes functions

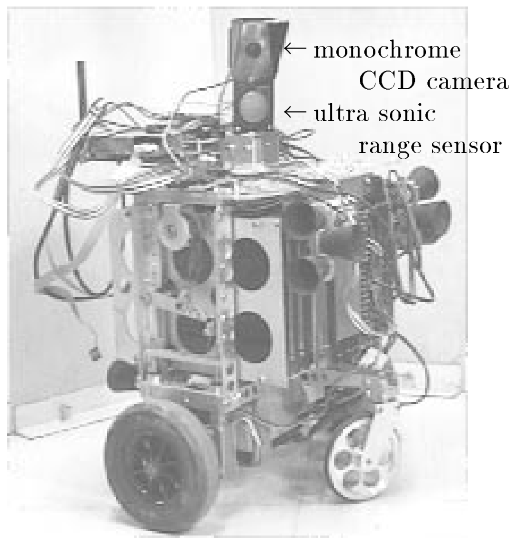

Fig. 7. Experimental mobile robot YAMABICO

of position estimation based on dead reckoning and calculation for maximum likelihood estimation, also implemented as a single board sub-system and named POEM II(Position Estimation module II).

The controller of YAMABICO is modularized, and it consists of a master module and several function modules. In the master module, robot's motion is decided and controlled. Each function module has its own specified function(motor control, range sensing, and so on). This schematic architecture is shown in Fig.8.

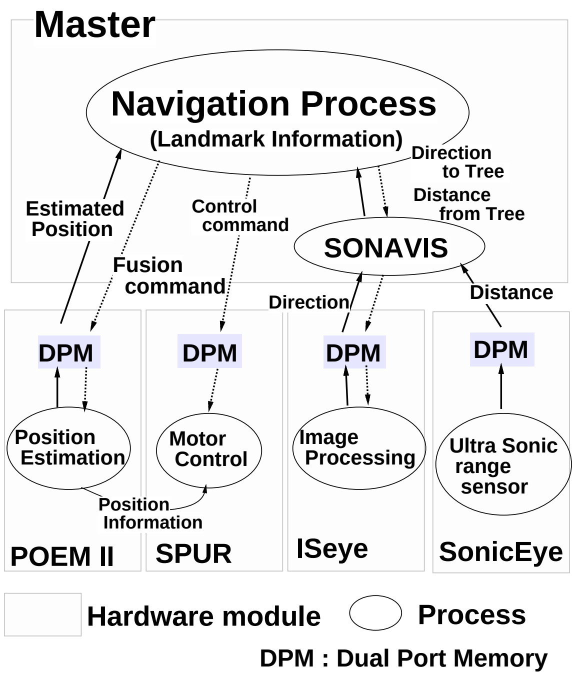

Fig. 8. Module Structure of YAMABICO

In hardware, each module has an independent CPU(SONAVIS: 68000(MOTOROLA),POEM II: T805(SGS-THOMSON)), and all processing is done in parallel. Communication between the master module and function modules are performed using dual ported memories. The sensor parts of SONAVIS use a ultrasonic sensor produced by Poraloid Corporation and the CCD monochrome camera "WAT-906" produced by WATEC CO.,LTD. The maximum distance which is measured with this ultrasonic sensor is approximately 10 meters long, and, the ultrasonic beam width is about 20 degrees. The view area of this camera is approximately 50 degrees.

POEM II calculates it the estimated position and sends it to the vehicle control module(SPUR) every 5msec through a transputer link which is fast serial communication channel. In SPUR, robot is always controlled to follow the given path. Therefore, by correcting the estimated position, robot changes its path to follow.

The navigation process and SONAVIS process run on the master module. The navigation process controls the total robot's behavior for navigation. SONAVIS process communicates between the SONAVIS module and the navigation process. Therefore, the program uses SONAVIS as one module by a function call which sends data to the SONAVIS process. Poem II is also used by a function call which writes data to dual ported memory directly.

B. Experiment

Experiment was performed in the environment which includes the paved road lined with trees and the tiled road with trees and hedges. At first, robot runs the straight 50 meters long using tree landmarks. Then robot turns to the right and runs the straight 30 meters long using tree and hedge landmarks(Fig.9). Fig.1 is the photograph of this environment.

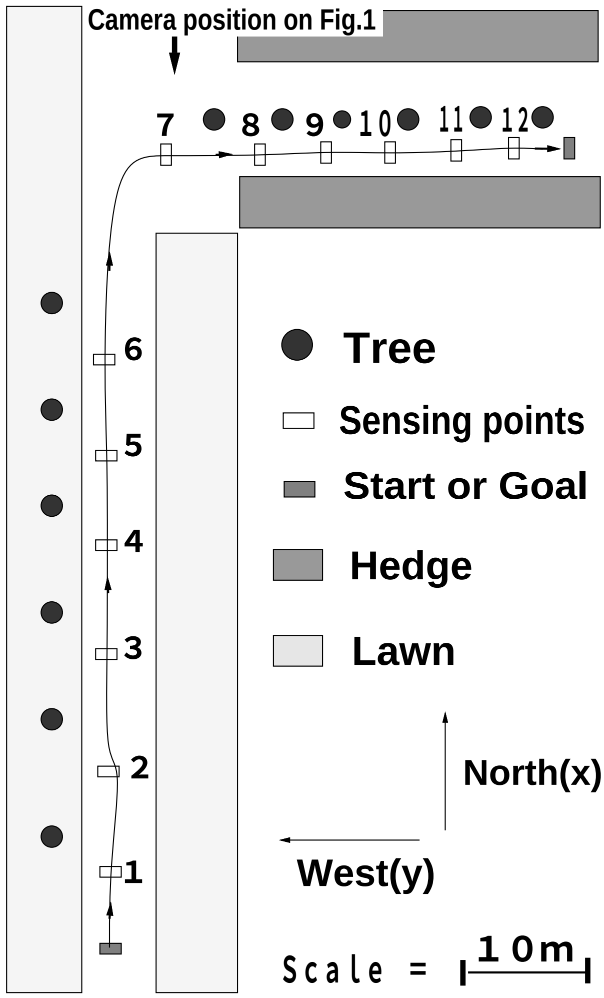

Fig. 9. Self-guidance of a mobile robot by using tree and hedge

Sensing points are priori decided and given to the robot. Tree positions measured on the sensing points are also priori measured with SONAVIS and given. The values of the deviation of errors are given as bellow.

$\sigma_r = 60_{[mm]}, \sigma_\alpha = 2_{[degree]}, (\sigma_{t_x}, \sigma_{t_y}) = (30_{[mm]}, 30_{[mm]})$

Trajectory of robot in the result of experiment is shown in Fig.9. Robot could reach on the goal point and the error was less than 10cm. Fig.10 shows the correction of robot position on each sensing point (2, 6, 7 and 11).

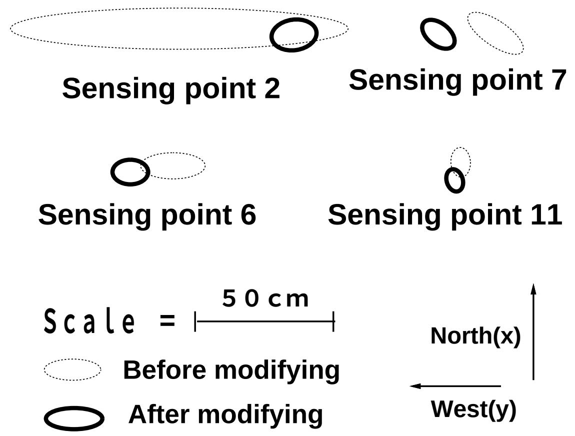

Fig. 10. Error ellipsoid for each modifying point(before and after modifying)

VI. Conclusion

We presented a method of mobile robot navigation using trees. We proposed a method for fusion of tree detection sensor and dead reckoning information for positioning. We have developed the sensor system combining sonar and vision to detect and measure trees. In the result of experiment of self-guidance of a mobile robot, it became obvious that the robot can be guided using trees as landmarks by the proposed method. And, it became clear that natural landmark such as tree is usefull to correct the robot position by the proposed method.

By experimentation, we realized that it is very difficult to make precise environmental map for navigation, which includes the exact locations of trees. Therefore, in our future work, we also should consider a half-automatic system for making environmental map with the locations of tree, while the operator manually guide the robot. By such means, it will be possible to achieve a long distance navigation by an autonomous mobile robot.

VII. acknowledgment

The authors express their gratitude to the members of Intelligent Robot laboratory of University of Tsukuba, and to Dr. Alex Zelinsky of the University of Wollongong, Australia for his useful suggestions in the preparation of this paper.

References

[1] E.Krotkov and R.Simmons "Autonomous Walking in Natural Terrain: A Retrospective on the Performance of the Ambler" Proc. of ISER, pp.85-90(1993)

[2] R.Chatila,S.Fleury,M.Herrb,S.Lacroix,C.Proust "Autonomous Navigation in Natural Environment" Proc. of ISER,pp.4-14(1993)

[3] K.Kluge,C.Thorpe "Representation and Recovery of Road Geometry in YARF" Proc. of Intelligent Vehicles,pp.114-119(1992)

[4] H.Ishiguro,K.Nishikawa,H.Mori "MOBILE ROBOT NAVIGATION BY VISUAL SIGN PATTERNS EXISTING IN OUTDOOR ENVIRONMENT" Proc. of IROS,pp.636-641(1992)

[5] K.Nishikawa,H.Mori "Rotation Control and Motion Estimation of Camera for Road Following" Proc. of IROS,pp.1313-1318(1993)

[6] K.Kimoto,S.Yuta "Sonar Based Outdoor Navigation" Proc. of ICAR,pp.239-244(1993)

[7] K.Komoriya, E.Oyama and K.Tani "Planning of Landmark Measurement for the Navigation of a Mobile Robot" Proc. of IROS vol.2 pp.1476-1481(1992)

[8] K.Nagatani,S.Yuta "Path and Sensing Point Planning for Mobile Robot Navigation to Minimize the Risk of Collision" Proc. of IEEE Int'l Conf. on IROS ,pp.2198-2203(1993)

[9] Y.Watanabe,S.Yuta "Position Estimation of Mobile Robots with Internal and External Sensors Using Uncertainty Evolution Technique" Proc. of IEEE Int'l Conf. on R & A ,pp.2011-2016(1990)

[10] A.Elfes "Using Occupancy Grids for Mobile Robot Perception and Navigation" Computer, vol.22, no.6, pp.46-57 (1989)

[11] A.M.Flynn "Combining Sonar and Infrared Sensors for Mobile Robot Navigation" Robotics Research ,vol.7,No.6, pp.239-244(1993)