Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 4
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima,
Soo-Young Lee, and Xin Yao (Editors)

A METHOD TO MEASURE 3D POSITIONS OF ELEVATOR BUTTONS FROM A MOBILE ROBOT USING A 2D ARTIFICIAL LANDMARK, A LASER NAVIGATION SYSTEM AND A COMPETITIVE NEURAL NET

S.Kurogi, Y.Fuchikawa, T.Ueno, K.Matsuo, and T.Nishida

Department of Control Engineering Kyushu Institute of Technology
Kitakyushu 804-8550, Japan

ABSTRACT

This article describes a new method to measure 3D positions of elevator buttons from the mobile guard robot which we have developed in order for the robot to autonomously get on and off the conventional elevators of a building for patrolling different floors. The measurement system uses a single camera on the robot and a 2D artificial landmark on the wall holding the elevator buttons, and it also takes advantage of the laser navigation system which enables the robot to stop facing the wall. Since the images captured by the camera mounted at the front body of the robot consisted of perspective patterns of the wall, we utilizes two neural schemes: one is the competitive neural net for pattern recognition invariant to projective transformations, and the other is the competitive learning scheme for vector quantization in order to obtain efficient template patterns. We also develop a method to calculate the 3D positions of the elevator buttons and examine the measurement errors for the robot to push the buttons.

1. INTRODUCTION

A number of vision systems to measure 3D positions have been developed for different goals, so they vary widely in many parameters such as accuracy, cost, size, stability, and so on. In a project of NEDO (New Energy and Industrial Technology Development Organization) of Japan, we have developed a mobile guard robot which autonomously gets on and off the conventional elevators of a building for patrolling different floors, where it is necessary to measure the relative position of a button from the robot for operating the elevator. Here, note that a lot of stereo-vision systems have been studied for measuring 3D positions and controlling robot manipulators (e.g. [1]), but they require two or more cameras. Moreover, several methods of localization using one camera and visual features in environment [2] are studied, but the accuracy is not so high. So, in our project, from the practical points of view such as size, acurracy, stability and cost, we have developed a new method to measure 3D positions of elevator buttons using a single camera and a 2D artificial landmark [3]. Although the method assumes that the robot and the wall holding the elevator buttons stand upright to the floor, the assumption does not fulfilled sometimes.

In this article, we present a modified method which utilizes the laser navigation system enabling the robot to stop facing the wall. Exactly speaking, two laser devices facing forward are placed parallel and horizontally at the bottom of the body and the horizontal two points on the wall from the devices can be equidistant. This however does not exclude the possibility that the robot and the wall tilt each other, namely the robot or the wall on the elevator sometimes tilt owing to the robot's weight (about 100kg) unbalanced on the four wheels. In order to utilize this property via the navigation system, we have modified the previous method [3]. First, owing to the tilt, the image captured by the camera mounted on the front body of the robot consists of the projective patterns of the features on the wall surface involving the buttons. So, we use the same neural schemes as in [3] although the training feature patterns are modified for the tilt. Here, note that the neural schemes used are the competitive neural net for pattern recognition invariant to projective transformations [4, 5] and the learning method for vector quantization (VQ) [6, 7] to store efficient template patterns. Although we use a single camera and a 2D artificial landmark as in [3], we should develop a new calculation algorithm because the constraint between the wall and the camera is different.

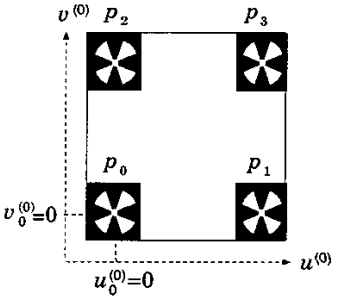


Figure 1: Artificial landmark consisting of four feature patterns p_0, p_1, p_2 and p_3 which are identical except the positions.

2. NOTATIONS

Suppose the camera captures a landmark on the wall involving elevator buttons. The landmark is 2D and consists of identical four feature patterns $p_i(\boldsymbol{u}^{(0)}) \triangleq p(\boldsymbol{u}^{(0)} + \boldsymbol{u}_i^{(0)})$ for $i \in I_f = \{0,1,2,3\}$ on the corners as shown in Figure 1, where $\boldsymbol{u}^{(0)} \triangleq (u^{(0)}, v^{(0)})^T$ indicates the position on the landmark plane, $p(\boldsymbol{u}^{(0)})$ is the original feature pattern, and $\boldsymbol{u}_i^{(0)}$ for $i = 0,1,2,3$ are the center positions of the lower-left, lower-right, upper-left, upper-right feature patterns, respectively. In addition to $\boldsymbol{u}_i^{(0)}$ $(i \in I_f)$ which we call feature positions of the landmark, the vectors $\boldsymbol{u}_i^{(0)}$ $(i \in I_b = \{4,5,\cdots\})$ indicate the center positions of the elevator buttons, where the values of $\boldsymbol{u}_i^{(0)}$ are previously measured relatively

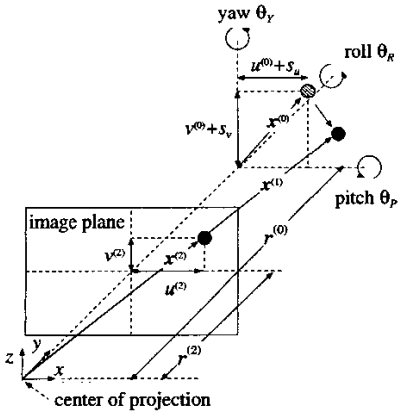


Figure 2: Relations of the positions.

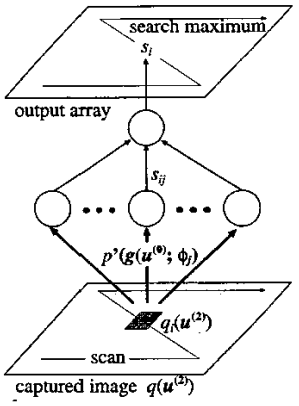


Figure 3: Feature extraction using a neural net.

to the origin being the lower-left feature position $\boldsymbol{u}_0^{(0)} \triangleq (0,0)^T$. As shown in Figure 2, suppose that the camera coordinate system consists of the axes x, y and z whose directions are rightward, forward and upward, respectively, from the camera. The position $\boldsymbol{u}^{(0)}$ on the landmark plane is extended to the 3D position $\boldsymbol{x}^{(0)} \triangleq (x^{(0)}, y^{(0)}, z^{(0)}) = (u^{(0)} + s_u, 0, v^{(0)} + s_v)^T$ which is orthogonal to the y-axis and $\boldsymbol{s} \triangleq (s_u, s_v)^T$ indicates the difference of the origins of $\boldsymbol{u}^{(0)}$ and $\boldsymbol{x}^{(0)}$. In Figure 2, we put the origin of $\boldsymbol{x}^{(0)}$ to $(0, r^{(0)}, 0)$, where $r^{(0)}$ is the distance of the wall to the center of projection of the camera along the y-axis.

Next, let us transform $\boldsymbol{x}^{(0)}$ to $\boldsymbol{x}^{(1)}$, which represents the position on the wall, by means of a pitch, a roll and a yaw rotation by the angles θ_P, θ_R and θ_Y, respectively, and then the translation by $\boldsymbol{r}^{(0)} = (0, r^{(0)}, 0)^T$. Here, we should consider the camera faces the wall as described above, then we have $\theta_Y = 0$ and the pitch followed by the roll rotation should be applied to $\boldsymbol{x}^{(0)}$. Thus, we have

$$\boldsymbol{x}^{(1)} = \boldsymbol{A}\boldsymbol{x}^{(0)} + \boldsymbol{r}^{(0)}, \tag{1}$$

where

$$\boldsymbol{A} \triangleq \boldsymbol{A}_{\theta_R}\boldsymbol{A}_{\theta_P} \triangleq \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta_R & \sin\theta_R \sin\theta_P & \sin\theta_R \cos\theta_P \\ 0 & \cos\theta_P & -\sin\theta_P \\ -\sin\theta_R & \sin\theta_R \sin\theta_P & \cos\theta_R \cos\theta_P \end{bmatrix}. \tag{2}$$

Here, note that the order of the rotations is important because we have $a_{10} = 0$ in this case, but we have $a_{01} = 0$ for $\boldsymbol{A} = \boldsymbol{A}_{\theta_P}\boldsymbol{A}_{\theta_R}$. Further, we have $a_{02} = 0$ for $\boldsymbol{A} = \boldsymbol{A}_{\theta_P}\boldsymbol{A}_{\theta_Y}$ as in [3], which gives the constraint to calculate the 3D positions of the buttons presented in Sect.4.

Last, let transform $\boldsymbol{x}^{(1)}$ to $\boldsymbol{x}^{(2)} \triangleq (x^{(2)}, r^{(2)}, z^{(2)})^T$ on the image plane of the camera by means of a perspective projection, where $r^{(2)}$ is the focal length or the distance of the image plane from the center of projection. Thus, we have

$$\boldsymbol{x}^{(2)} = k\boldsymbol{x}^{(1)}, \tag{3}$$

and the actual 2D image position $\boldsymbol{u}^{(2)} \triangleq (u^{(2)}, v^{(2)})^T = (x^{(0)}, z^{(2)})^T$. Thus, the relation from $\boldsymbol{u}^{(0)}$ to $\boldsymbol{u}^{(2)}$ is given by

$$\begin{aligned} \boldsymbol{u}^{(2)} &= \boldsymbol{g}(\boldsymbol{u}^{(0)} + \boldsymbol{s}; \boldsymbol{\phi}) \\ &\triangleq \frac{r^{(2)} \begin{bmatrix} a_{00} & a_{02} \\ a_{20} & a_{22} \end{bmatrix} \begin{bmatrix} u^{(0)} + s_u \\ v^{(0)} + s_v \end{bmatrix}}{(a_{10}, a_{12}) \begin{bmatrix} u^{(0)} + s_u \\ v^{(0)} + s_v \end{bmatrix} + r^{(0)}}, \end{aligned} \tag{4}$$

where $\boldsymbol{\phi} \triangleq (\boldsymbol{A}, r^{(0)})$ indicates the parameters of this transformation.

3. NEURAL SCHEMES FOR FEATURE EXTRACTION

The objective here is to obtain the center positions $\boldsymbol{u}_i^{(2)}$ of the transformed feature patterns $p_i(\boldsymbol{u}^{(2)}) = p(\boldsymbol{g}(\boldsymbol{u}^{(0)} + \boldsymbol{u}_i^{(0)} + \boldsymbol{s}; \boldsymbol{\phi}))$ on the camera image. From Eq.(4), we can see that $p_i(\boldsymbol{u}^{(2)})$ is also represented by $p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi}') + \boldsymbol{u}_i^{(2)})$ or a translated pattern of $p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi}'))$. So, we can find out $\boldsymbol{u}_i^{(2)}$ by searching $p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi}))$ on the captured image, say $q(\boldsymbol{u}^{(2)})$ and see Figure 3, where we encounter two problems; the first one is that the inner product of normalized $p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi})$ and an image segment of $q(\boldsymbol{u}^{(2)})$ as a standard matching index is not invariant to nonlinear transformations such as perspective projection when the patterns are not exactly the same owing to the lightning, the noise, the lens distortion and so on. The second problem is that we cannot use all possible parameter values $\boldsymbol{\phi}$ because $\boldsymbol{\phi}$ is continuous or there are infinite number of possible $\boldsymbol{\phi}$.

3.1. Invariant Pattern Recognition and Extraction

To overcome the first problem, we use the method of neural nets for invariant pattern recognition [4, 5] as follows; we first modify the pattern $p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi}))$;

$$p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi}) \triangleq p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi}))J(\boldsymbol{g}), \tag{5}$$

where $J(\boldsymbol{g})$ is the Jacobian given by

$$J(\boldsymbol{g}) \triangleq \left| \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}^{(0)}} \right| = \frac{r^{(0)}(r^{(2)})^2(a_{00}a_{22} - a_{02}a_{20})}{(a_{10}u^{(0)} + a_{12}v^{(0)} + r^{(0)})^3}. \quad (6)$$

Intuitively, the modification by Eq.(5) make the value of $p(\boldsymbol{g}(\boldsymbol{u}^{(0)}; \boldsymbol{\phi}))$ become smaller for the area enlarged by $\boldsymbol{g}$, which make the following matching index or the similarity be invariant to nonlinear transformations; let $q_i(\boldsymbol{u}^{(2)})$ be the segment of the captured image $q(\boldsymbol{u}^{(2)})$ given by

$$q_i(\boldsymbol{u}^{(2)}) \triangleq \begin{cases} q(\boldsymbol{u}^{(2)} + \boldsymbol{u}_i^{(2)}) & \text{if } \boldsymbol{u}^{(2)} \in D_i, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where D_i is a rectangular area whose center is $\boldsymbol{u}_i^{(2)}$. Then, the similarity s_{ij} between $q_i(\boldsymbol{u}^{(2)})$ and the modified feature pattern $p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi}_j)$ for a parameter $\boldsymbol{\phi}_j$ is calculated by

$$s_{ij} \triangleq \int \frac{q_i(\boldsymbol{u}^{(2)})}{\|q_i(\boldsymbol{u}^{(2)})\|} \frac{p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi}_j)}{\|p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi}_j)\|} d\boldsymbol{u}^{(2)}. \quad (8)$$

The similarity s_i at the position $\boldsymbol{u}_i^{(2)}$ is given by the maximum s_{ij} for all $\boldsymbol{\phi}_j$ (see Figure 3), and then we have the four feature positions $\boldsymbol{u}_i^{(2)}$ which has the largest s_i for all positions $i = 1, 2, \cdots, N$.

3.2. Learning Method for Vector Quantization

To overcome the second problem, we use the gradient and reinitialization learning (CRL) method for vector quantization (VQ) presented in [6, 7]; suppose the modified feature pattern $p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi})$ is represented by a k-dimensional vector, and let us denote the normalized $p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi})$ by $\boldsymbol{p}'_{\boldsymbol{\phi}} = p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi}) / \|p'(\boldsymbol{u}^{(2)}; \boldsymbol{\phi})\|$. Then, the objective of VQ with the weight vectors $\boldsymbol{w}_i$ for $i \in I_w = \{1, 2, \cdots, M\}$ is to minimize the distortion given by

$$D = \frac{1}{k} \sum_{i \in I_w} \int_{V_i} \|\boldsymbol{w}_i - \boldsymbol{p}'_{\boldsymbol{\phi}}\|^2 f(\boldsymbol{\phi})\, d\boldsymbol{\phi}, \quad (9)$$

where $f(\boldsymbol{\phi})$ is the probability density function, V_i is the Voronoi region given by

$$V_i = \{\boldsymbol{p}'_{\boldsymbol{\phi}} \mid \|\boldsymbol{w}_i - \boldsymbol{p}'_{\boldsymbol{\phi}}\| \le \|\boldsymbol{w}_j - \boldsymbol{p}'_{\boldsymbol{\phi}}\| \text{ for all } j \in I_w\}. \quad (10)$$

Here, note that $\boldsymbol{\phi}$ has three degrees of freedom with variables θ_P, θ_R, $r^{(0)}$, and then the whole pattern set $V = \{\boldsymbol{p}'_{\boldsymbol{\phi}} \text{ for all } \boldsymbol{\phi} \in \Phi\}$ forms a three dimensional manifold in the k-dimensional vector space. Although the discussion in [6, 7] basically is for the k-dimensional vector space, we proceed the analogy; we consider asymptotic situation where the number M of the weight vectors are very large, then we have the optimal condition that says D takes the minimum value when each subdistortion $D_i = (1/k) \int_{V_i} \|\boldsymbol{w}_i - \boldsymbol{p}'_{\boldsymbol{\phi}}\|^2 f(\boldsymbol{\phi})\, d\boldsymbol{\phi}$ takes the same value for all i. Then, using the variable d_i for estimating D_i, and the entropy H of all d_i given by

$$H = -\sum_{i \in I_w} \frac{d_i}{\sum_{j \in I_w} d_j} \ln \left(\frac{d_i}{\sum_{j \in I_w} d_j} \right), \quad (11)$$

the CRL algorithm is given by the following steps:

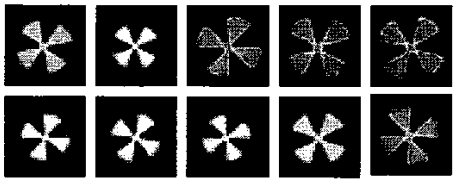


Figure 4: Weight vectors obtained by the CRL.

Step 1: (Reinitialization) Calculate the condition

$$d_c > r_d d\text{mean} \quad \text{and} \quad H < r_H \ln(N_c) \quad (12)$$

where d_c is of the cth weight vector which is closest to the input $\boldsymbol{p}'_{\boldsymbol{\phi}(t)}$, $d$mean is the mean value of all d_j, H, and $r_d (> 1)$ and $r_H (< 1)$ are constants. If the above condition holds, the following modification is applied

$$\boldsymbol{w}_i := \begin{cases} \boldsymbol{p}'_{\boldsymbol{\phi}(t)} & \text{if } i = l, \\ \boldsymbol{w}_i & \text{otherwise,} \end{cases} \quad (13)$$

$$d_i := \begin{cases} \eta d\text{mean} & \text{if } i = l \text{ or } i = c, \\ \eta d_i & \text{otherwise,} \end{cases} \quad (14)$$

where $:=$ indicates substitution, $\eta (< 1)$ is a constant, and l is the index of the weight vector $\boldsymbol{w}_l$ which has the minimum distortion d_l for all d_j.

Step 2: (Competitive Learning) If the above condition does not hold, the following modification is applied

$$\boldsymbol{w}_i := \begin{cases} \boldsymbol{w}_i + \alpha(t) \left(\boldsymbol{p}'_{\boldsymbol{\phi}(t)} - \boldsymbol{w}_i \right) & \text{if } i = c, \\ \boldsymbol{w}_i & \text{otherwise,} \end{cases} \quad (15)$$

$$d_i := \begin{cases} \eta d_i + \|\boldsymbol{p}'_{\boldsymbol{\phi}(t)} - \boldsymbol{w}_i\|^2 & \text{if } i = c, \\ \eta d_i & \text{otherwise,} \end{cases} \quad (16)$$

where $\alpha(t) = 1 - H / \ln(N_c)$ is the forgetting rate depending on the entropy H.

An example of weight vectors trained by the CRL method is shown in Figure 4, where, in the incremental learning, we have randomly chosen the parameter value $\theta_R[^\circ]$ from the range $[-30, 30]$, $\theta_P[^\circ]$ from $[-30, 30]$, and $r^{(0)}$[mm] from $[300, 900]$. The weight vectors in Figure 4 looks like the patterns that are modified by only magnifications, where we should note that we have also made an effort to design the feature pattern of the landmark to be unaffected by perspective projections in order to reduce the required memory and the calculation time while sustaining the same performance in invariant recognition. In other words, the weight vectors in Figure 4 are supposed to be the ones that minimize the distortion for the feature patterns modified with the above parameter values.

4. CALCULATION OF 3D POSITIONS

Here, we show the procedure to calculate the 3D positions $\boldsymbol{x}_i^{(1)}$ for $i \in I_b$ of elevator buttons from the values of the positions $\boldsymbol{u}_i^{(0)}$ for $i \in I_f \cup I_b$ on the landmark plane, the values of $\boldsymbol{u}_i^{(2)}$ for $i \in I_f$ on the camera image, and the focal length $r^{(2)}$.

We have placed the feature positions $\boldsymbol{u}_i^{(0)}$ on the landmark as follows;

$$\boldsymbol{u}_0^{(0)} = (0,0)^T \tag{17}$$
$$\boldsymbol{u}_0^{(1)} = (d_x,0)^T \tag{18}$$
$$\boldsymbol{u}_0^{(2)} = (0,d_z)^T \tag{19}$$
$$\boldsymbol{u}_0^{(3)} = (d_x,d_z)^T \tag{20}$$

where d_x and d_z are about 100mm long. Thus, we have the following relations.

$$x_0^{(2)} = k_0 a_x \tag{21}$$
$$r^{(2)} = k_0 a_y \tag{22}$$
$$z_0^{(2)} = k_0 a_z \tag{23}$$
$$x_1^{(2)} = k_1(a_{00}d_x + a_x) \tag{24}$$
$$r^{(2)} = k_1 a_y \tag{25}$$
$$z_1^{(2)} = k_1(a_{20}d_x + a_z) \tag{26}$$
$$x_2^{(2)} = k_2(a_{02}d_z + a_x) \tag{27}$$
$$r^{(2)} = k_2(a_{12}d_z + a_y) \tag{28}$$
$$z_2^{(2)} = k_2(a_{22}d_z + a_z) \tag{29}$$
$$x_3^{(2)} = k_3(a_{00}d_x + a_{02}d_z + a_x) \tag{30}$$
$$r^{(2)} = k_3(a_{12}d_z + a_y) \tag{31}$$
$$z_3^{(2)} = k_3(a_{20}d_x + a_{22}d_z + a_z) \tag{32}$$

where

$$a_x = a_{00}s_x + a_{02}s_z \tag{33}$$
$$a_y = a_{10}s_x + a_{12}s_z + r^{(0)} \tag{34}$$
$$a_z = a_{20}s_x + a_{22}s_z \tag{35}$$

Suppose the distance $r^{(2)}$ and the original positions $\boldsymbol{x}_i^{(0)}$ for $i = 1,2,\cdots$ are fixed and previously measured. Further, the positions $\boldsymbol{x}_i^{(2)}$ are supposed to be obtained by means of an image processing described in other section. Here, we have to obtain $\boldsymbol{x}_i^{(1)}$, or the values of transformation parameters A and $\boldsymbol{x}_s$, where it is important to examine robustness and precision of a lot of methods available because $\boldsymbol{x}_i^{(2)}$ are not so stable and fluctuate. As a result of such examinations, we have utilized the following method.

4.1. Calculation of parameter values

From Eq.(21) to Eq.(26), we obtain the estimated value $\widehat{\theta}_R$ of θ_R by

$$\widehat{\theta}_R := \tan^{-1}\frac{z_1^2 - z_0^2}{x_1^2 - x_0^2}, \tag{36}$$

which determines the values of $\widehat{a}_{00}$ and $\widehat{a}_{20}$ via Eq.(2). Further, from Eq.(21) to Eq.(26), we successively have

$$\widehat{a}_y := \frac{r^{(2)} d_x \widehat{a}_{00}}{x_1^{(2)} - x_0^{(2)}} \tag{37}$$
$$\widehat{a}_x := \frac{\widehat{a}_y x_0^{(2)}}{r^{(2)}} \tag{38}$$
$$\widehat{a}_z := \frac{\widehat{a}_y z_0^{(2)}}{r^{(2)}}, \tag{39}$$

and from Eq.(27) to Eq.(32), we have

$$\widehat{a}_{12} := \frac{-\widehat{a}_{00} d_x r^{(2)}}{d_z(x_2^2 - x_3^2)} - \frac{\widehat{a}_y}{d_z} \tag{40}$$
$$\widehat{a}_{02} := \frac{(\widehat{a}_{12}d_z + \widehat{a}_x)x_2^2}{d_z r^{(2)}} - \frac{\widehat{a}_x}{d_z} \tag{41}$$
$$\widehat{a}_{22} := \frac{(\widehat{a}_{12}d_z + \widehat{a}_x)z_2^2}{d_z r^{(2)}} - \frac{\widehat{a}_z}{d_z} \tag{42}$$

and then

$$\widehat{\theta}_P := -\tan^{-1}\frac{\widehat{a}_{12}}{\sqrt{\widehat{a}_{22}^2 + \widehat{a}_{02}^2}}, \tag{43}$$

which determines the values of $\widehat{a}_{01}$, $\widehat{a}_{11}$ and $\widehat{a}_{21}$ from Eq.(2). Finally, we have

$$\widehat{s}_x := \frac{\widehat{a}_{22}\widehat{a}_x - \widehat{a}_{02}\widehat{a}_z}{\widehat{a}_{00}\widehat{a}_{22} - \widehat{a}_{02}\widehat{a}_{20}} \tag{44}$$
$$\widehat{s}_z := \frac{-\widehat{a}_{20}\widehat{a}_x + \widehat{a}_{00}\widehat{a}_z}{\widehat{a}_{00}\widehat{a}_{22} - \widehat{a}_{02}\widehat{a}_{20}} \tag{45}$$

and

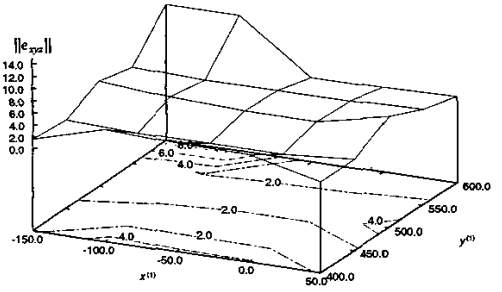
$$r^{(0)} := \widehat{a}_y - (\widehat{a}_{10}\widehat{s}_x + \widehat{a}_{12}\widehat{s}_z). \tag{46}$$

Finally, the estimated positions $\widehat{\boldsymbol{x}}_i^{(1)}$ for the actual 3D positions $\boldsymbol{x}_i^{(1)}$ are obtained from Eq.(1) with replacing $\boldsymbol{A}$, $\boldsymbol{s}$ and $r^{(0)}$ by the above calculated values and the previously measured $\boldsymbol{u}_i^{(0)}$.

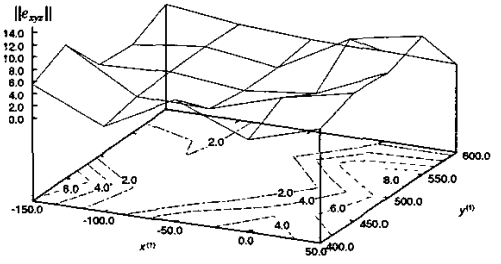
5. EXPERIMENTS

We have incorporated the above method to the vision system for the mobile guard robot, where we use a conventional camera (Sonny EVI-D30 with a wide conversion lens), a conventional image capture board (Hitachi IP5000) and a conventional notebook computer (Gateway Solo 5300).

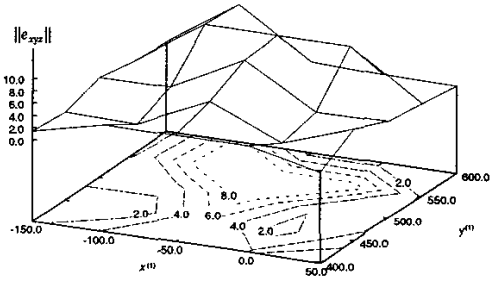
To evaluate the present visual measurement system, we have calculated the measurement error of the lower left feature position $\boldsymbol{x}_0^{(1)}$ for $x_0^{(1)} = -150, -100, -50, 0, 50, 100$, $y_0^{(1)} = 400, 450, 500, 550, 600$, and $z_0^{(1)} = 0$ (each unit is millimeter from here), and show the result in Figure 5, where we denote the error for each axis and their norm by $e_x = \widehat{x}_0^{(0)} - x_0^{(0)}$, $e_y = \widehat{y}_0^{(0)} - y_0^{(0)}$ $e_z = \widehat{z}_0^{(0)} - z_0^{(0)}$ and $\|e_{xyz}\| = \sqrt{e_x^2 + e_y^2 + e_z^2}$, respectively. Here, note that the center of the landmark on the $x^{(1)}$-axis is at $x^{(1)} = -50$ because we use the 2D landmark with the feature positions $\boldsymbol{u}_0^{(0)} = (0,0)^T$ $\boldsymbol{u}_1^{(0)} = (100,0)^T$ $\boldsymbol{u}_2^{(0)} = (0,100)^T$ and $\boldsymbol{u}_3^{(0)} = (100,100)^T$. In Figure 5(a), the error $\|e_{xyz}\|$ is small around $y^{(1)}$=500, where we have calibrated the camera parameters because the robot is expected to stop to push the buttons around the



(a) $\theta_P = 0°, \theta_R = 0°$



(b) $\theta_P = 0°, \theta_R = 30°$



(c) $\theta_P = 30°, \theta_R = 0°$

Figure 5: Distribution of the measurement error norm $\|e_{xyz}\|$. The dashed lines indicate the contour lines of the error surfaces.

position. From Figure 5(a), (b) and (c), we can see that the error for the nonzero angles θ_R and θ_P does not grow so much than the error for $\theta_R = 0$ and $\theta_P = 0$.

The maximum error is summarized in Table 1, where we can see that the errors $|e_x|$ and $|e_z|$ for the horizontal and the vertical directions, respectively, are less than 2.7 which is small enough relatively to the areas of the usual elevator buttons which are about 40×40. Although the error $|e_y|$ for the forward direction becomes 9.3, the robot can push the button because the finger to push the button has a flexible structure using a spring.

6. CONCLUSION

We have incorporated two neural schemes into the visual measurement system for the mobile guard robot to push the elevator buttons. By taking advantage of the laser navigation system, we have developed the system capable of dealing with the pitch and the roll rotations which sometimes occur in actual situation. Further, the previous method [3] is still useful to deal with the pitch and the yaw rotations when the robot and the landmark can stand upright. These methods are based on the architecture using a single camera and a 2D artificial landmark, which is superior in cost and size to the stereo-vision methods requiring several cameras, although they require laser navigation system or some assumptions.

Table 1: The maximum errors.

θ_P	θ_R	$\|e_x\|$	$\|e_y\|$	$\|e_z\|$	$\|e_{xyz}\|$
0°	0°	2.7	8.9	1.2	9.0
0°	30°	2.2	8.2	1.1	8.3
30°	0°	1.5	9.3	1.5	9.4

7. REFERENCES

[1] Nguyen, M.-C., "Stereo Vision-Based Robot Control Without Calibration", *Proc. VACETS Technical International Conference'97*, CD-ROM, 1997

[2] Sim, R. and Dudek, G., "Mobile Robot Localization from Learned Landmarks", *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Victoria, BC, v. 2,pp. 1060-1065.,1998.

[3] S.Kurogi, Y.Fuchikawa, T.Ueno, K.Matsuo and T.Nishida, "Visual measurement of 3D positions of elevator buttons from a mobile robot using a competitive neural net for pattern recognition invariant to projective transformations," *Proc. SCI2002*, 2002 (to appear).

[4] T.Nishida and S.Kurogi, "A multi-layered competitive net for pattern recognition invariant to coordinate transformations," *Journal JNNS (In Japanese)*, Vol.7, No.4, pp.106-114, 2000.

[5] T.Nishida and S.Kurogi, "An analysis of a multi-layered competitive net for invariant pattern recognition," *Proc. NNSP2000*, pp.346-355, 2000.

[6] T.Nishida, S.Kurogi and T.Saeki, "Adaptive vector quantization using re-initialization method," *Trans. IEICE D-II (In Japanese)*, Vol.J84-D-II, No.7, pp.1503-1511, 2001.

[7] T.Nishida and S.Kurogi, "An analysis of competitive and reinitialization learning for adaptive vector quantization," *Proc. IJCNN'01*, pp.978-983, 2001.

[8] S.Kurogi, T.Nishida and K.Yamamoto, "Image transformation of local features for rotation invariant pattern matching," *Proc. ICONIP-2001*, pp.693-698, 2001.