Position and Pose Adjustment of Camera with Foveated Wide-angle Lens

Nobuyuki Kita Hai-quan Yang
Yasuyo Kita
Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology

A method is proposed to correct the position and pose of a camera-head by aligning a 3D model of its surrounding environment with an observed 2D image that is captured by a foveated wide-angle lens in the camera. Because of the wide field of view of the lens, the algorithm can converge even when the initial error is large, and the precision of the result is high since the resolution of the fovea of the lens is high.
Keywords: alignment, registration, 3D model, mobile robot, plant

1 Introduction

It is quite important to know the position and pose of the camera accurately for variable applications[1] [2]. When a 3D model of the environment surrounding a camera is provided, it is possible to know the position and pose of the camera by aligning the 3D model with the observed image. Some methods have been proposed for 3D-2D alignment using the proximity between the observed 2D points and the projections of the 3D model points [3] or the proximity between the projection rays from the observed 2D points and the 3D model points [4] [5]. These methods assumed the ability to detect identical features from both the observed images and 3D model, such as corners or roof edges. They could not be applied to cases in which there were only curved surfaces in the environment. This difficulty was overcome by the method proposed in [6] which enabled the use of occluding edges as features. That method was successful in robust 3D-2D alignment even in complicated scenes by using the two-type predicted image effectively which are calculated from the 3D environmental model by a graphics system (e.g. OpenGL etc).

In this study, the method [6] is extended to deal with a foveated wide-angle lens [7] suitable for the many visual functions. The camera's wide field of view means that the observed image has more features than when a perspective lens is used. This is very important for aligning the 3D model with the observed image, especially when the initial error is large. When the iterations almost converge to the result, the high resolution of the fovea is used to make the result more accurate.

An outline of the method in [6] and the design and simulation of a foveated wide-angle lens is given in Section 2 and 3. Our proposed method is described in section 4. Our study's experimental results and evaluations of this proposed method are given in Section 5 using simulated images. The last section concludes this paper.

2 Overview of registration

The basic scheme of 3D-2D alignment using a perspective lens [6] is reviewed in this section. It uses occluding edges as features to align the 3D model with an observed image. Suppose that an image is observed by a camera whose initial position and pose are known (e.g. from dead reckoning data). Because of estimation error, the projection of the environment model on the observed image is deviated. Concrete procedures for correcting this deviation are as follows:

i) Calculation of 3D model points

The 3D model points corresponding to the observed edges are calculated from the 3D environment model and the initial estimated state of the camera.

This process is done quickly by reading the 3D coordinates of the occluding edge points of the depth image calculated by a graphics system such as OpenGL.

The intensity image calculated by the graphics system is used to remove any model points whose strength is weak in the intensity image. The maximum gradient directions in the intensity image around the projection of the model points are used as their directional attributes which are classified into 8 directions. When a perspective lens is used, the di-

rectional attributes do not change with the 3D translation and rotation along the non-view axis. It is a robust attribute of the edge point.

ii) 3D-2D point matching

The observed edge points, which are calculated by the canny operator, corresponding to 3D model points are determined based on the proximity to the observed image and the consistency of the directional attributes. The maximum gradient directions in the observed image around the observed edge are used as their directional attributes which are also classified into 8 directions. Only points with the same directional attributes can be paired.

Since a small change in the camera angle causes a large translation in the image, only the first time before 3D-2D matching, the projected 3D model points are 2D-translated on the image to the position where the model points overlap best on observed edges with the same direction attribute. Territory-based 3D-2D matching[8] which uses an isotropic search on restricted regions obtained from the projected shape of the model enables a high ratio of correct pairs.

iii) Calculation of 3D transformation

The method used in this step was proposed by Heuring et. al. in [9].

Consider the rotation R and translation t that transform a 3D model point $P_i = (X_i, Y_i, Z_i)$, referred in a local object-based frame which is the initial frame of the camera, into a position in the target camera frame. The point in the target camera frame must lie along the back projected ray from the observed image position $p_i = (x_i, y_i, f)$, so that

$$RP_i + t = \lambda_i p_i, \quad i = 1, \ldots, m. \tag{1}$$

where λ_i is unknown.

Obviously, $R(P_i - P_j) = \lambda_i p_i - \lambda_j p_j$, so the translation can be eliminated by confining $R(P_i - P_j)$ merely to lie in the plane spanned by vectors p_i and p_j. This leads to

$$(p_i \times p_j) \cdot R(P_i - P_j) = 0, \tag{2}$$

for $i = 1, \ldots, m-1, j = i+1, \ldots, m$. The optimal rotation can be found by minimizing

$$\min_R \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} ((p_i \times p_j) \cdot R(P_i - P_j))^2. \tag{3}$$

Once the rotation R is known, the translation can be calculated by solving $2m$ simultaneous equations on (t_x, t_y, t_z):

$$\frac{f(X_i' + t_x)}{Z_i' + t_z} = x_i \quad \frac{f(Y_i' + t_y)}{Z_i' + t_z} = y_i \tag{4}$$

where $P_i' = (X_i', Y_i', Z_i') = RP_i$.

iv) Iteration

In step ii), the matching pairs may be incorrect because the 3D point, which corresponds to an observed edge point, may be occluded and invisible in the predicted view. The linear approximation adopted in iii) to get the rotation matrix also leads to some errors. By iterating the process from i) to iii), the predicted view converges to the observed image and the correct position and pose of the camera are achieved.

3 Foveated wide-angle lens

One of the authors et. al. proposed a novel design and implementation of a foveated wide-angle lens[7]. It is most suitable for active-vision applications in complex and dynamic environments.

3.1 Design

The core part of this lens design is the projection curve shown in Figure 1, which maps the incident angle ϕ of an optic ray entering the lens system to an image height h on the CCD surface, and both are measured from the optical axis. The curve consists of three parts, a, b and c, which are represented by the following equations respectively.

$$h = f_1 \tan\phi, \quad 0 \le \phi \le \phi_1 \tag{5}$$

$$h = \log_a(f_2\phi) - p, \quad \phi_1 \le \phi \le \phi_2 \tag{6}$$

$$h = f_3\phi + q, \quad \phi_2 \le \phi \le \phi_{\max} \tag{7}$$

a is tangent representing perspective part, c is linear representing spherical projection, and b is logarithmic and connects a and c.

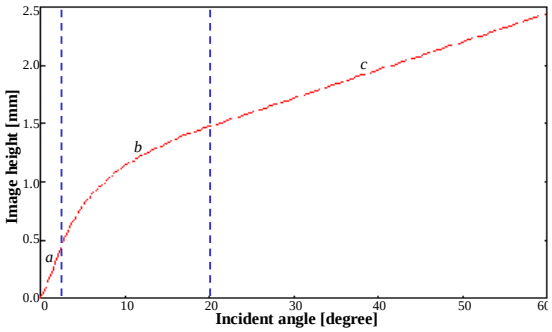


Figure 1: Projection curve

As shown in Figure 2, the resolution of central part is high and oppositely low in the periphery. The

field of view is 120 degrees. Figure 3 shows an actual image taken with the foveated wide-angle lens for our experimental mock-up of a nuclear power plant.

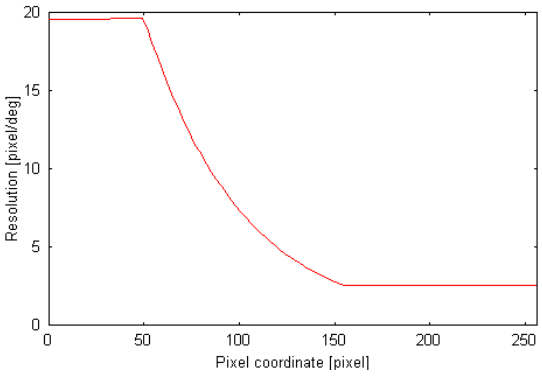


Figure 2: Spatial resolution curve

Figure 3: An actual image from the lens

3.2 Simulation in virtual environment

Wide field of view is generated from normal perspective views synthesized by a conventional graphics system like OpenGL[10]. The basic concept is dividing the wide field of view of a generated image into several views each of which is covered by a perspective view and calculating the pixel value of the generated image from the corresponding pixels values in the perspective images that are generated by a conventional graphics system.

According to the projection curve, the 4 vertices of a pixel in a generated image are mapped onto 4 points in a perspective image as shown in Figure 4. The intensity and the depth values of the original pixel, p and d, are calculated from averaged pixel values surrounded by P_s^1, P_s^2, P_s^3 and P_s^4.

Figure 5 is a generated intensity image and Figure 6 is a generated depth image.

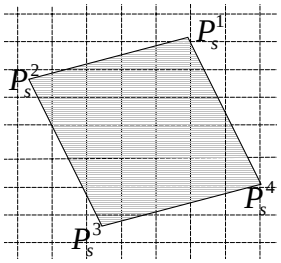


Figure 4: Transform a pixel from a generated image into a perspective image

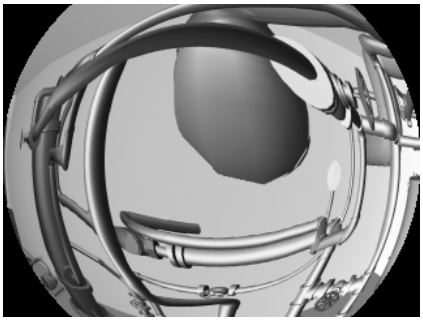


Figure 5: A generated intensity image

4 Proposed method

Please see the Appendix for notations used in this section.

4.1 Calculation of 3D model points

Since the current 3D model includes walls behind the pipes, the fixed depth threshold is introduced to detect occluding edges. If the threshold is set not to loose occluding edges made by pipes near the walls, over estimation becomes possible, especially in the

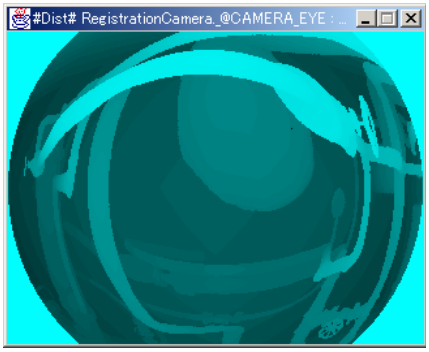


Figure 6: A generated depth image

periphery because the low resolution exaggerate the depth difference between the next pixels even on the same surface. To reduce this over estimation, only the depth edge is selected when it is also intensity edge and both directions are the same. The model points are selected from the edge points by constant sampling. Actually, the model points are shifted one pixel towards the opposite direction of depth derivative so as to avoid the unreliability of depth values on occluding edges. The dotted points show the selected 3D model points in Figure 7.

4.2 3D-2D point matching

In the case of warped projection as shown in Figure 1, pure translation causes changes of edge direction. The comparison of directions between edge points is performed after the following conversion. (d_1, d_2) is a original direction and (e_1, e_2) is the direction on the virtual perspective projection plane (See appendix).

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{8}$$

As same as previous algorithm, for the first iteration, pure 2D translation is adopted for the correct point pairs to get close. In the case of perspective projection, just translation of model points on image plane was enough, but it is not in the case. Then the possible 2D translations of model points are performed on the virtual perspective projection plane and the best translation is selected when the model points after projected back onto the foveated projection plane overlap best to input edges.

Since the central area of the lens has high resolution, the 2D position error between the estimated image and the observed image in this area is large, even after the first iteration. Figure 7 shows the matching pairs after the first iteration. The color points in image B are the model points in the matching pairs, and the color represents their directions. The color line segments in image A represent the correspondence relation of the pairs. One endpoint of the line segment is a observed edge point and the other represents the position of the corresponding model point. Figure 7 shows that the matching pairs in the center portion are not correct.

4.3 Calculation of 3D transformation

For the 3D transformation calculation, the same method with the previous algorithm is used after converting the observed edge points in the point pairs onto the virtual perspective projection plane. But the consideration about the different quantized errors caused by variable resolution is necessary. When the camera pose estimation gets close to the correct one, the point pairs in the periphery coincides completely even the pairs around fovea still have some distances. To reduce this negative effect, the equations for the pairs around fovea are given higher weights according to the resolution when conversion is progressing. The discussion about the partially different quantized errors will be made in the different place.

4.4 Iteration

To get a good convergence, the iteration process is split into 4 stages according to the residual error. To decide when a stage or the whole iteration should be terminated, some criteria are necessary.

4.4.1 Criteria to evaluate the convergence

The following quantities are considered.

Pairs ratio is the ratio of the number of point pairs to the number of model points. This tends to 1 while converging.

Pairs error is the average distance between the observed points and the model points forming pairs. This tends to 0 while converging.

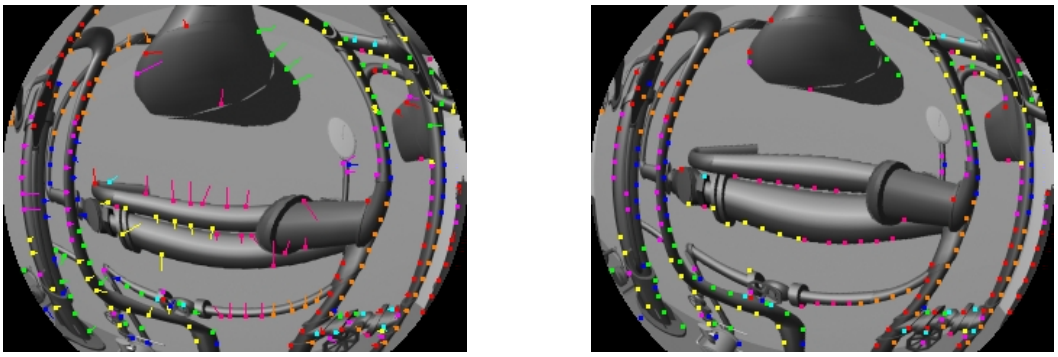
Practically, another quantity, which mixes the pairs ratio and the pairs error is also used, and it is called mixed quantity.

4.4.2 Four stages of iteration

The first stage contains only the first iteration. The 2D translation is performed before the 3D-2D matching and the original 3D transformation method is used.

In the second stage, only the model points in the outer portion of the image are used to avoid the affect of mismatches at the center, and the method to calculate the 3D transformation is the same as the first stage. In the following experiments, the outer portion is the outside of a disc whose radius is 80 pixels in the image of 320×240 pixels. This also facilitates to reduce the amount of calculation. This stage terminates when the pairs error seems to increase for the next iteration below a constant C_1. If the pairs error can not reach C_1 within the limited iterations, the whole algorithm stops and reports a "failure".

The third stage uses all of the model points and the weighted equations for the calculation of the 3D



A. Image from observed state B. Image from an estimated state

Figure 7: Matching pairs when error is large

transformation. After the second stage, the pixel errors of the matching pairs in the outer portion will be small, and this makes the step size of the 3D transformation very small or even 0, so the method in Subsection 4.3 is effective for obtaining a large step size and speeding up the iteration. Even after most of the pixel errors of the matching pairs in the outer portion become 0, the method still enables convergence. If the mixed quantity is less than a constant C_2 at some iteration, the algorithm enter the next stage before the mixed quantity increases after that iteration.

In the last stage, for getting subpixel accuracy, only matching pairs whose pixel errors are 0 or 1 are considered. Here the calculation of 3D transformation is very sensitive to the wrong pairs. To reduce the sensitivity, in this stage the original non-weighted method is used for the 3D transformation calculation. The terminate condition is the same as the third stage with another constant $C_3 < C_2$.

A maximum number of iterations is set for each stage. If the iterations reach the maximum in the third and the last stage, the algorithm finishes and reports a "success" with information about the pairs ratio and pairs error that represents the possible precision.

5 Experiments and evaluation

To avoid several difficulties when using an actual foveated image, such as calibration of the lens, inaccurate 3D model and so on, the method was evaluated using simulated images. Figure 8 shows the 3D model used in our experiments. The three arrows show the virtual camera positions and poses used in the following experiments. For the initial estimation, systematic errors were intentionally added to the state of the simulated observed image.

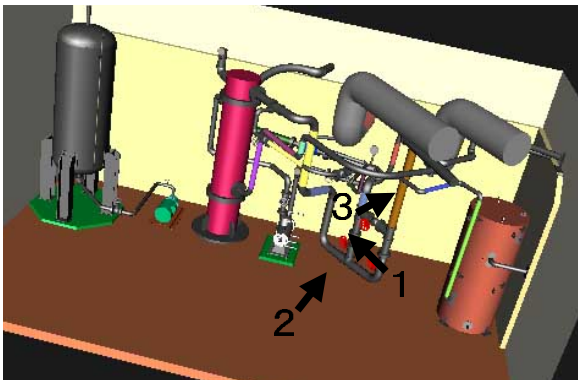
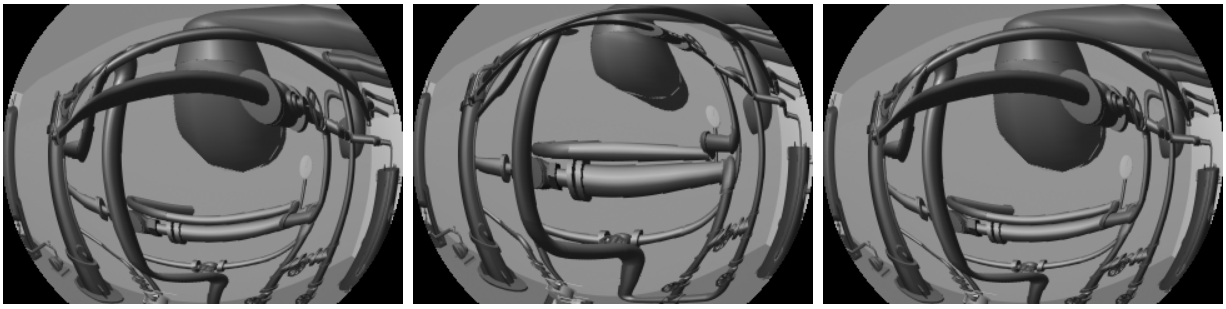


Figure 8: 3D model of the mockup and three camera positions in the experiments

Figure 9 shows a typical example of the experiments. In Figure 9, image A is generated by the simulation system of the foveated wide-angle lens to simulate an observed image, and the error of the initial state is 0.2 meter along the vertical axis of the "observed" state. Image B is the generated image from the initial state. The image from the resultant state is shown by image C. The absolute translation error between the resultant state and the observed state is 0.002095 meter, and the angle between the view axis of the resultant state and that of the observed state is 0.079129 degree.

To evaluate the accuracy of convergence with a single value, the 2D positions of a 3D point located at $P_w = (0, 0, -1)$ referred to the observed frame, whose 2D position in the observed image is the center, were calculated. Because the resolution of the center is the highest among all the pixels in the image, the 2D coordinate of this point is most sensitive to 3D errors of the position and pose of the camera. Figure



A. Image from observed state B. Image from initial state C. Image from resultant state

Figure 9: An example of experiment

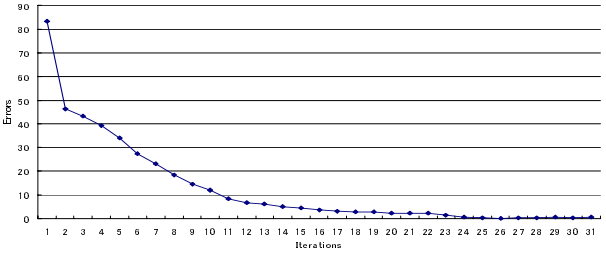


Figure 10: 2D errors

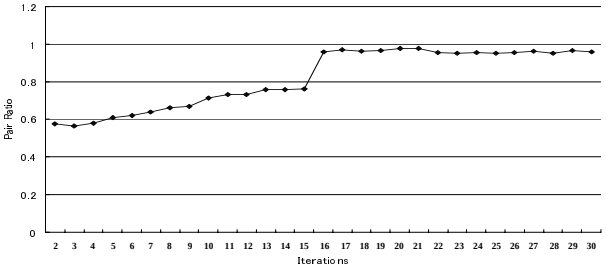


Figure 11: Pairs ratio

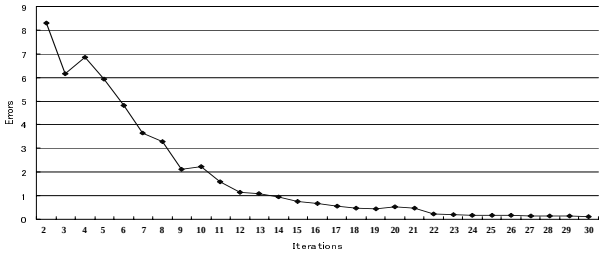


Figure 12: Pairs error

10 shows this value for each iteration. The second stage was iteration $2 \sim 16$, the third stage $17 \sim 22$, and the last stage $23 \sim 30$. The error 1.612425 pixels at the end of the third stage was reduced to 0.582614 pixel after the final stage.

Figures 11 and 12 show the quantities changes of the proposed criterion. The constant C_1 was 2.0, C_2 was 0.8 and C_3 was 0.5. These figures show that the proposed criterion give reasonable estimation for the convergence.

Initial Errors						Final Errors						
Translation(m)			Rotation(deg.)			Translation(m)			Rotation(deg.)			
r	u	v	R	U	V	r	u	v	R	U	V	
0	0	0	0	0	0	0.000299	0.000410	-0.000309	0.000000	0.027976	0.027976	○
-0.2	0	0	0	0	0	0.000638	0.000134	-0.001116	0.034264	0.039565	0.027976	○
0.2	0	0	0	0	0	-0.001769	-0.000296	-0.000848	0.055953	0.019782	0.055953	○
0	-0.2	0	0	0	0	-0.001737	-0.000077	0.000436	0.059347	0.034264	0.052339	○
0	0.2	0	0	0	0	-0.000725	0.000237	-0.000400	0.039565	0.034264	0.034264	○
0	0	-0.2	0	0	0	-0.000269	-0.000257	-0.000554	0.027976	0.027976	0.000000	○
0	0	0.2	0	0	0	0.000238	0.000093	-0.001765	0.019782	0.027976	0.019782	○
0	0	0	0	-10	-10	-0.002216	0.000771	-0.000135	0.076617	0.065611	0.071326	○
0	0	0	0	10	10	-0.004147	-0.000514	-0.002652	0.123541	0.048457	0.117034	○
0	0	0	-10	0	-10	0.000511	-0.000323	-0.000186	0.034264	0.027976	0.027976	○
0	0	0	10	0	10	-0.000820	0.000688	-0.001383	0.039565	0.052339	0.048457	○

Table 1: Experimental results for position 2 by foveated lens

Table 1 shows the results when some errors were

Initial Errors						Final Errors						
Translation(m)			Rotation(deg.)			Translation(m)			Rotation(deg.)			
r	u	v	R	U	V	r	u	v	R	U	V	
0	0	0	0	0	0	-0.020086	0.017328	0.007625	0.693798	0.798194	0.923233	○
-0.2	0	0	0	0	0	0.275708	-0.006470	-0.037303	7.264094	1.249135	7.157961	X
0.2	0	0	0	0	0	-0.023712	0.015363	0.009204	0.744147	0.677242	0.928095	○
0	-0.2	0	0	0	0	0.000391	0.239806	0.019740	1.362431	6.182979	6.031402	X
0	0.2	0	0	0	0	-0.044494	-0.246846	0.035641	1.919589	6.328246	6.365688	X
0	0	-0.2	0	0	0	-0.016637	0.013869	0.015999	0.589503	0.602634	0.773037	○
0	0	0.2	0	0	0	0.015971	0.037181	-0.143248	1.362287	2.083469	2.452799	X
0	0	0	0	-10	-10	-0.024920	0.015287	0.010432	0.787582	0.614212	0.948120	○
0	0	0	0	10	10	-0.019244	0.016716	0.010660	0.593802	0.706375	0.857750	○
0	0	0	-10	0	-10	-0.030022	0.016120	0.009689	0.971363	0.688986	1.142096	○
0	0	0	10	0	10	-0.028131	0.014493	0.009944	0.932092	0.625263	1.060171	○

Table 2: Experimental results for position 2 by perspective lens

added to the observed state 2 in Figure 8. The translation errors (r, u, v) refer to the local frame of the camera in the observed state, v is along the view axis, r is along the horizontal axis, and u is along the vertical axis. Also, the rotation errors R, U, V are the angles between the horizontal axis, the vertical axis, the view axis of the initial state (or final state) and those of the observed state. The distance between the camera and the closest pipe for this position was about 0.8 meter. The algorithm converges to resultant states whose 3D errors were very small for all of the initial states, and 2D errors like those in Figure 10 for all the resultant states were less than 1 pixel. On the other hand, Table 2 shows the experimental results using a perspective lens after 20 iterations. There were several cases, marked by "X", in which the algorithm did not converge. Even for the convergences, marked by "◯", the 3D errors were much larger than those in Table 1, and the 2D errors were all greater than 3 pixels.

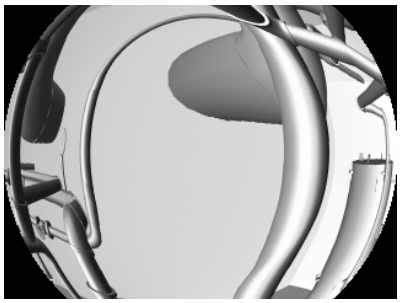


Figure 13: Observed image from position 3

Figure 13 shows the observed image from position 3. The result is shown in Table 3. The distance between the camera and the closest pipe for this position was less than 0.2 meter. The "F" in Table 3 means a "failure" was reported by the algorithm. Table 4 shows the experimental results usinng the perspective lens. The final errors in Table 3 for which a "success" was reported were much smaller than the final errors shown in Table 4 using the perspective lens.

These experiments show that the 3D models were well aligned with the observed images and that 3D localization converged if the initial position error was less than 0.2 meter and the initial error of the pose was less than 10 degrees when the observed image contained adequate information. The resultant translation and rotation errors were almost less than 0.002 meter and 0.1 degree, respectively. The permitted initial error was larger than that when using the perspective lens, and the precision of the result

Initial Errors						Final Errors						
Translation(m)			Rotation(deg.)			Translation(m)			Rotation(deg.)			
r	u	v	R	U	V	r	u	v	R	U	V	
0	0	0	0	0	0	0.002111	0.001055	0.000740	0.121299	0.103489	0.159655	◯
-0.2	0	0	0	0	0	-0.000153	-0.000444	-0.000003	0.040592	0.085099	0.085635	◯
0.2	0	0	0	0	0	-0.000116	-0.000194	-0.000043	0.056842	0.094517	0.090821	◯
0	-0.2	0	0	0	0	F	F	F	F	F	F	
0	0.2	0	0	0	0	F	F	F	F	F	F	
0	0	-0.2	0	0	0	0.000744	-0.000469	0.000393	0.108401	0.093085	0.108672	◯
0	0	0.2	0	0	0	F	F	F	F	F	F	
0	0	0	0	-10	-10	0.000675	-0.000096	0.000873	0.079557	0.115445	0.137768	◯
0	0	0	0	10	10	0.002910	0.002317	-0.001770	0.274515	0.210635	0.275274	◯
0	0	0	-10	0	-10	F	F	F	F	F	F	
0	0	0	10	0	10	0.000620	0.000933	0.000071	0.113893	0.158517	0.151312	◯

Table 3: Experimental results for position 3 using foveated lens

Initial Errors						Final Errors						
Translation(m)			Rotation(deg.)			Translation(m)			Rotation(deg.)			
r	u	v	R	U	V	r	u	v	R	U	V	
0	0	0	0	0	0	-0.001239	0.036729	0.013146	0.804533	1.659791	1.453106	◯
-0.2	0	0	0	0	0	F	F	F	F	F	F	
0.2	0	0	0	0	0	-0.369607	-0.136128	-0.203297	12.96971	6.853064	14.00386	X
0	-0.2	0	0	0	0	0.010898	0.192849	-0.003499	2.799923	7.153085	6.638737	X
0	0.2	0	0	0	0	-0.002023	-0.156225	0.003785	1.998902	5.670861	5.321828	X
0	0	-0.2	0	0	0	0.003860	-0.007434	0.006543	0.356712	0.362623	0.501547	◯
0	0	0.2	0	0	0	0.044716	0.071801	-0.195353	5.342638	4.636064	3.538404	X
0	0	0	0	-10	-10	-0.000557	0.034029	0.011993	0.768689	1.547559	1.343445	◯
0	0	0	0	10	10	0.002194	0.010788	0.009396	0.376121	0.470306	0.429461	◯
0	0	0	-10	0	-10	0.001432	0.022131	0.010552	0.448814	0.937595	0.853929	◯
0	0	0	10	0	10	0.000780	0.018287	0.011266	0.485524	0.816656	0.675045	◯

Table 4: Experimental results for position 2 using perspective lens

was much higher than that when using the perspective lens. To acquire adequate information, the position of the camera should not be too close to or too far away from the pipes, and the pipes should be distributed within the field of view.

The number of iterations for the examples in Table 1 were 15 to 40. The average processing time for these was 66.43 seconds by 500Hz Dual Pentium 3 machine. 54.49 seconds within this time was spent to receive intensity and depth images synthesized by other PCs. The calculation time for 20 iterations for the perspective lens was less than 4 seconds on the same Pentium 3 machine.

6 Summary

A method was proposed to correct the position and pose of a camera-head by aligning a 3D model of its surrounding environment, which was a nuclear plant mock-up in our case, with an observed 2D image captured by a foveated wide-angle lens in the camera. Because the field of view of the lens was wide, the algorithm could converge even with large initial errors, and the precision of the result was high since the resolution of the fovea of the lens was high.

After some preparations, such as carefully cali-

brating an actual foveated wide-angle lens, our proposed method will be used for actual images.

The calculation time is longer than when using a perspective lens. One reason is that it takes a long time to render the images. We will discuss how to shorten the rendering time in another paper.

Acknowledgements: This study was financially supported by the Budget for Nuclear Research of the Ministry of Education, Culture, Sports, Science and Technology, based on the screening and counseling by the Atomic Energy Commission.

References

[1] Kita, N., Kuniyoshi, Y., Hara, I., Matsui, T., Hori, T., Hirai, S., Davison, A.J., Rougeaux, S: Mobile Sensing, Robots for Nuclear Power Plant Inspection, *Advanced Robotics*,vol. 13, no. 3, pp. 355–356. 1999.7.

[2] Kita, N., Kita, Y. and Yang, H: Archiving Technology for Plant Inspection Images Captured by Mobile Active Cameras - 4D Visible Memory - , *(3DPVT '2002)*, pp. 208–213, 2002.

[3] Feldmar, J., Malandian, G., Ayache, N., Fernandezvidal, S.,Maurincomme, E. and Trousset, Y.: Matching 3D MR Angiography Data and 2D X-ray Angiograms, *Proc. of CVRMed-MRCAS'97*, pp. 129–138, 1997.

[4] Lavallee, S. and Szeliski, R.: Recovering the Position and Orientation of Free-form Objects from Image Contours Using 3D Distance Maps, *IEEE trans. Pattern Anal. & Mach. Intell.*, 17.4., pp. 378–390, 1995.

[5] Wunsch, P. and Hirzinger, G.: Registration of CAD-Models to Image by Iterative Inverse Perspective Matching, *Proc. of 13th International Conference on Pattern Recognition*, pp78–83,1996.

[6] Kita, Y. and Kita, N.: Position and Pose Detection of Active Camera-head in a Nuclear Power Plant, *Proc. of Int. Conf. on Intelligent Robots and Systems(IROS '2000)*, pp. 1872–1879,2000.

[7] Kuniyoshi, Y., Kita, N., Sugimoto, K., Nakamura, S., and Suehiro, T.: A Foveated Wide Angle Lens for Active Vision, *Proc. of the 1995 IEEE International Conference on Robotics and Automation(ICRA '1995)*, pp. 2982–2988,1995.

[8] Kita, Y., Wilson, D. L., Noble, A. and Kita, N.: A Quick 3D-2D Registration Method for a Wide-Range of Applications, *Proc. of 15th Int. Conf. on Pattern Recognition(ICPR '2000)*, pp. 981–986,2000.

[9] Heuring, J. J. and Murray, D. W.: Visual Head Tracking and Slaving for Visual Telepressence, *Proc. of the 1996 IEEE International Conference on Robotics and Automation(ICRA '1996)*, pp. 2908–2914,1996.

[10] Kita, N.: Next Gaze Point Decision in the Foveated View Field, *CVIM 133-25, May 9th. 2002*, pp. 185–192,2002. (in Japanese)

Appendix: Suppose the projection curve is denoted as a function $h = h(\phi)$ and its inverse as $\phi = \phi(h)$. The relation between coordinates on foveated image plane and virtual perspective image plane for a the same 3D point is depicted in Figure 14 and represented by the following equations.

$$\begin{array}{rcccl} u & = & u(x,y) & = & \cos(\psi)h(\phi), \\ v & = & v(x,y) & = & \sin(\psi)h(\phi), \end{array} \tag{9}$$

Here, $\phi = \arctan(\sqrt{x^2+y^2})$ and $\psi = \arctan(y/x)$.

$$\begin{array}{rcccl} x & = & x(u,v) & = & \frac{u}{h}\tan(\phi(h)), \\ y & = & y(u,v) & = & \frac{v}{h}\tan(\phi(h)), \end{array} \tag{10}$$

Here, $h = \sqrt{u^2+v^2}$.

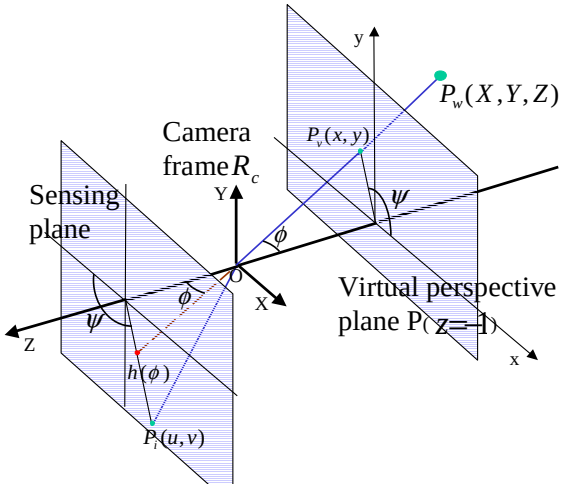


Figure 14: Notation of projection