Position and pose detection of active camera-head in a nuclear power plant

Yasuyo Kita **Nobuyuki Kita**

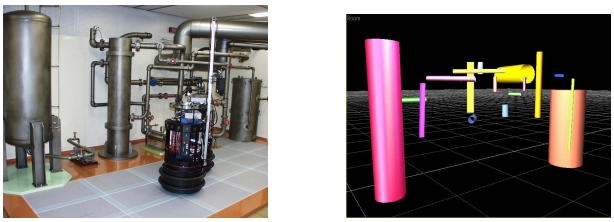
Research Institute of Intelligent Systems,
National Institute of Advanced Industrial Science and Technology (AIST)
y.kita@aist.go.jp n.kita@aist.go.jp

Abstract

A method to determine the position and pose of an active camera-head by aligning a 3D model of its surrounding environment with an observed 2D image is proposed. The camera-head is mounted on a mobile robot and freely moves in a 3D space. We aim at visual feedback to correct the estimation error of its position and pose obtained from dead reckoning. Since the nuclear power plant where the robot moves about consists of many pipes without particular marks, most of features in the observed images are occluding edges of the pipes. For robustly finding 3D-2D point correspondences on the occluding edges, two-type predicted images which are calculated from the 3D environmental model by a graphics system (eg. OpenGL etc) are used as follows: 1) 3D model points which correspond to the observed occluding edges are quickly obtained from the predicted depth image; 2) The predicted intensity image is used to select only the 3D model points which are expected to appear clearly in the observed image. As a result, point correspondences between the observed image and the 3D model can be robustly found even in complicated scenes. Preliminary experiments using actual plant mock-up have shown that the method is promising.

1 Introduction

When the task of inspecting some environment is given to a robot, it is effective that the robot freely changes view points while freely moves around. Based on this philosophy, we have mounted a high-performance active camera head on a mobile robot aiming at autonomous inspection of nuclear power plants like shown in Fig.1a. Here, it is quite important to accurately know the position and pose of the camera both to navigate the robot among the pipes and to compare the observed images with the normal states modeled in the computer. If we know the initial state of the robot, the current camera state in world coordinates can be calculated from the set values of its actuator modules. However, almost always, the values include some errors owing to many factors, such as tire slips, backlash of gears and so on. Therefore, we need to correct the values.



(a) (b)

Figure 1: Experimental environment: (a) plant-mockup; (b) its partial models consisting of 17 cylinders.

When a 3D model of the environment surrounding a camera is given, it is possible to know the position and pose of the camera by aligning the 3D model with the observed image. This strategy has a merit that the camera coordinates is directly calibrated with the environment which is inspection target. This subject is equal to the determination of the position and pose of a 3D rigid object from its 2D view, which is a fundamental and important problem in computer vision research. One typical approach for this purpose is feature-based one: it first extracts features(eg. edges, corners) and matches them between the 3D model and its 2D view. Once the 3D-2D point correspondences are obtained, the position and pose of the model can be quickly calculated (ex. [1]). Usually, however, neither robust feature extraction nor robust feature

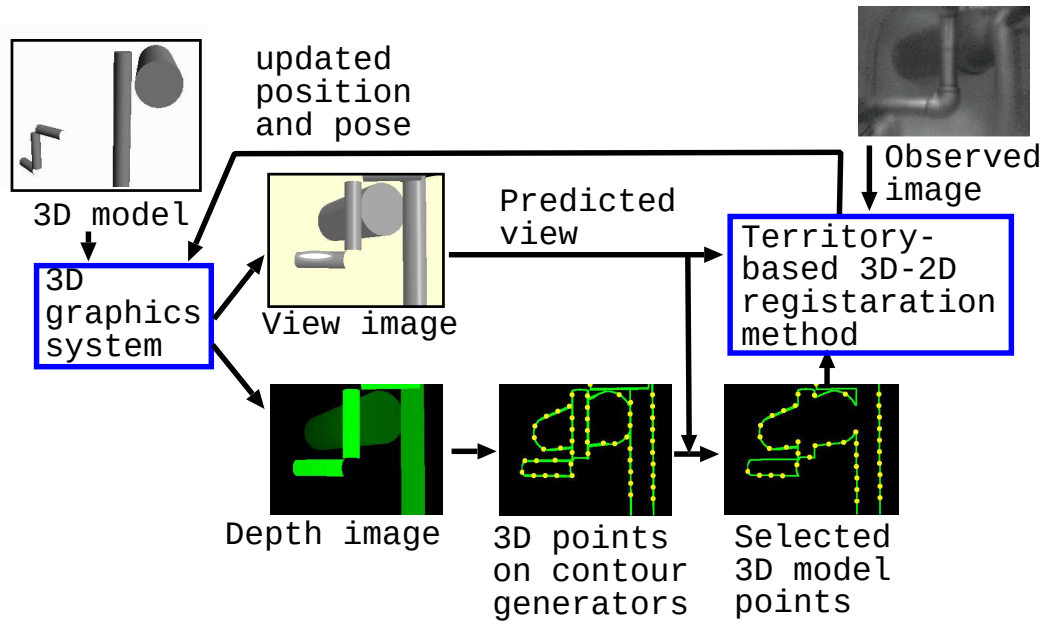


Figure 2: Scheme of determination of the position and pose of a camera using occluding edges

matching are easy. Observed images in our subject are also one of difficult images to extract and match the features, because complex combination of simple pipes produces complicated edges without prominent feature points. Additionally, the specular reflections on the surface of pipes increases the complexity of the features.

As described above, since the position and pose of the camera-head is autonomously controlled, its approximate values is known in advance. In such a case, it is an effective strategy to iteratively transform the 3D model towards the correct position and pose using the corresponding pairs between the observed and the model points, which are matched on the basis of the closeness at each state[2]. In [3], a 3D-2D registration method with similar strategy realized quick alignment of a 3D model of brain vessels with the observed X-ray images. The method can be extended so as to determine the relative position and pose of a camera by aligning a 3D model of its surrounding environment with the observed image. As described in the paper, to obtain high ratio of correct 3D-2D point pairs is indispensable for the strategy to work well. Although it is a way for this purpose to adopt some statistical method like M-estimation[4], even in the case, it is desirable that the ratio of correct pairs are high from the beginning.

In this paper, we propose to effectively use two-type predicted images which are calculated from the 3D environmental model by a graphics system (eg. OpenGL etc) for obtaining robust 3D-2D point correspondences. The predicted depth image is used to quickly calculate the contour generator that is the 3D line on the model's surface corresponding to the occluding contour in the observed image. The predicted intensity image is used to select only model points which are expected to appear clearly in the observed image. In Section 2, the whole scheme is described with explanations on each elemental process. The experiments using actual plant mock-up are shown with discussion on the accuracy of the results in Section 3.

2 Basic scheme of 3D-2D alignment

2.1 Whole procedures

Fig. 2 shows a scheme of our strategy for determining the position and pose of a camera by aligning the 3D model with occluding edges in an observed image. Suppose that an image is observed by a camera whose initial position and pose are estimated (eg. data from dead reckoning). Because of the estimation error, the projection of the environmental model on the observed image is deviated as shown in Fig. 4d. The concrete

procedures to correct the deviation are as follows:

i) Calculation of 3D model points
The 3D model points corresponding to the observed edges are calculated from the 3D environmental model according to the initial estimated state of the camera.

ii) 3D-2D point matching
Observed edge points corresponding to the 3D model points are determined based on the closeness on the observed image.

iii) Calculation of 3D transformation
The current position and pose of the camera is renewed to satisfy the 3D-2D point correspondences.

By iterating the processes from i) to iii), the predicted view is converging to the observed image and the correct position and pose of the camera are obtained. The details of each process are explained in the following subsections.

2.2 Calculation of 3D model points

This process is done quickly by reading the 3D coordinates of the edge points of the depth image calculated by a graphics system. White points in Fig. 4e show 3D model points calculated in such a way. The intensity image predicted by the graphics system is also used as follows:

- Only model points which are expected to robustly extract are selected.
  When the strength of edge is predicted to be weak, the corresponding 3D model points are removed away.

- Expected grey levels around the projection of the 3D model points are used as their attributes.
  In the experiments of this paper, the maximum gradient direction around the projection of the model points are used.

2.3 3D-2D point matching

Basically, the observed and the model points are matched on the basis of the closeness of the 2D distances between the observed edge point and the projected points of the 3D model points on the image. In addition to the basis, the following devices are taken to improve the ratio of correct pairs.

• Territory-based matching
The territory-based 3D-2D matching uses anisotropic search regions determined automatically from the projected shape of the model to prevent the bad influence

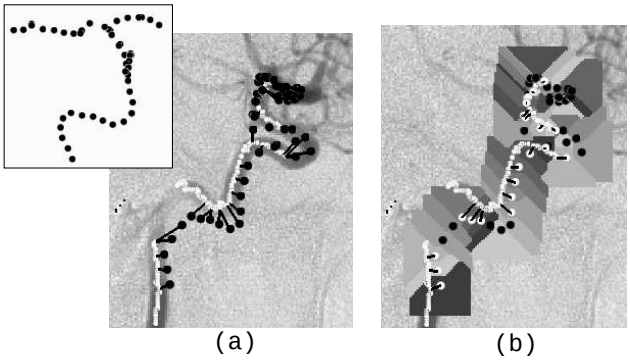


Figure 3: Territory-based 3D-2D matching: (a) correspondences based on only closeness in the 2D distance; (b) correspondences using territory-based search regions.

caused by lack of observed features, You can see the effect in the example of the registration of a 3D model of a blood vessel with its X-ray image shown in Fig. 3. The black points in Fig. 3a are the projection of the 3D model, shown at the upper left of the image. Since the given position and pose of the 3D model is a little different from the actual ones, the points are deviated from the observed vessel appearing as grey shadows. The white lines in Fig. 3a show the features, the skeleton of the vessels, extracted from the observed image. Because of lack of observed features and complex self-overlapping, the 3D-2D pairs defined based on the closeness in the 2D image plane include many undesirable pairs as shown with the black lines. On the other hand, if we use territory-based search restriction as shown in Fig. 3b, undesirable pairs are automatically excluded.

• Consistency of directional attributes
The maximum gradient directions around the observed edges are classified into eight directions. The corresponding attributes of the model points are determined from the maximum gradient directions around the projection of the 3D model points in the predicted intensity image. Only the points having the same directional attributes can be paired.

2.4 Calculation of 3D transformation

Once 3D-2D pairs are obtained, the 3D transformation of the model to satisfy the relations is quickly calculated using the following equations.

The observed image coordinates corresponding to n model points, $\mathbf{u} = (u_1, v_1, u_2, v_2, ...u_n, v_n)^T$ can be expressed as a function of the parameters determining

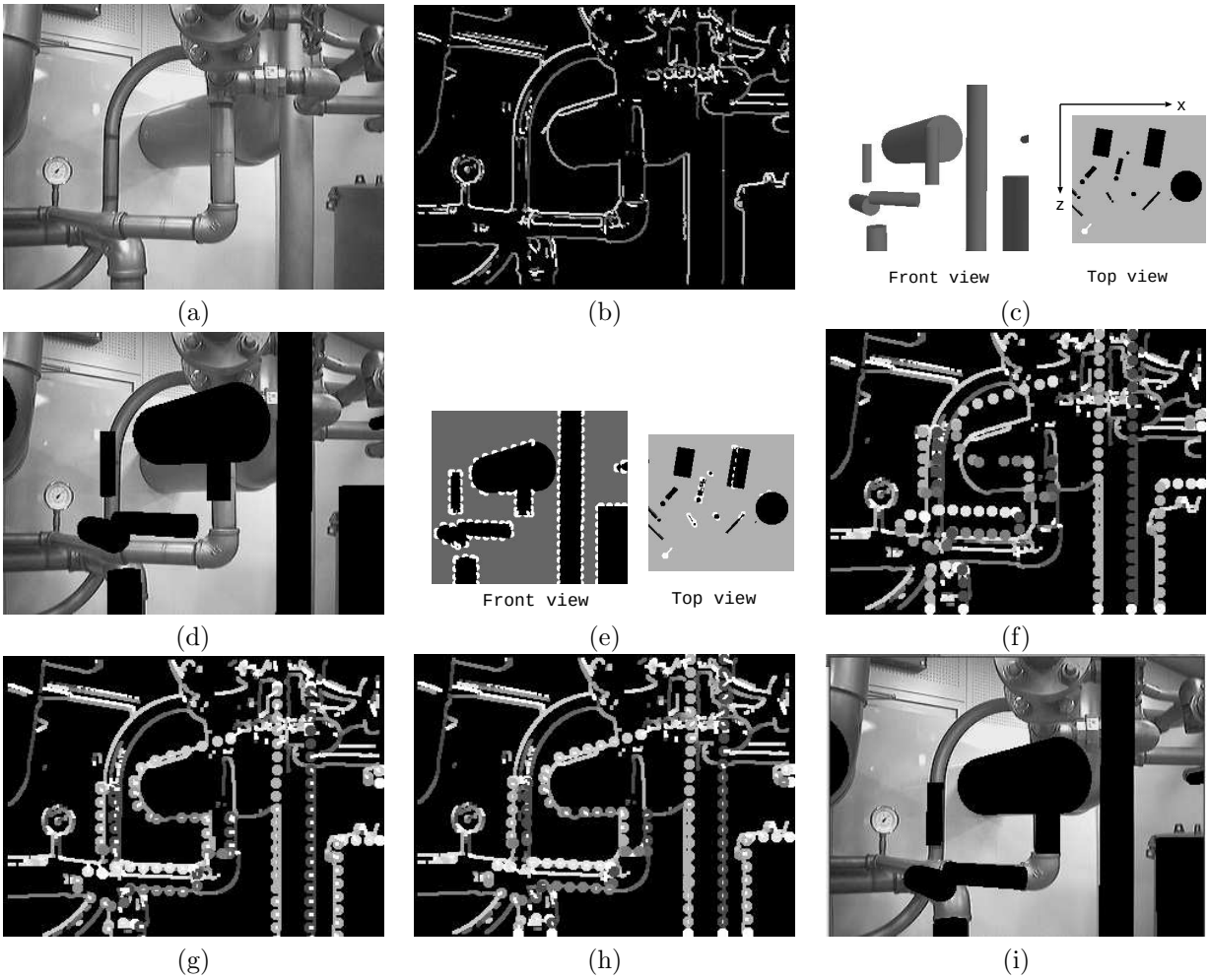


Figure 4: Experiment 1: (a) observed image; (b) edges in the observed image (classified by different grey levels); (c) front and top views of the 3D model; (d) projection of the model at the initial estimated state; (e) front and top views of the 3D model points (white points); (f) projection of the 3D model points on the observed edge image; (g) projection of the 3D model points after initial 2D translation; (h) projection of the 3D model points after convergence; (i) projection of the 3D model after convergence.

transformation of the model, $\mathbf{q}$, when all other viewing parameters (including camera parameters) are known:

$$\mathbf{u} = \mathbf{F}(\mathbf{q}). \tag{1}$$

Here, in this case, $\mathbf{q} = (t_x, t_y, t_z, r_x, r_y, r_z)^T$, where t_x, t_y, t_z and r_x, r_y, r_z represent translation along the (x, y, z) axis and rotation around the (x, y, z) axis respectively. We assume that $\mathbf{u_o}$ is observed, $\mathbf{q_c}$ is an initial (close) estimation $\mathbf{q}$ of its state, and $\mathbf{u_c}$ contains the projected coordinates at $\mathbf{q_c}$. By expanding Equation 1 in a Taylor series around $\mathbf{u_c}$ and taking terms up to first order, we obtain:

$$\mathbf{u_o} = \mathbf{u_c} + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{q}}\right|_{\mathbf{u_c}} \Delta\mathbf{q} \ . \tag{2}$$

Here, in the current subject,

$$\frac{\partial \mathbf{F}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial u_1}{\partial t_x} & \frac{\partial u_1}{\partial t_y} & \frac{\partial u_1}{\partial t_z} & \frac{\partial u_1}{\partial r_x} & \frac{\partial u_1}{\partial r_y} & \frac{\partial u_1}{\partial r_z} \\ \frac{\partial v_1}{\partial t_x} & \frac{\partial v_1}{\partial t_y} & \frac{\partial v_1}{\partial t_z} & \frac{\partial v_1}{\partial r_x} & \frac{\partial v_1}{\partial r_y} & \frac{\partial v_1}{\partial r_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\partial u_n}{\partial t_x} & \frac{\partial u_n}{\partial t_y} & \frac{\partial u_n}{\partial t_z} & \frac{\partial u_n}{\partial r_x} & \frac{\partial u_n}{\partial r_y} & \frac{\partial u_n}{\partial r_z} \\ \frac{\partial v_n}{\partial t_x} & \frac{\partial v_n}{\partial t_y} & \frac{\partial v_n}{\partial t_z} & \frac{\partial v_n}{\partial r_x} & \frac{\partial v_n}{\partial r_y} & \frac{\partial v_n}{\partial r_z} \end{bmatrix} . \tag{3}$$

Then, the $\mathbf{q}$ consistent with $\mathbf{u_o}$ is obtained by adding the following $\Delta\mathbf{q}$ to $\mathbf{q_c}$:

$$\Delta\mathbf{q} = \left[\left.\frac{\partial \mathbf{F}}{\partial \mathbf{q}}\right|_{\mathbf{u_c}}\right]^{\dagger} (\mathbf{u_o} - \mathbf{u_c}) \ . \tag{4}$$

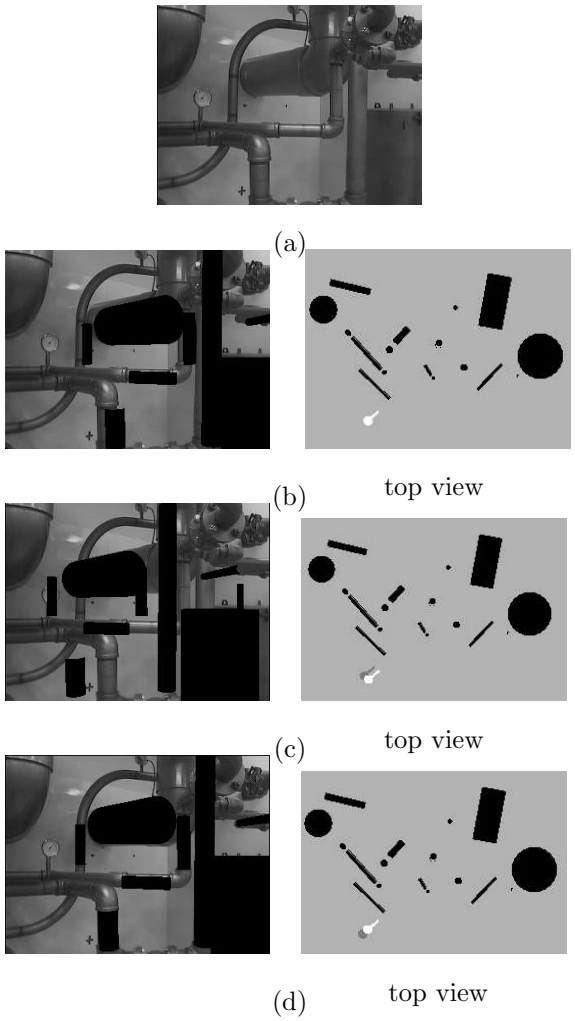


Figure 5: Example of localization result: (a) observed image; (b) measured state; (c) initial state; (d) result

Here, $[\mathbf{A}]^{\dagger}$ represents the pseudo-inverse of matrix $\mathbf{A}$. Because of the first-order approximation leading to Equation 2 and inaccurate matching pairs, the obtained solution may include some errors, and an iterative calculation using the obtained $\mathbf{q}$ as a new $\mathbf{q_c}$ for each step leads to convergence to the correct state.

3 Experiments

3.1 Alignment using actual images

Fig.1a shows our experimental environment, a plant-mockup. A robot with an active camera head moves around in the environment. 17 pipes in this environment are selected and modeled with cylinders in OpenGL as shown in Fig.1b. The 3D world coordinate system is defined as shown in Fig. 4c so that the x and z axes lie in the horizontal floor face; the y axis completes the left-handed coordinate system, and is in the vertical direction.

Fig. 4a is an observed image. Fig. 4b is the edge image, obtained by thresholding the differentiated data calculated with Canny operator[5]. The maximum gradient directions around the edges are classified into eight directions and represented with different grey levels. Fig. 4c shows a front and a top views of the partial model represented with cylinders in OpenGL. In the top view, the white circle and the white line sticking out from the point illustrate the position and the view direction of the camera. In Fig. 4d, the predicted view of the pipes at the position and pose given by the data from dead reckoning. Pipes are displayed with black color to be easy to see. The measuring error causes the deviation of the predicted view from the actual pipes. Fig 4e shows a front and a top views of the 3D model points (white points) obtained from the edges of the predicted depth image. Since the occluding edges which are expected to appear as weak edges are excluded based on the predicted intensity image, the occluding edges with the background of similar color pipes are not selected. As the same manner as the edges of observed image, the maximum gradient directions around the projection of the 3D model points in the predicted intensity image are classified into eight directions and become the attribute of each model point. In Fig. 4f, the model points are overlaid on the observed edge image with different grey levels describing the attributes.

Since a little change in camera angle causes a big translation in the image, such translation should be taken into consideration when we use the closeness on the image as a clue to determine the 3D-2D point correspondences. Concretely, the projected 3D model points are two dimensionally translated on the image to search for the best position where the model points overlapped on the edges with the same direction attribute. Fig. 4g shows the position after this initial translation. The 3D-2D point correspondences obtained by the territory-based matching at this position are illustrated with white lines. The camera is relatively moved to the 3D transformation calculated by inputing the corresponding pairs to Equation 4. At the new state, the same processes except the 2D translation on the image are iterated. In this example, the camera was converged to the state which gives the projection of the model as shown in Fig. 4h, i after 10 iterations. The camera movement from the initial

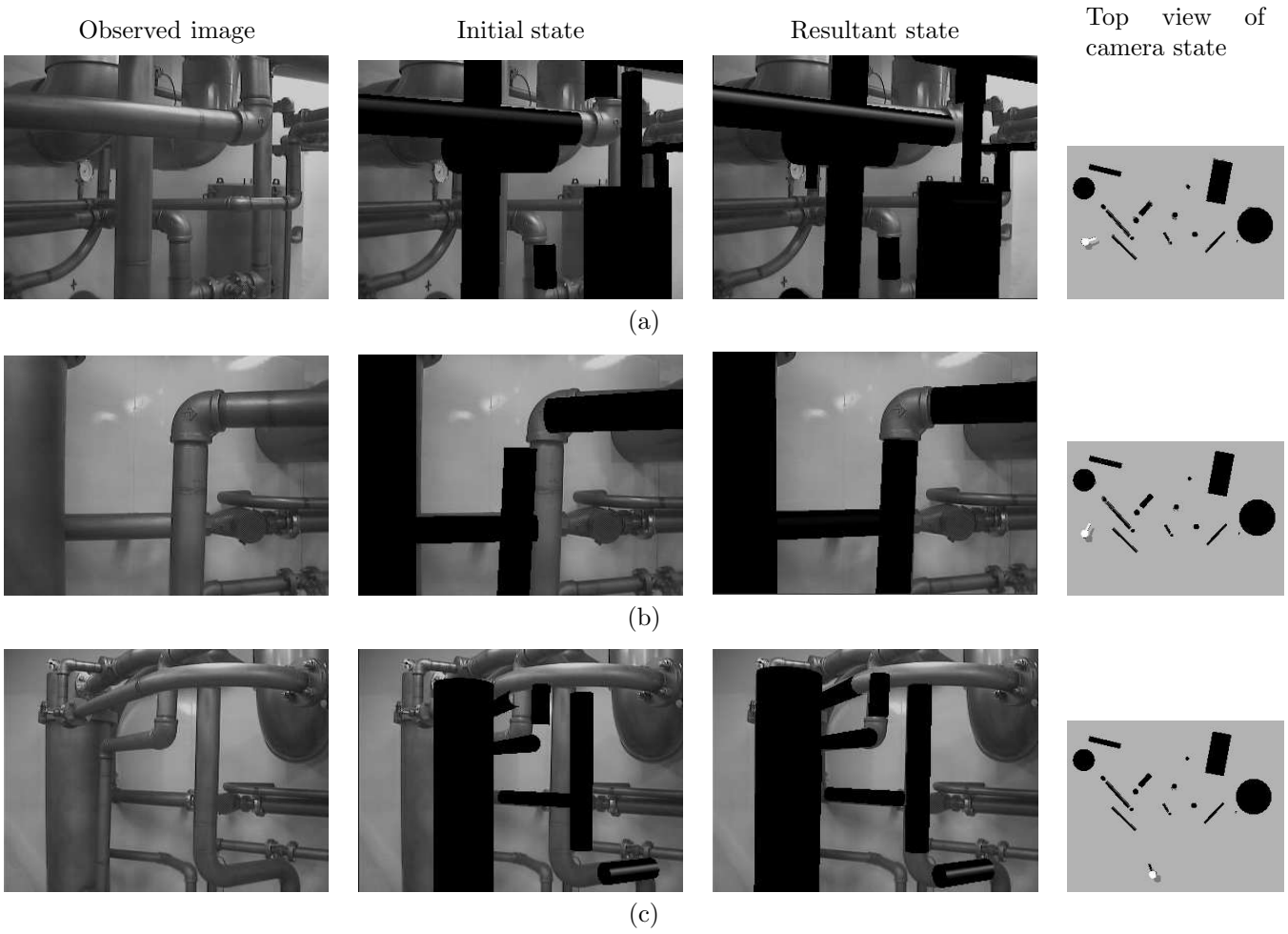


Figure 6: Experiment 3: Corrections of the position and pose of the camera while moving the camera in the environment.

state was $(35.3, 0.69, -60.5)$(mm) translation and 3.7 degree rotation around the axis $(-0.65, 0.068, -0.76)$. Computational time was 2.1 sec for initial translation and 0.2 sec for a loop of calculation of the 3D model points, 3D-2D matching and renewal of the position and pose of the camera, and hence the total time was 4.1 sec (Pentium II(333MHz)).

3.2 Accuracy in 3D localization

We examined accuracy of the 3D localization calculated from the proposed 3D-2D alignment method by comparing with manual measurements.

Fig. 5a shows an example of the images observed by a camera mounted on the robot. The position and pose of the camera head was manually measured with great care and illustrated in the top view of Fig. 5b. The accuracy of the manual measurement is about $\pm$5mm in translation and $\pm$3 degrees in rotation. Because of this slight error, the projection of the model at the state shows a little deviation from the observed image as shown in Fig. 5b.

We intentionally add some errors to the camera state and use it as the initial estimate. Fig. 5c shows the projection of the 3D model when giving the camera state after adding (50,0,50) mm translation and 5 degree rotation around the y axis to the measured state. In the top view, the white circle shows the current camera position, while the gray circle overlapped by the white circle shows the measured state. Fig. 5d shows the result after correcting the camera state by the method described in Section 2. The model is well aligned with the image. Nevertheless, as shown in the top view, the translation error occurred mainly in the view direction, which is about 90 mm.

We have done similar experiments using more than 10 images observed at various locations. In all the experiments, 3D models are well aligned with observed

images and 3D localization is converged. This showed the robustness of the method in such a complex scene. However, the translation error in the view direction appears in all cases.

3.3 Analysis of error factors

The factors causing the 3D localization errors can be counted up as follows:

1. Pixel quantization
2. Inexact camera internal parameters
3. Inaccuracy of the 3D models
4. Wrong 3D-2D correspondences

The accuracy of our 3D model is based on the manual measurement and about ±5mm in translation and ±3 degrees in rotation. Since we calculate the camera state from the 3D-2D corresponding pairs based on least-squares estimation at the present, the wrong correspondences left at the final state deteriorate the accuracy. However, these errors should produce random errors in the 3D localization. From the observation of the clear tendency for the error to be translation in the view direction, we focus on the two suspicious factors, pixel quantization and the focal length of the camera internal parameters.

To analyze the effect of these errors, we conducted experiments using synthetic images [6]. As a result, it was found the error in the 3D location caused by the quantized error is small: the translation and rotation errors are about 0.6 mm and 0.03 degrees in a similar situation to Fig.5. This accuracy is supported by the fact that the method uses lots of 3D-2D corresponding pairs which distributed in a whole image (in this case, about 120 pairs).

On the other hand, the 3D location error caused by the inaccurate focal length was proved so large that cannot be disregarded. When giving a shorter focal length for synthetic views, the location deviated further in the view direction. In the situation in Fig. 5, the magnitude of the translation error in the view direction is about −36mm per 0.1 mm error in the focal length.

After this observation, we carefully measured the angle of the field of view of the actual camera to calculate the focal length. We found it is actually 48 degrees, although we had used the focal length corresponding to 50 degree angle of the field of view. In the case of data in Fig. 5, translation error in the view direction is decreased from 90 mm to 4mm by correcting this camera parameter.

Fig. 6 shows the some other results after correcting the focal length. In the top views, the white circle shows the resultant camera position, while the gray circle overlapped by the white circle shows the initial state given by adding some errors to each measured state. Although the measured states are also displayed with the black circle in the top views, they are hardly seen because the white circles almost perfectly overlapped. In most of cases, the differences between the calculated final state and the measured state are less than ± 30mm translation and ± 3.0 degree rotation.

In the paper [6], we proposed usage of two cameras for compensating the error cased by the inaccurate focal length, for the case that it is difficult to know the accurate values especially when a robot need to change the camera focus and/or zoom during a sequential task.

4 Summary and Conclusions

We proposed a method to determine the position and pose of the camera using occluding edges with the condition that the 3D environmental model and an approximate initial state are given. The characteristics of the method are as follows:

1) Quickness.

- Only 3D model points corresponding to the observed features are calculated rapidly from the predicted depth image.
- The structures and connectivity of corresponding features do not need to be extracted.

2) Robustness.

- Model points which are expected to vaguely appear on the observed image are removed away.
- Territory-based search restriction and usage of the edge directional attribute raise the ratio of the correct pairs of 3D-2D point correspondences.

The results of the preliminary experiments show that the proposed method robustly aligns the 3D model of an environment with observed images despite of noisy complex scene with specular reflections. As far as we know, it is the first 3D-2D alignment method which can use so complicated occluding edges as the clue to determine the position and pose of an object. Allowable initial estimation error is about 10 cm translation and 10 degree rotation, which is enough bigger than actual estimation errors from dead reckoning.

We also investigated accuracy of the 3D localization obtained by the proposed method. From the ex-

perimental results, the method seems to offer 3D localization accuracy within at most ±30mm in translation and ±3 degrees in rotation. These values are enough for the purpose of the robot navigation in narrow spaces of the plant. Aiming at applying to the tasks which requires more accuracy[7], we will investigate more the influences of the error factors noted in Section 3.3.

Our future work will focus on detection of abnormality in the environment. Once an observed image is accurately aligned with the 3D model, simple subtraction of the predicted view from the observed image can tell difference from the normal state, such as a surface defect of pipes and so on. We plan to conduct experiments on this matter aiming at autonomous inspection.

Acknowledgments

We are thankful to Dr. Kazuo Tanie, and Dr. Shigeoki Hirai and the members of Robots group and Computer Vision group in AIST. We are especially grateful to Dr. Takashi Suehiro for his advice on the calculation of 3D transformation from 3D-2D point correspondences.

References

[1] M. Sugai, Y. Hori, T. Ogasawara and H. Tsukune: "Active vision system based on the structual tracking by using robot manipulator", *JRSJ(Journal of Robotics Society of Japan*, Vol. 13, No. 3, pp. 411–419, 1995 [In Japanese].

[2] P. J. Besl and N. D.Mckay: "A method for registration of 3D shapes", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No.2, pp. 239–256, 1992.

[3] Y. Kita, D. L. Wilson and J. A. Noble: "Real-time registration of 3D cerebral vessels to X-ray angiograms", In *Proc. of 1st International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 1125–1133, 1998.

[4] M. D. Wheeler: *"Automaic modeling and Localization for object recognition"*, PhD thesis, Carnergie Mellon University, 1996.

[5] J. Canny: "A Computational Approach to Edge Detection", *IEEE Trans.Pattern Anal. & Mach. Intell.*, Vol. 8, No. 6, pp. 679–698, 1986.

[6] Y. Kita and N. Kita: "On accuracy of 3D localization obtained by aligning 3D model with observed 2D occluding edges", In *Proc. of 5th Asian Conference on Computer Vision*, (To appear), 2002.

[7] N. Kita, Y.Kita and H. Yang: "Environment Server: Digital Field Information Archival Technology", In *Proceedings of AIR'02*, (To appear), 2002.