Automatic Calibration and Trajectory Reconstruction of Mobile Robot in Camera Sensor Network

Yonghoon Ji, Atsushi Yamashita, and Hajime Asama

***Abstract*—To operate mobile robots in an intelligent space such as a distributed camera sensor network, pre-calibration of all environmental cameras (i.e., determining the absolute poses of each camera) is an essential task that is extremely tedious. The optimization problem for camera calibration with a mobile robot has been intensively studied in the past. However, most existing solutions have limitations in that they can estimate only three degree of freedom (DOF) parameters (*x*, *y*, *yaw*) with restrictive assumptions. In this paper, we propose a novel method that achieves trajectory reconstruction of a mobile robot and calibration of complete 6DOF (*x*, *y*, *z*, *roll*, *pitch*, *yaw*) external parameters of distributed cameras by utilizing easily obtainable grid map information of the environment as prior information. In addition, a novel two-way observation model is proposed. The map information and two-way observation model help seek a global minimum solution (i.e., 6DOF camera parameters and robot trajectory) within the objective function containing many local minimums. We evaluate the proposed method in a simulation environment with a virtual camera network of up to 10 cameras and a real environment with a mobile robot in a wireless camera network. The results demonstrate that the proposed framework can estimate the 6DOF camera parameters and the target trajectory successfully.**

I. Introduction

In recent years, many studies about intelligent space, which includes distributed sensors in a human-robot coexistence environment, have been performed [1], [2]. A camera sensor network with a multi-camera system is the most general example that constructs an intelligent space. By constructing the camera sensor network in such an environment, it is able to recognize various events that occur in the environment so that mobile agents in the space can provide appropriate services to humans, as shown in Fig. 1. To operate mobile agents in a distributed camera sensor network, pre-calibration of all environmental cameras (i.e., determining the absolute poses of each camera) is an essential task that is extremely tedious. In this respect, a number of studies that provide Bayesian filter-based probabilistic estimates of sensor parameters and target tracks have been conducted. Foxlin proposed the simultaneous localization and auto-calibration (SLAC) concept, which is a very general architectural framework for navigation and tracking systems with environment sensors [3]. Taylor et al. also implemented a simultaneous localization, calibration, and tracking (SLAT) system using radio and ultrasound pulse-based range sensors as environment sensors [4]. However, these methods can only be applied with range and bearing sensors and cannot make use of information from the camera sensor network, which is a popular network system for the intelligent space.

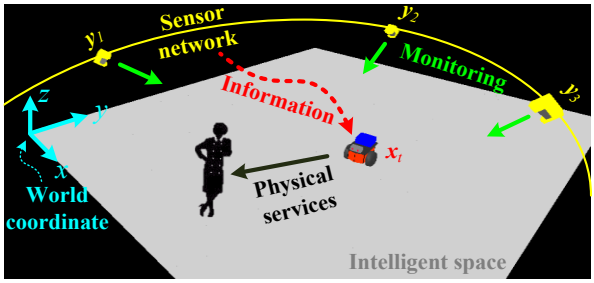


Figure 1. Concept of intelligent space.

Chen et al. employed an approach that optimizes robot motion to minimize camera calibration error; however, it needs an assumption that the robot motion has no uncertainty and rough parameters of the camera should be initialized by human observation [5]. Proposals by Rahimi et al. and Funiak et al. recovered the most likely camera poses and the target trajectory given in a sequence of observations from the camera network without taking into account special prior conditions [6], [7]. These approaches are based on maximum a posteriori (MAP) estimation, which is similar to ours, but rather than estimating complete six degree of freedom (DOF) poses (*x*, *y*, *z*, *roll*, *pitch*, *yaw*) for each camera, they only estimate 3DOF (*x*, *y*, *yaw*) external parameters with a restrictive assumption that requires aligning each camera's ground-plane coordinate system with a global ground-plane coordinate system. The optimization problem of including orientation parameters for all axes (*roll*, *pitch*, *yaw*) may have a myriad number of local minimum solutions without additional constraints because many indistinguishable observations may exist even if the poses of cameras are different.

Here, easily obtainable priors (e.g., the map information of the environment) can greatly improve solutions to the problem. Therefore, to realize a 6DOF parameter estimation, we propose a novel approach that uses the grid map information of the environment as an additional constraint for the objective function within the optimization process. The grid map information that contains information of the interior wall can be very useful for the additional constraint since the

This work was in part supported by Tough Robotics Challenge, ImPACT Program (Impulsing Paradigm Change through Disruptive Technologies Program).

Yonghoon Ji (corresponding author) is with the Dept. of Precision Eng., The University of Tokyo, Tokyo, Japan (phone: +81-5841-6486; e-mail: ji@robot.t.u-tokyo.ac.jp).

Atsushi Yamashita is with the Dept. of Precision Eng., The University of Tokyo, Tokyo, Japan (e-mail: yamashita@robot.t.u-tokyo.ac.jp).

Hajime Asama is with the Dept. of Precision Eng., The University of Tokyo, Tokyo, Japan (e-mail: asama@robot.t.u-tokyo.ac.jp).

cameras are generally installed on the interior wall because of space limitations. Such grid map information can be easily obtained from the blueprint of the artificial environment (e.g., CAD data) or a traditional simultaneous localization and mapping (SLAM) scheme. In our research, therefore, the additional constraint for the objective function is obtained by transforming the grid map information. Additionally, the use of map information allows our system to estimate the rough position of the mobile robot that can also be added to the prior information even if it has some uncertainty.

Compared with the traditional observation model, which models measurement information from the distributed camera to the target (i.e., one-way observation), we define a novel two-way observation model based on an assumption that the camera and the target can observe each other in order to strengthen the constraints of the objective function.

The contribution of this research is as follows. Previous studies in which only 3DOF poses could be estimated have a significant limitation on the camera network installation. On the other hand, our complete 6DOF calibration system is able to construct a camera network system in arbitrary poses on the wall plane and easily calibrate its parameters simply by controlling the mobile robot.

The remainder of this paper is organized as follows: Section II defines the state variables to be estimated in the camera network system with a mobile robot. Likelihood functions and prior distributions to define the objective function are presented in Section III. Section IV describes the mathematical derivation of the objective function based on MAP estimation and the method to find a minimum solution. The effectiveness of the proposed method is evaluated with experimental results in Section V. Finally, Section VI gives our conclusions.

II. State Definition in Camera Sensor Network

A. State Variables

In this paper, to estimate robot trajectory and the state of the camera network system (i.e., the external parameters for all distributed cameras), the state vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ are defined as follows:

$$\boldsymbol{X} = \left[(\boldsymbol{x}_1)^{\mathrm{T}} \quad (\boldsymbol{x}_2)^{\mathrm{T}} \quad \cdots\right]^{\mathrm{T}} \tag{1}$$

$$\boldsymbol{x}_t = \left[x_t \quad y_t \quad \varphi_t\right]^{\mathrm{T}} \tag{2}$$

$$\boldsymbol{Y} = \left[(\boldsymbol{y}^1)^{\mathrm{T}} \quad (\boldsymbol{y}^2)^{\mathrm{T}} \quad \cdots \quad (\boldsymbol{y}^n)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{3}$$

$$\boldsymbol{y}^k = \left[x_c^k \quad y_c^k \quad z_c^k \quad \psi_c^k \quad \theta_c^k \quad \varphi_c^k\right]^{\mathrm{T}} \tag{4}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{y}^k$ are the 3DOF robot pose at time t and the k^{th} distributed camera pose (i.e., 6DOF external parameters), respectively. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the complete states of the robot trajectory and the camera network system that should be estimated. n is the number of distributed cameras composing the camera network system for the intelligent space. In short, the optimization problem in this paper is estimating a high dimensional state vector containing the whole robot trajectory $\boldsymbol{X}$ and the camera poses $\boldsymbol{Y}$.

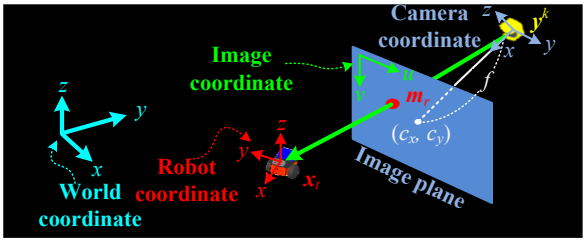


Figure 2. Relationship between world coordinate and image coordinate according to camera calibration.

B. Camera Calibration Model

The camera calibration model that defines the relationship between the image coordinate and world coordinate consists of an intrinsic parameter matrix $\boldsymbol{A}$ and extrinsic parameter matrix $\boldsymbol{T}$, which are given by

$$\boldsymbol{m} = \boldsymbol{A}\boldsymbol{T}\boldsymbol{x} \tag{5}$$

$$\boldsymbol{A} = \begin{bmatrix} f & f_{\mathrm{skew}} & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where the internal parameters (c_x, c_y), f, and f_{skew} denote principal point, focal length, and skew coefficient, respectively. If the state vector of the camera network system (4) is estimated, the extrinsic parameter matrix $\boldsymbol{T}$ made up of the rotation matrix and translation vector can be easily transformed by using the Eular angle (i.e., it expresses the 6DOF pose of the camera with respect to the world coordinate). Therefore, if the intrinsic parameter matrix $\boldsymbol{A}$ is known, the calibration task can be completed and it is possible to convert the world coordinates $\boldsymbol{x} = [x\ y\ z\ 1]^{\mathrm{T}}$ to image coordinates $\boldsymbol{m} = [u\ v\ 1]^{\mathrm{T}}$ using (5). For example, the robot state $\boldsymbol{x}_t$ observed by distributed camera $\boldsymbol{y}^k$ is mapped to the pixel coordinate $\boldsymbol{m}_t = [u_t\ v_t\ 1]^{\mathrm{T}}$ in the image, as shown in Fig. 2.

III. Likelihoods and Priors for Objective Function

A. Observation Models for Likelihood

The observation model for both the camera network system and the camera mounted on the mobile robot (i.e., as in the two-way observation model shown in Fig. 3) provides the likelihood function that imposes stronger constraints on the robot pose and the camera parameters. The observation model for the camera network $\boldsymbol{H}_c(\cdot)$ based on the predicted robot state vector $\boldsymbol{x}_t$ and the state vector of the k^{th} distributed camera $\boldsymbol{y}^k$ is defined as follows by using the camera calibration model (5):

$$\hat{\boldsymbol{z}}_c^k = \boldsymbol{H}_c(\boldsymbol{x}_t, \boldsymbol{y}^k) = \boldsymbol{A}_c \boldsymbol{T}^k \breve{\boldsymbol{x}}_t \tag{7}$$

$$\begin{aligned} p(\boldsymbol{z}_c^k \mid \boldsymbol{x}_t, \boldsymbol{y}^k) &= N(\boldsymbol{z}_c^k \mid \hat{\boldsymbol{z}}_c^k, \sigma_z^2 \boldsymbol{I}) \\ &= \det\left(2\pi\sigma_z^2 \boldsymbol{I}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{z}_c^k - \hat{\boldsymbol{z}}_c^k)^{\mathrm{T}} \boldsymbol{\Omega}(\boldsymbol{z}_c^k - \hat{\boldsymbol{z}}_c^k)\right) \end{aligned} \tag{8}$$

$$p(\boldsymbol{Z}_c \mid \boldsymbol{X}, \boldsymbol{Y}) = \prod_{<k,t> \in \boldsymbol{O}_c} p(\boldsymbol{z}_c^k \mid \boldsymbol{x}_t, \boldsymbol{y}^k) \tag{9}$$

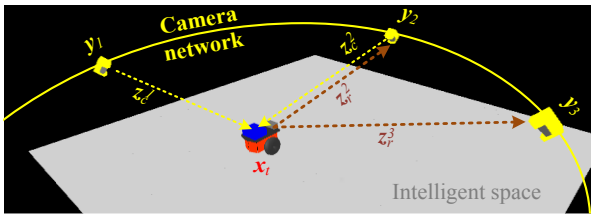


Figure 3. Concept of two-way observation model.

where $\hat{z}_c^k$ is the predicted observation in the pixel coordinate (u, v) of the k^{th} distributed camera at time t from the predicted state vector $\boldsymbol{x}_t$ and $\boldsymbol{y}^k$. $\boldsymbol{A}_c$ and $\boldsymbol{T}^k$ denote the intrinsic parameter matrix and the extrinsic parameter matrix of the k^{th} distributed camera, respectively. $\breve{\boldsymbol{x}}_t = [x_t \quad y_t \quad 0 \quad 1]^{\mathrm{T}}$ is the position vector of the robot state with the exception of the orientation elements. Equation (8) describes a likelihood function for this observation model with covariance $\sigma_z^2\boldsymbol{I}=\boldsymbol{\Omega}^1$ that means uncertainty of the pixel observation. Since each observation can be considered independent of one another, the likelihood function that includes a collection of all observations is represented as (9). Let $\boldsymbol{O}_c$ be the set of pairs of indices $<k, t>$, where $<k, t>\in \boldsymbol{O}_c$ iff the k^{th} distributed camera observes the mobile robot at time t.

The observation model $\boldsymbol{H}_r(\cdot)$ for the camera mounted on the mobile robot can also be defined in the same manner as the camera network's observation model as follows:

$$\hat{z}_r^k = \boldsymbol{H}_r(\boldsymbol{x}_t, \boldsymbol{y}^k) = \boldsymbol{A}_r\boldsymbol{T}_r\breve{\boldsymbol{y}}^k \tag{10}$$

$$\begin{aligned} p(z_r^k \mid \boldsymbol{x}_t, \boldsymbol{y}^k) &= N(z_r^k \mid \hat{z}_r^k, \sigma_z^2\boldsymbol{I}) \\ &= \det\left(2\pi\sigma_z^2\boldsymbol{I}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_c^k - \hat{z}_c^k)^{\mathrm{T}}\boldsymbol{\Omega}(z_c^k - \hat{z}_c^k)\right) \end{aligned} \tag{11}$$

$$p(\boldsymbol{Z}_r \mid \boldsymbol{X}, \boldsymbol{Y}) = \prod_{<t,k>\in\boldsymbol{O}_r} p(z_r^k \mid \boldsymbol{x}_t, \boldsymbol{y}^k) \tag{12}$$

where $\hat{z}_r^k$ is the predicted observation in the pixel coordinate (u, v) of the camera mounted on the mobile robot at time t from the predicted robot state vector $\boldsymbol{x}_t$ and $\boldsymbol{y}^k$. $\boldsymbol{A}_r$ and $\boldsymbol{T}_r$ denote intrinsic and extrinsic parameter matrices of the camera mounted on the mobile robot, respectively. $\breve{\boldsymbol{y}}^k = [x_c^k \quad y_c^k \quad z_c^k \quad 1]^{\mathrm{T}}$ is the position vector of the k^{th} distributed camera excepting the orientation elements. The likelihood function is also represented in the same manner as (8) and (9). Here, the set of $\boldsymbol{O}_r$ denotes a set of $<t, k>$ pairs, where $<t, k>\in \boldsymbol{O}_r$ iff the mobile robot observes the k^{th} distributed camera at time t.

B. Localization for Prior Information

Most mobile robot systems essentially operate on a sequence of range measurements and the control input data. The robot pose is also estimated based on such sequential measurements. In this respect, roughly estimated robot pose data with its uncertainty (i.e., covariance) can be used as prior information for the global constraints even though the accuracy is quite low because of several noises. The Bayesian filtering-based localization method, which is the most widely used probabilistic scheme, is represented as:

$$\begin{aligned} p(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{l}_t) &= \eta \cdot p(\boldsymbol{l}_t \mid \boldsymbol{x}_t) \int p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{u}_t) p(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \\ &= N(\boldsymbol{x}_t \mid \hat{\boldsymbol{x}}_t, \boldsymbol{P}_t) \\ &= \det\left(2\pi\boldsymbol{P}_t\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t)^{\mathrm{T}}\boldsymbol{\Theta}_t(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t)\right) \end{aligned} \tag{13}$$

$$p(\boldsymbol{X} \mid \boldsymbol{U}, \boldsymbol{L}) = \prod_t p(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{l}_t) \tag{14}$$

where $\boldsymbol{u}_t$ and $\boldsymbol{l}_t$ mean the control input data that are obtained from encoder information and range measurements, respectively. Here, the resulting product is generally not a probability (i.e., it may not integrate to 1). Hence, the result is normalized by virtue of the normalization constant η. Over the past few decades, a considerable number of studies have been conducted on this type of localization method based on the Kalman filter or particle filter [8]. Because the localization results follow normal distribution (i.e., these provide not only pose data $\hat{\boldsymbol{x}}_t$ but also its covariance data $\boldsymbol{P}_t=\boldsymbol{\Theta}_{t\text{-}1}$ explicitly), we can easily apply it as the prior distribution.

C. Map Information for Prior Information

We can additionally take the prior information into consideration by incorporating the grid map information of the environment as the camera network is generally installed on the wall plane represented in such map information. To this end, we define a prior distribution using the map information as follows:

$$\begin{aligned} p(\boldsymbol{y}^k \mid \boldsymbol{M}) &= p(x_c^k, y_c^k \mid \boldsymbol{M}) p(z_c^k, \psi_c^k, \theta_c^k, \varphi_c^k \mid \boldsymbol{M}) \\ &= \lambda \cdot \boldsymbol{m}(x_c^k, y_c^k) \end{aligned} \tag{15}$$

$$p(\boldsymbol{Y} \mid \boldsymbol{M}) = \prod_k p(\boldsymbol{y}^k \mid \boldsymbol{M}) \tag{16}$$

$p(z_c^k, \psi_c^k, \theta_c^k, \varphi_c^k \mid \boldsymbol{M})$ in (15) can be considered as a uniform distribution since the map information $\boldsymbol{M}$ cannot provide any constraints about the elements except (x, y) position; thus, this term can be ignored. λ is the normalization constant. Figure 4 shows the procedure for generating the function $\boldsymbol{m}(\cdot)$ for the prior distribution from the grid map information. First, contour lines that represented wall information are generated from the original map using a thinning algorithm as shown in Fig. 4 (a) and (b). Then, to generate a probability-like distribution, Gaussian blur is applied to the contour lines as shown in Fig. 4 (c) and (d). These processes can be performed by image processing techniques. Intuitively, a high value of $\boldsymbol{m}(\cdot)$ means that the probability over the camera position is high. Therefore, (16) implies prior distribution that is a conditional probability of camera parameters $\boldsymbol{Y}$ given grid map $\boldsymbol{M}$ and it can be considered to generate probability distribution about the existence of the wall from the grid map. Despite the fact that only x and y elements of the camera poses cannot be constrained, this plays an important role when seeking the minimum solution of the objective function. Note that if we use a 3D map (e.g., voxel map [9] or OctoMap [10]) instead of the 2D grid map, prior distribution can be

expanded to include not only the wall but also the ceiling information of the indoor environment.

IV. Finding Minimum Solution of Objective Function

This section describes the optimization method to obtain the most likely robot trajectory and the camera poses. The posterior probability of trajectory $\boldsymbol{X}$ and camera parameters $\boldsymbol{Y}$ given the whole conditions mentioned in Section III is defined by

$$\begin{aligned} & p(\boldsymbol{X},\boldsymbol{Y} \mid \boldsymbol{U},\boldsymbol{L},\boldsymbol{Z}_c,\boldsymbol{Z}_c,\boldsymbol{M}) \\ & \propto p(\boldsymbol{Z}_c,\boldsymbol{Z}_r \mid \boldsymbol{X},\boldsymbol{Y},\boldsymbol{U},\boldsymbol{L},\boldsymbol{M}) p(\boldsymbol{X},\boldsymbol{Y} \mid \boldsymbol{U},\boldsymbol{L},\boldsymbol{M}) \\ & = p(\boldsymbol{Z}_c,\boldsymbol{Z}_r \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{X},\boldsymbol{Y} \mid \boldsymbol{U},\boldsymbol{L},\boldsymbol{M}) \\ & = p(\boldsymbol{Z}_c \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{Z}_r \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{X} \mid \boldsymbol{U},\boldsymbol{L},\boldsymbol{M}) p(\boldsymbol{Y} \mid \boldsymbol{U},\boldsymbol{L},\boldsymbol{M}) \\ & = p(\boldsymbol{Z}_c \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{Z}_r \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{X} \mid \boldsymbol{U},\boldsymbol{L}) p(\boldsymbol{Y} \mid \boldsymbol{M}) \end{aligned} \tag{17}$$

Here, some variables that imply independence are omitted. The goal of this MAP estimation is to find the configuration of state variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ that maximizes this posterior distribution. Therefore, the most a posteriori probable robot trajectory and camera parameters are given by

$$\begin{aligned} & (\boldsymbol{X}^*,\boldsymbol{Y}^*) \\ & = \arg\max_{\boldsymbol{X},\boldsymbol{Y}} \left[p(\boldsymbol{Z}_c \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{Z}_r \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{X} \mid \boldsymbol{U},\boldsymbol{L}) p(\boldsymbol{Y} \mid \boldsymbol{M}) \right] \end{aligned} \tag{18}$$

Here, $p(\boldsymbol{Z}_c \mid \boldsymbol{X}, \boldsymbol{Y})$ and $p(\boldsymbol{Z}_r \mid \boldsymbol{X}, \boldsymbol{Y})$ are the likelihood functions defined by (9) and (12), respectively. $p(\boldsymbol{X} \mid \boldsymbol{U}, \boldsymbol{L})$ and $p(\boldsymbol{Y} \mid \boldsymbol{M})$ are prior distributions on the trajectory and the calibration parameters, which are defined by (14) and (16), respectively. By applying negative log, the optimization equation of (18) becomes a non-linear least squares problem over the state variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ as follows:

$$\begin{aligned} & (\boldsymbol{X}^*,\boldsymbol{Y}^*) \\ & = \arg\min_{\boldsymbol{X},\boldsymbol{Y}} \left[-\log p(\boldsymbol{Z}_c \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{Z}_r \mid \boldsymbol{X},\boldsymbol{Y}) p(\boldsymbol{X} \mid \boldsymbol{U},\boldsymbol{L}) p(\boldsymbol{Y} \mid \boldsymbol{M}) \right] \\ & = \arg\min_{\boldsymbol{X},\boldsymbol{Y}} \begin{bmatrix} \text{const.} + \frac{1}{2} \sum_{\langle k,t \rangle \in Zc} (\boldsymbol{z}_c^k - \hat{\boldsymbol{z}}_c^k)^{\mathrm{T}} \boldsymbol{\Omega} (\boldsymbol{z}_c^k - \hat{\boldsymbol{z}}_c^k) \\ + \frac{1}{2} \sum_{<k,t> \in Zc} (\boldsymbol{z}_r^k - \hat{\boldsymbol{z}}_r^k)^{\mathrm{T}} \boldsymbol{\Omega} (\boldsymbol{z}_r^k - \hat{\boldsymbol{z}}_r^k) \\ + \frac{1}{2} \sum_t (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t)^{\mathrm{T}} \boldsymbol{\Theta}_t (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t) - \sum_k \log \boldsymbol{m}(x_c^k, y_c^k) \end{bmatrix} \end{aligned} \tag{19}$$

To find the optimal $\boldsymbol{X}^*$ and $\boldsymbol{Y}^*$ that minimize the objective function (19), we use the Levenberg-Marquardt method [11]. The Levenberg-Marquardt method finds the solution by using both the Gauss–Newton method and the gradient descent method; therefore, convergence speed is relatively fast and an optimal solution can be found reliably.

V. Experiments

A. Simulation

We simulated the proposed method in a large-scale simulation environment with a virtual camera network of up to 10 cameras. Figure 5 shows the grid map of the simulation environment with true trajectory and poses of virtual cameras. Here, the target trajectory was generated on an assumption that it has no uncertainty (i.e., we set $\boldsymbol{X}$ to the true trajectory with $\boldsymbol{\Theta}_t = diag(\infty, \infty, \infty)$ and optimized (19) for the camera parameters only) in the case of the simulation experiment. The Levenberg-Marquardt iterations began with an initial estimate for the all camera configurations at $\boldsymbol{y}^k_{\text{init}}$ = [15 m, 12 m, 4 m, 0 deg, 90 deg, 90 deg]$^{\mathrm{T}}$ (i.e., the position of the map center). Here, we set the internal parameters f, f_{skew}, and (c_x, c_y) for the virtual cameras as 615, 0, and (320, 240), respectively. The variance factor σ_z^2 that represents uncertainty of the observation is determined as 100.

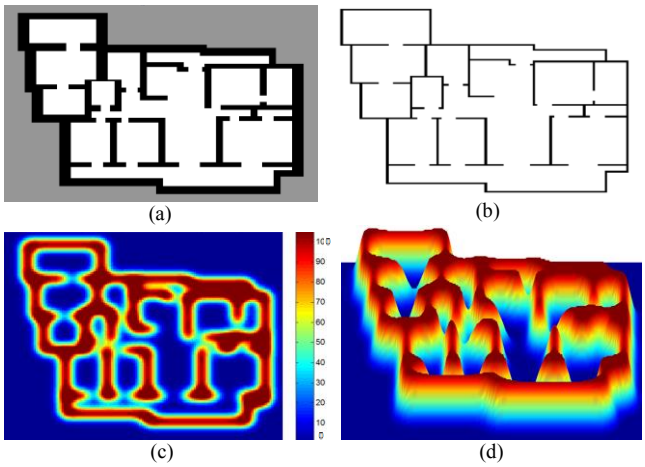


Figure 4. Procedure for generating prior distribution $\boldsymbol{m}(\cdot)$ from grid map information: (a) original grid map, (b) contour lines representing wall information, (c) probability-like distribution after applying Gaussian blur, and (d) its 3D view.

The several stages of the Levenberg-Marquardt iterations (10, 300, and 1,500 iterations) and convergence process for the 2nd camera poses are illustrated as an example in Fig. 6 and Fig. 7, respectively. After 1,500 iterations, most camera poses (colored axes) are accurately converged to the real poses (gray axes). Here, estimated orientation parameters of the 2nd camera (-120 deg, 120 deg, 50 deg) are equivalent to the true states (60 deg, 60 deg, 230 deg) within the framework of the Eular angle for the 3-axis rotation. As shown in Fig. 6 (c), the reason the 3rd, 4th, and 8th distributed cameras were not precisely converged to the true poses can be considered to be a lack of observations; meanwhile, others were accurately estimated. In conclusion, the simulation results show that the complete 6DOF external parameters are estimated very accurately on the condition of sufficient observations and little uncertainty in the localization of the mobile robot.

B. Real Data

Figure 8 shows the experimental setup used to investigate the performance of the proposed method in the real environment. The mobile robot used in the experiment was a Pioneer 3-DX (MobileRobots) mobile robot equipped with an RGB-D sensor (ASUS Xtion Live Pro). The RGB-D sensor was mounted in the frontal direction; it was used to acquire both range data $\boldsymbol{L}$ for rough localization and the image data for the observations $\boldsymbol{Z}_r$. We also implemented a camera network system by using three wireless IP cameras (AXIS

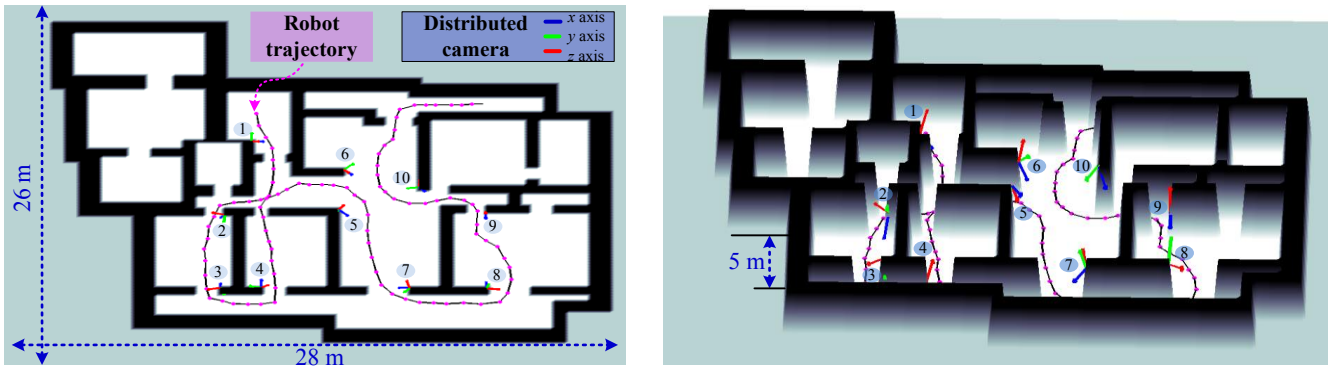
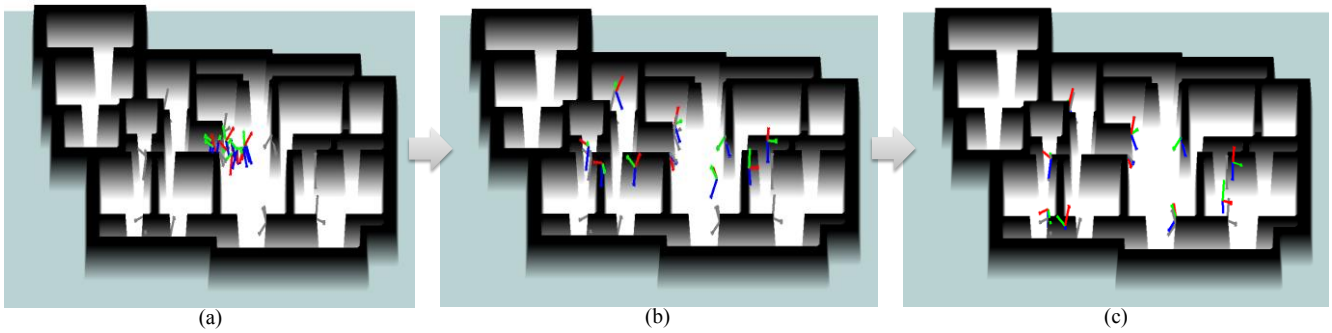


Figure 5. Grid map of simulation environment with true trajectory and poses of virtual camera network.



(a) (b) (c)

Figure 6. Simulation result (gray axes illustrate real poses): (a) after 10 iterations, (b) after 300 iterations, and (c) convergence after 1,500 iterations.

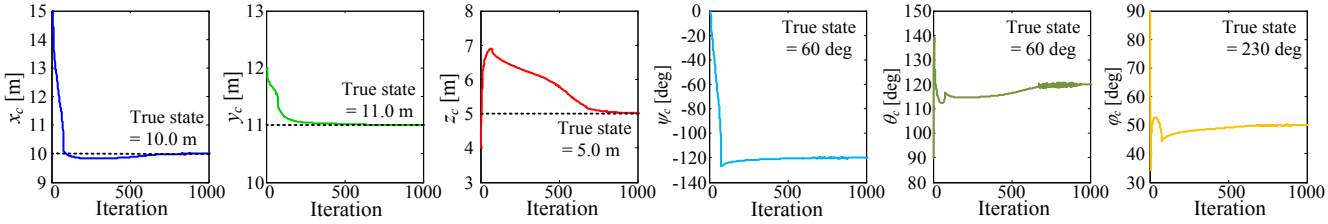


Figure 7. Convergence process of camera state variables (x_c^2, y_c^2, z_c^2, ψ_c^2, θ_c^2, φ_c^2) for 2th distributed camera.

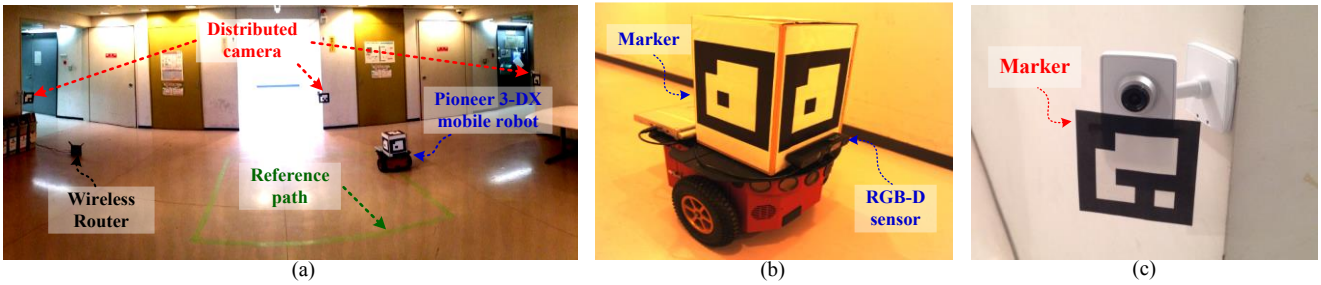
M1004-W). The cameras were mounted on the wall in an indoor environment and these roll and pitch orientation angles were installed 0 deg to easily identify the true angles. The calibration task for the internal camera parameter was done before the experiment. In order to simplify the recognition task, markers were attached on the robot and each IP camera for representing their positions. To compute prior (14), particle filter-based localization was performed and covariance matrix $\boldsymbol{P}_t$ was computed based on the distribution of the particles. In our experiment, the average speed of the robot was about 0.3 m/s and a laptop computer with a 2.6 GHz quad core CPU was used to execute the proposed method.

The calibration results using the real data are shown in Fig. 9. The range data from the RGB-D sensor used for the experiment have very limited range scope (the reliable range data is about 3 m or less); thus, the estimated robot trajectory for the prior information has large uncertainty (i.e., large covariance $\boldsymbol{P}_t$). Despite the use of this uncertain prior, as shown in the results, the proposed method produced 6DOF external parameters and small errors like the simulation results. Here, a couple of causes of error in case of using the real data can be considered: transport delay of the image sent from the camera network, and the error of the map information itself which is the basis of the position information.

After estimating all camera poses, we performed the recovery task for the robot trajectory. In other words, we set $\boldsymbol{Y}$ to the estimated camera poses and optimized (19) for the trajectory $\boldsymbol{X}$ only. The result is illustrated in Fig. 10. After 650 iterations, the rectangular trajectory like the reference path was well recovered because the observations from the distributed cameras with accurately estimated parameters provided reliable constraints for the objective function.

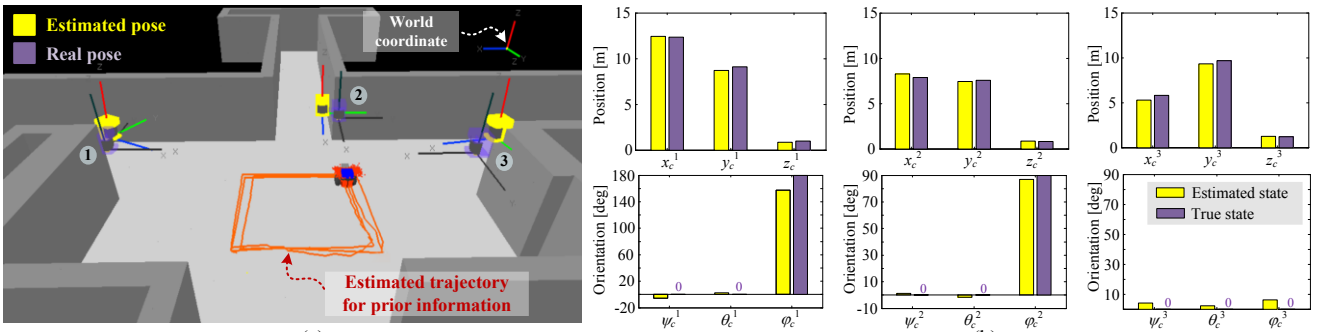
VI. Conclusion

In this paper, an automatic calibration and trajectory reconstruction scheme was developed for the mobile robot and the camera network system, which is most widely used for the intelligent space. Our solution is based on the MAP estimation over the robot trajectory and the external camera parameters. In order to implement complete 6DOF external parameter estimation of the distributed cameras, grid map information that contains wall plane information was



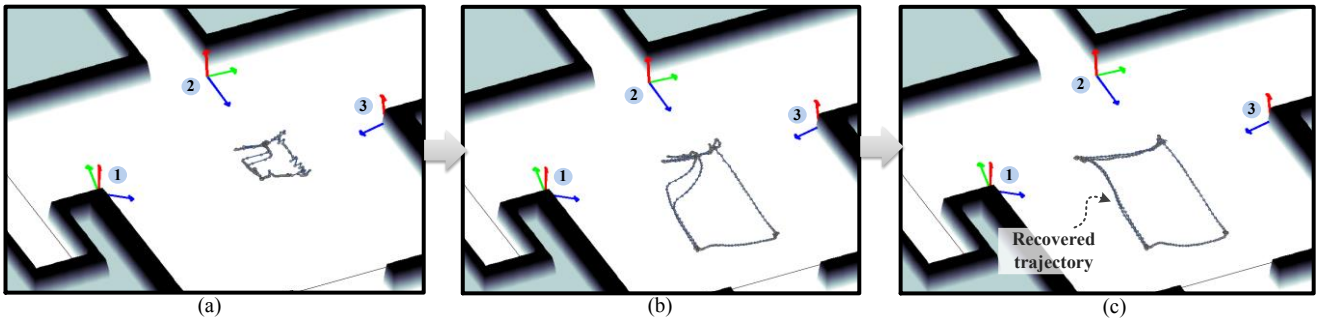
(a) (b) (c)

Figure 8. Experimental setup: (a) experimental environment, (b) Pioneer 3-DX mobile robot, and (c) wireless IP camera for distributed sensor.



(a) (b)

Figure 9. Experiment result for camera parameter estimation: (a) estimated results with prior trajectory of mobile robot (black axes illustrate real camera poses) and (b) comparison with estimated states and true states.



(a) (b) (c)

Figure 10. Experiment result for trajectory reconstruction: (a) after 10 iterations, (b) after 60 iterations, and (c) convergence after 650 iterations.

incorporated into the objective function as prior information. Additionally, we suggested a novel two-way observation model to strengthen the constraints of the objective function. These serve to avoid the local minimum solution; therefore, a global minimum solution can be obtained reliably.

The proposed approach was demonstrated in a simulation of a virtual environment and an experiment on a real environment with a wireless camera network system. We showed that our system is able to estimate and recover all 6DOF distributed camera poses and the robot trajectory accurately.

REFERENCES

[1] J.-H. Lee and H. Hashimoto, "Intelligent Space–concept and contents," *Advanced Robotics*, vol. 16, no. 3, pp. 265–280, 2002.

[2] T. Sato, Y. Nishida, and H. Mizoguchi, "Robotic Room: Symbiosis with Human through Behavior Media," *Robotics and Autonomous Systems*, vol. 18, no. 3, pp. 185–194, 1996.

[3] E. M. Foxlin, "Generalized Architecture for Simultaneous Localization, Auto-Calibration, and Map-building," *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 527–533, 2002.

[4] C. Taylor, A. Rahimi, and J. Bachrach, "Simultaneous Localization, Calibration, and Tracking in an Adhoc Sensor Network," *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pp. 28–33, 2006.

[5] H. Chen, K. Matsumoto, J. Ota, and T. Arai, "Self-calibration of Environmental Camera for Mobile Robot Navigation," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 177–190, 2007.

[6] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous Calibration and Tracking with a Network of Non-Overlapping Sensors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 187–194, 2004.

[7] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed Localization of Networked Cameras," *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pp. 34–42, 2006.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, 2005

[9] S. Kim, J. Kang, and M. J. Chung, "Probabilistic Voxel Mapping Using an Adaptive Confidence Measure of Stereo Matching," *Intelligent Service Robotics*, vol. 6, no. 2, pp. 89–99, 2013.

[10] A. Horunge, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[11] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *The Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.