Data-Driven Kinematic Control for Robotic Spatial Augmented Reality System with Loose Kinematic Specifications

Ahyun Lee, Joo-Haeng Lee, and Jaehong Kim

We propose a data-driven kinematic control method for a robotic spatial augmented reality (RSAR) system. We assume a scenario where a robotic device and a projector-camera unit (PCU) are assembled in an ad hoc manner with loose kinematic specifications, which hinders the application of a conventional kinematic control method based on the exact link and joint specifications. In the proposed method, the kinematic relation between a PCU and joints is represented as a set of B-spline surfaces based on sample data rather than analytic or differential equations. The sampling process, which automatically records the values of joint angles and the corresponding external parameters of a PCU, is performed as an off-line process when an RSAR system is installed. In an on-line process, an external parameter of a PCU at a certain joint configuration, which is directly readable from motors, can be computed by evaluating the pre-built B-spline surfaces. We provide details of the proposed method and validate the model through a comparison with an analytic RSAR model with synthetic noises to simulate assembly errors.

Keywords: Robot kinematics, spatial augmented reality, B-spline surface fitting, data-driven control, real-time systems.

I. Introduction

Spatial augmented reality (SAR) can be used to enhance the efficiency of real-world tasks by directly projecting relevant information over the surface of the work environment, rather than relying on an external monitor such as glasses or a smart phone as in augmented reality [1], [2]. It is conventional to install a large projector near the ceiling of the work space because securing a wide projection area is critical to SAR. In a real-world scenario, however, this type of installation is not always feasible owing to various constraints. For example, a near-ceiling installation may be highly inefficient for an ad hoc task in a spacious room. In such a case, robotic spatial augmented reality (RSAR) can be an effective solution to handle the installation problem [3], [4]. RSAR, which combines a simple robotic mechanism with an SAR technique, enables the position and orientation of a projector to be changed for a wider workspace, as shown in Fig. 1. However, an RSAR system requires an additional procedure to control and monitor the status of the robotic device it employs (where the projector and camera unit are mounted). If the exact kinematic specifications of the parts and assembly involved are available, such as the link lengths and joint orientations, then it can be applicable to a traditional robot kinematics technique.

Assuming that the kinematic specifications are available, Fig. 2 shows a typical execution flow of an RSAR application. It is composed of two steps: on-line and off-line processes. In the off-line process, we calibrate a projector-camera unit (PCU) and formulate the kinematic equations using the known specification. In particular, we set up the kinematic relation

Manuscript received Aug. 6, 2015; revised Oct. 15, 2015; accepted Oct. 29, 2015.

This work was supported by the MOTIE under the Robot R & D Program supervised by the KEIT, Rep. of Korea (10048920, Development of Modular Manipulation System Capable of Self-Reconfiguration of Control and Recognition System for Expansion of Robot Applicability).

Ahyun Lee (azsure@etri.re.kr), Joo-Haeng Lee (corresponding author, joohaeng@etri.re.kr), and Jaehong Kim (jhkim504@etri.re.kr) are with the SW & Content Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.16.0115.0610

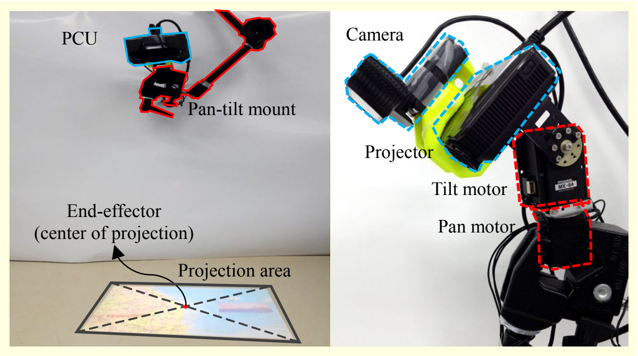


Fig. 1. Configuration of RSAR system considered in this paper: (a) installation on table-top environment and (b) PCU and pan-tilt motors.

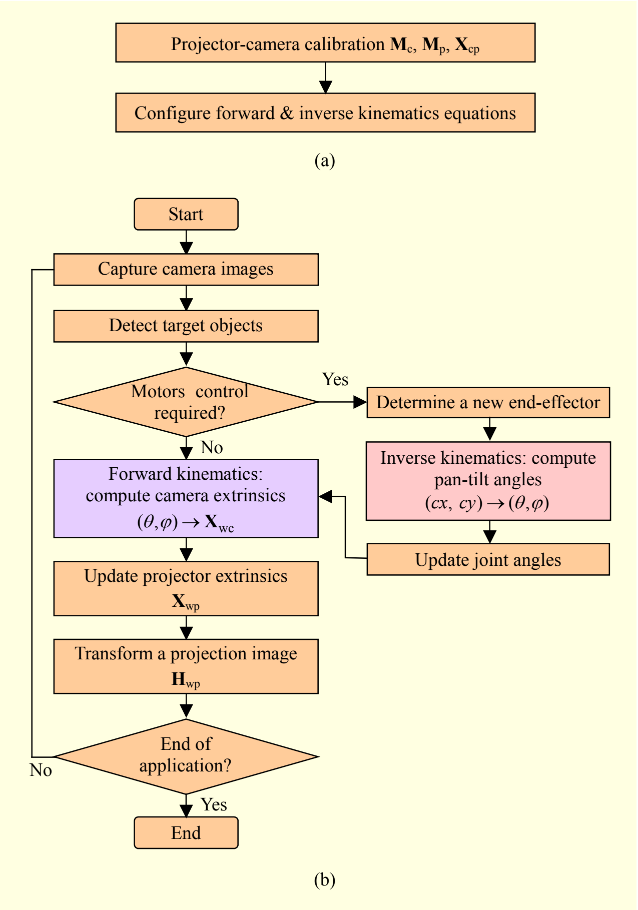

Projector-camera calibration $\mathbf{M}_c$, $\mathbf{M}_p$, $\mathbf{X}_{cp}$
Configure forward & inverse kinematics equations
(a)
Start
Capture camera images
Detect target objects
Motors control required?
Yes
Determine a new end-effector
Inverse kinematics: compute pan-tilt angles $(cx,\ cy) \rightarrow (\theta, \varphi)$
Update joint angles
No
Forward kinematics: compute camera extrinsics $(\theta, \varphi) \rightarrow \mathbf{X}_{wc}$
Update projector extrinsics $\mathbf{X}_{wp}$
Transform a projection image $\mathbf{H}_{wp}$
End of application?
No
Yes
End
(b)

Fig. 2. Typical execution flow of RSAR application: (a) off-line and (b) on-line processes.

between the camera and the projector assuming they are relatively fixed during the operation.

The on-line process of Fig. 2(b) assumes a typical RSAR application where a real-world object is detected from a camera image and the relevant information is augmented through a projector. In the system model of Fig. 2, we use inverse kinematics to determine the position of the projection area and forward kinematics to compute the extrinsic parameters of the moving PCU. The extrinsic parameters of a projector can be computed using the extrinsic parameters of a camera computed during the on-line process and the relative relation calibrated during the off-line process. Finally, a projection image is transformed according to the extrinsic parameters of the projector.

In Fig. 2, computations of both the forward and inverse kinematics are shown based on kinematic equations set up during the off-line process, assuming that the exact kinematic specifications (for example, from a CAD file) are available [5]–[7]. However, in our RSAR scenario, we assume that a moving mechanism (such as a pan-tilt mount) is assembled in an ad hoc manner using an off-the-shelf robot kit and consider the conditions of the work environment. Hence, it is difficult to assume that the exact kinematic specifications are available. Moreover, even if such a user-created robotics (UCR) system is assembled based on a CAD model, it is susceptible to assembly errors, which may invalidate the kinematic equations. To handle this situation, we propose a data-driven kinematic control method for an RSAR system.

1. Related Works

A feature of the SAR technique is that a user's physical environment is augmented with a projected virtual image using a projector, as shown in Fig. 3. Jones and others [8]; Oswald and others [9]; and Wilson and others [10] augment the existing physical environment with projected visualizations to enhance traditional gaming experiences. PlayAnywhere [11], Interactive Dining Table [12], and ActivitySpace [13] are interactive projection vision systems with a compact table projection and sensing system. Their multiple-touch interfaces naturally support a direct manipulation style of interaction with virtual objects, where a user can initiate object interaction through touch and natural gestures. HP Sprout [14] is a new computing platform using SAR. Hewlett Packard's Sprout touch mat takes the place of a traditional mouse and keyboard. Its projector can place a virtual keyboard on a capacitive touch panel; thus, users can interact with the PC and its applications.

VisiCon [15] is a robot navigation game using a handheld projector. It enables a user to intuitively control a robot and share the displayed robot information with others through a projected display. LuminAR [16], developed by MIT Media Lab, is a robotic projector-camera system that can dynamically change projection parameters. The robotic system is similar to a desk lamp, but uses a pico-projector instead of a light bulb for projecting a virtual image at the user-desired position. A projector system [17] and a ubiquitous display [18] are used in movable mobile projection robots. These platforms enable images to be projected on any surface, such as a wall, a floor,

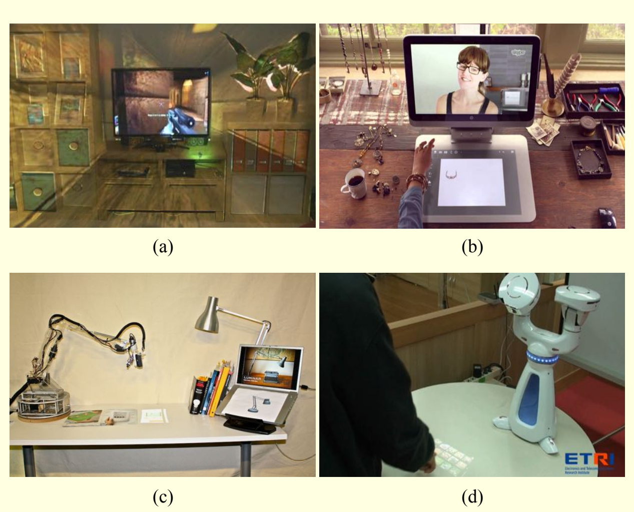


Fig. 3. Example of SAR/RSAR systems: (a) MS IllumiRoom [8], (b) HP sprout [14], (c) MIT LuminAR [16], and (d) ETRI Future Robotic Computer [4].

or an object.

The Future Robotic Computer [4], developed by the Electronic Telecommunications Research Institute (ETRI), enables the end position of a robot device equipped with a PCU to be changed. The Future Robotic Computer is the leading RSAR system, and actively applies an SAR technique with a robotic device for user interaction. It can be applied to various applications that need to expand or change the projection area or the user environment.

The initial idea of data-driven control was conceived in [19]. In this paper, we advance this technique in multiple ways to be applied in a real-world scenario. First, sampling is performed to cover an entire work area, rather than a fixed area, by moving the calibration markers. The sample data are then merged into a single parametric form. We evaluate the accuracy of the proposed approach by comparing with an analytic RSAR model with synthetic noises.

The rest of this paper is structured as follows. In Section II, we overview the RSAR system model and describe the limitations of the previous control method under loose kinematic specifications. In Section III, we describe the proposed method in detail in the context of a proof-of-concept application. In Section IV, we show our experimental results. Finally, in Section V, we conclude the present paper with a discussion of future works.

II. RSAR System Model

The RSAR system, which has an attached PCU at the end position of the robot device, recognizes real-world objects using a camera, and projects virtual images on the surface of the objects. In this section, we describe the projector-camera calibration, RSAR kinematic control, and loose kinematic specifications.

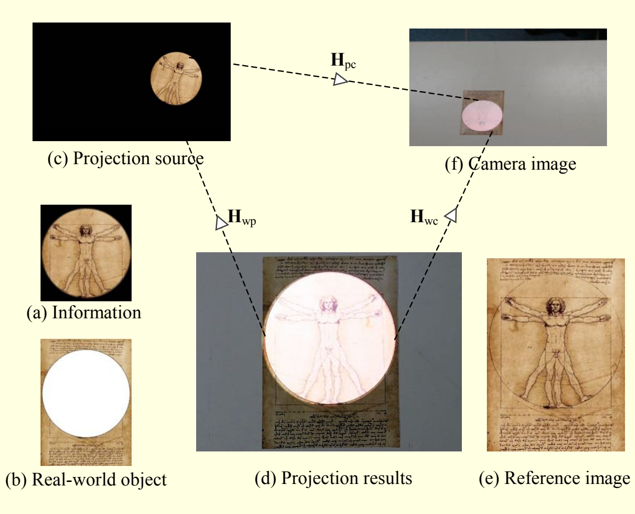


Fig. 4. Major coordinate system in RSAR and transformation among coordinates: (a) circular sub image of (e), (b) modified (e) by removing (a) and printed on paper, (c) transformation of (a) as projection source, (d) projection of (c) onto (b) captured using external camera, (e) *The Vitruvian Man* by Leonardo da Vinci, and (f) image of (d) from camera of PCU.

1. Projector-Camera Calibration

We use Zhang's method [20] for the camera calibration. The correlation between $\mathbf{p}_c$, which is a point in the camera coordinates, and $\mathbf{p}_w$, which is a point in the real-world coordinates, is defined through homography matrix $\mathbf{H}_{wc}$. In (1), $\mathbf{H}_{wc}$ is combined with intrinsic parameter $\mathbf{M}_c$ and extrinsic parameter $\mathbf{X}_{wc}$ of a camera.

$$\mathbf{p}_c = \mathbf{M}_c \mathbf{X}_{wc} \mathbf{p}_w = \mathbf{H}_{wc} \mathbf{p}_w = \mathbf{H}_{cw}^{-1} \mathbf{p}_w. \tag{1}$$

In the case of projector calibration, we use Tsai's method [21] instead of Zhang's method, which has to know in advance the resolution of a projected image. As a result, the transformation matrix $\mathbf{X}_{cp}$ between the camera and the projector coordinates is defined as [22]–[26]

$$\mathbf{X}_{wp} \mathbf{X}_{wc}^{-1} = \mathbf{X}_{wp} \mathbf{X}_{cw} = \mathbf{X}_{cp} = \mathbf{X}_{pc}^{-1}. \tag{2}$$

The projector homography matrix $\mathbf{H}_{wp}$ is computed using $\mathbf{X}_{wc}$, projector intrinsic parameter $\mathbf{M}_p$, and the correlation matrix $\mathbf{X}_{cp}$ between the camera and the projector.

$$\mathbf{M}_p \mathbf{X}_{cp} \mathbf{X}_{wc} = \mathbf{M}_p \mathbf{X}_{wp} = \mathbf{H}_{wp} = \mathbf{H}_{pw}^{-1}, \tag{3}$$

$$\mathbf{p}_c = \mathbf{H}_{wc} \mathbf{H}_{pw} \mathbf{p}_p = \mathbf{H}_{pc} \mathbf{p}_p. \tag{4}$$

Figure 4 shows the correlation of the coordinate system for the RSAR. The camera, projector, and real-world coordinate systems are defined by the calibrated PCU in the RSAR system [27]. To combine the virtual image in Fig. 4(a) and the printed image on paper, shown in Fig. 4(b), the projection source needs to be warped according to the position and orientation of a projector, such as in Fig. 4(c). Figure 4(d) shows the registered result of the warped projection image and the printed image on paper, such as the original image in Fig. 4(e).

2. RSAR Kinematic Control

An *end-effector* is the interaction force device of a robot, which is placed at the end position of a robot arm. On the other hand, an end-effector is defined differently in the RSAR system. In this paper, end-effector is taken to mean the position of the center point of a projected image. In Fig. 5(a), the analytic RSAR model is designed as a two-axis robotic arm based on the kinematic specifications, such as the lengths of the links and the positions of the joints.

$$\text{End-effector} = \begin{bmatrix} \dfrac{1}{\cos q_0}(l_1 + l_2 + l_0 \sin q_0) \\ \dfrac{1}{\cos q_0}\{l_0 + (l_1 + l_2)\sin q_0\} \\ 0 \end{bmatrix}, \tag{5}$$

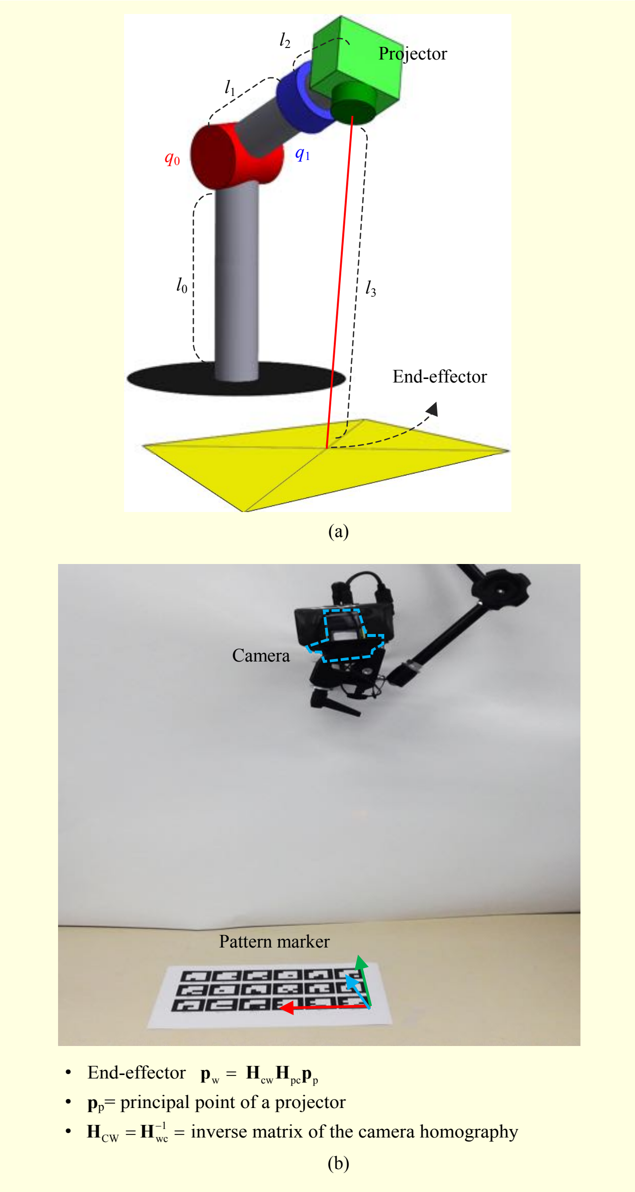
$$l_3 = -\frac{l_0 + l_1 \sin q_0 + l_2 \sin q_0}{\cos q_0 \cos q_1}. \tag{6}$$

In (5), q_0 is the angle of the tilt motor, and q_1 is the angle of the pan motor. The length of link l_0 is from the origin point to the axis of rotation of the tilt motor, and l_1 is the length of a link from the tilt motor to the pan motor. The length of link l_2 is from the axis of rotation of the pan motor to the principal point of the projector. The length of the last link, l_3, is variable according to the joint angles in (6).

The forward kinematic equation is geometrically defined based on the joints and links. In the case of an inverse kinematic equation, we use the inverse matrix of the forward kinematic equation through pseudoinverse or Jacobian transpose methods. In the absence of links or joints based on the kinematic specifications, the position of the end-effector can be computed using a camera and a pattern marker in the RSAR, as shown in Fig. 5(b). However, whenever a PCU is moved, it needs to conduct camera pose estimation using a pattern marker. In addition, it is impossible to use the inverse kinematics without kinematic specifications.

3. Loose Kinematic Specifications

A minimalism-based UCR has a strong point (which is of a reduced complexity), seeks the lowest cost, and minimizes the development time [28]. Users can easily make a robot system using robot kits available on the market. Our UCR-based RSAR system, which was designed to fit a table-top environment, as shown in Fig. 1, enables the usable range to be dynamically expanded through a moving PCU.



- End-effector $\mathbf{p}_w = \mathbf{H}_{cw}\mathbf{H}_{pc}\mathbf{p}_p$
- $\mathbf{p}_p$= principal point of a projector
- $\mathbf{H}_{CW} = \mathbf{H}_{wc}^{-1}$ = inverse matrix of the camera homography

(b)

Fig. 5. Comparison of calculation method of postion of end-effector based on (a) explicit kinematic specification and (b) camera pose estimation using pattern marker.

However, the UCR is usually fabricated without a CAD model. Even if the UCR is made based on a CAD model, it easily incurs assembly errors. Thus, it is difficult to integrate the kinematic equations for a UCR-based system with loose kinematic specifications. In this paper, we propose a data-driven kinematic control method with loose kinematic specifications for a UCR-based RSAR system.

III. Proposed Method

The proposed method is composed of the acquisition, representation, and evaluation of sample data for kinematic control. We assume that the UCR-based RSAR system is a two-axis robotic arm without kinematic specifications. In addition, it can compute the projector extrinsic parameters by conducting only a camera pose estimation because the calibrated camera and projector are physically fixed.

1. Acquisition of Sampled Data

The sampled-data for kinematic control are composed of joint angles and camera extrinsic parameters, as shown in Table 1. The rotation radius of the joints should be preset prior to the acquisition of the sampled data depending on the user's application environment. Pan and tilt motors are rotated at regular intervals within the range of configured rotation radii. In addition, the camera pose estimation is conducted by detecting a pattern marker in the camera input frames at each rotated motor angle. The finding-a-pattern-marker algorithm is implemented based on OpenCV-based marker recognition [29].

In previous research [19], we had a problem in that the acquisition area of the sampled data was too narrow, because of a fixed pattern marker used on the ground. In this paper, we enabled moving the pattern marker to expand the acquisition area of the sampled data. First, a pattern marker should be fixed on the ground of the table-top environment. In addition, the motors are rotated at regular intervals within the range of the configured rotation radii. Camera extrinsic parameters are computed at each angle of the rotated motors in Fig. 6(b). After acquisition of the sampled data, if a user wants to acquire the sampled data in another area, then they can move the marker to a desired area, as shown in Fig. 6(c). In addition, the position of the moved marker in the world coordinate in Fig. 6(d) should be computed. The acquisition process of the sampled data is repeated by moving to Fig. 6(a).

To compute the position of the moved marker, the following steps should be conducted:

1) Compute the camera extrinsic parameter $\mathbf{X}_{\mathrm{wc}}$ before the marker is moved.
2) After moving the marker, compute the corner points $\mathbf{p}_{\mathrm{c}}$ of the moved marker in the camera coordinate.
3) Compute $\mathbf{p}_{\mathrm{w0}}(=\mathbf{X}_{\mathrm{wc}}\mathbf{p}_{\mathrm{c}})$ and perspective transformation matrix $\mathbf{T}$ between $\mathbf{p}_{\mathrm{c}}$ and $\mathbf{p}_{\mathrm{w0}}$ using RANSAC for minimizing reprojection errors.
4) Finally, compute the moved position $\mathbf{p}_{\mathrm{w1}}(=\mathbf{T}\mathbf{p}_{\mathrm{c}})$ of a marker in the world coordinates.

Table 1. Sampled data.

Index	Joint angles	Camera extrinsic parameter
i	θ_i, φ_i	$\mathbf{X}_{\mathrm{wc}_i}$

Table 2. Input and output data for kinematic control.

	Forward kinematics	Inverse kinematics
Input	(θ, φ)	Position of end-effector (cx, cy)
Output	$\mathbf{X}_{\mathrm{wc}}$	(θ, φ)

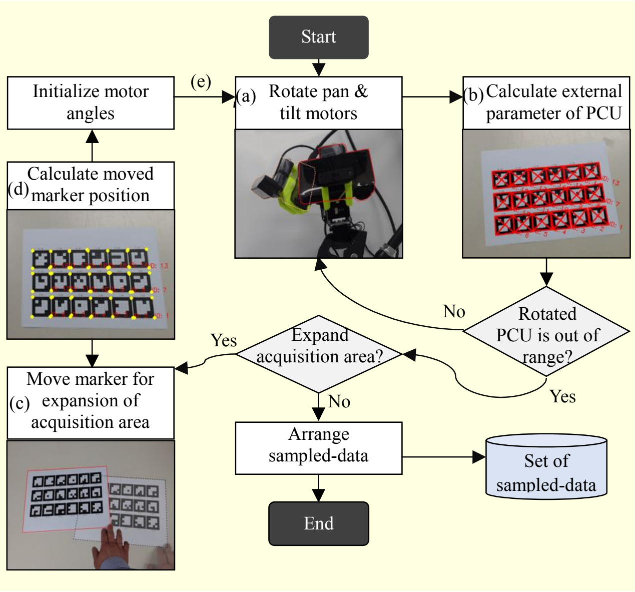

Start
Initialize motor angles
(e)
(a) Rotate pan & tilt motors
(b) Calculate external parameter of PCU
(d) Calculate moved marker position
No
Rotated PCU is out of range?
Yes
Expand acquisition area?
No
(c) Move marker for expansion of acquisition area
Arrange sampled-data
Set of sampled-data
End

Fig. 6. Acquition process of sampled data using printed pattern marker at multiple locations.

The acquired sampled data are classified as input or output data according to the forward or inverse kinematics. In the case of forward kinematics, the position of the end-effector is computed by measuring the angle of the joints. At this time, the joint angles are the input data, and a camera extrinsic parameter is the output data used to compute the position of the end-effector. In the case of inverse kinematics, the target position of the end-effector is the input data, and the joint angles are the output data. Table 2 shows the form of the classified input and output data for kinematic control.

To compute the output data at unacquired joint angles, we need to interpolate the set of sampled data. In particular, in the case of forward kinematics, the sampled data need to be decomposed. The camera extrinsic parameter $\mathbf{X}_{\mathrm{wc}}$, which is the output data of the forward kinematics, is decomposed into a total of six factors: the rotation values, *rx*, *ry*, and *rz*, and the translation values, *tx*, *ty*, and *tz*, of the *x*-, *y*-, and

Table 3. Recomposed sampled data for proposed forward kinematics.

	Input data		Output data					
			Rotation			Translation		
Index	Pan	Tilt	rx	ry	rz	tx	ty	tz
i	θ_i	φ_i	rx_i	ry_i	rz_i	tx_i	ty_i	tz_i

z-axis, respectively. In (8), $\mathbf{R}_{ij}$ indicates the ith row and jth column of $\mathbf{R}$ [30].

$$\mathbf{X}_{\text{wc}} = [\mathbf{R} \mid \mathbf{t}] = \left[\mathbf{R}(rx, ry, rz) \mid (tx, ty, tz)^{\text{T}}\right], \tag{7}$$

$$\begin{aligned} rx &= \tan^{-1}\left(\mathbf{R}_{21}, \mathbf{R}_{11}\right), \\ ry &= \sin^{-1}\left(-\mathbf{R}_{31}\right), \\ rz &= \tan^{-1}\left(\mathbf{R}_{32}, \mathbf{R}_{33}\right). \end{aligned} \tag{8}$$

The six decomposed factors are recomposed with joint angles θ and φ, as shown in Table 3, for the forward kinematics.

2. Representation of Sampled Data and Evaluation for RSAR Kinematic Control

In this paper, we use B-spline surface fitting for estimating the output data at the unacquired joint angles. B-spline surface fitting is suitable for interpolating in various unpredictable forms of the surfaces. In (9), $\mathbf{N}_{d_0,i}(u)$ and $\mathbf{N}_{d_1,j}(v)$ are basis functions for generating the B-spline surface function $\mathbf{B}(u, v)$, and d_0 and d_1 are degrees of the B-spline surface function. The B-spline surface has control point matrix $\mathbf{Q}$, which is formed in two-dimensional arrays [31]–[33].

$$\mathbf{B}(u, v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{N}_{d_0,i}(u)\mathbf{N}_{d_1,j}(v)\mathbf{Q}_{i,j}. \tag{9}$$

The sampled data need to be recomposed as three-dimensional coordinates to present a B-spline surface. For example, in the case of the rotation factor, rz of the z-axis in a camera extrinsic parameter is recomposed as $\left(\theta_i, \varphi_i, rz_i\right)$ for the forward kinematics. The recomposed sampled data are shown in Fig. 7(a). The results of the presented B-spline surface are shown in Fig. 7(b).

To compute the output data from the presented surfaces, we compute the z-axis value according to the input data value of the x- and y-axis. In the case of forward kinematics, we measure the joint angles (θ, φ) as input data, and compute the output data rx, ry, rz, tx, ty, and tz from the six represented surfaces. In addition, the six computed factors are recomposed into the camera extrinsic parameter $\mathbf{X}_{\text{wc}}$ in (10). A total of eight surfaces (six for the forward kinematics, and two for the inverse kinematics) are presented for the proposed kinematic control. Figure 8 shows the results of surfaces presented by 8 × 8 control points; d_0 and d_1 are cubic values in (9).

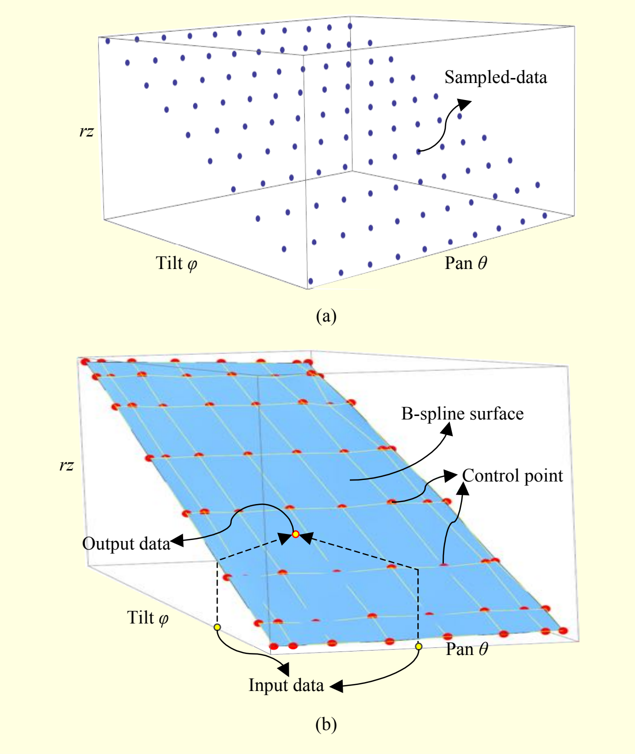


Fig. 7. Sampled data and their B-spline surface form.

3. Example of UCR-Based RASR System

An example of the UCR-based RSAR system enables a real-world object to be detected by a camera [34] and virtual contents to be projected by a projector. When an object is moved out of the projection area, the RSAR system can maintain the image registration by dynamically changing the kinematic parameters through the proposed inverse kinematics shown in Fig. 9(a). In addition, we compute the position of the PCU using the proposed forward kinematics. The source of projection image $\mathbf{P}_{\text{w}}$ is designed in the world coordinates. So,

$$\mathbf{X}_{\text{wc}} = \begin{bmatrix} \cos ry \cos rz & \sin rx \sin ry \cos rz - \cos ry \sin rz & \cos rx \sin ry \cos rz + \sin rx \sin rz & tx \\ \cos ry \sin rz & \sin rx \sin ry \sin rz + \cos ry \cos rz & \cos rx \sin ry \sin rz - \sin rx \cos rz & ty \\ -\sin ry & \sin rx \cos ry & \cos rx \cos ry & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

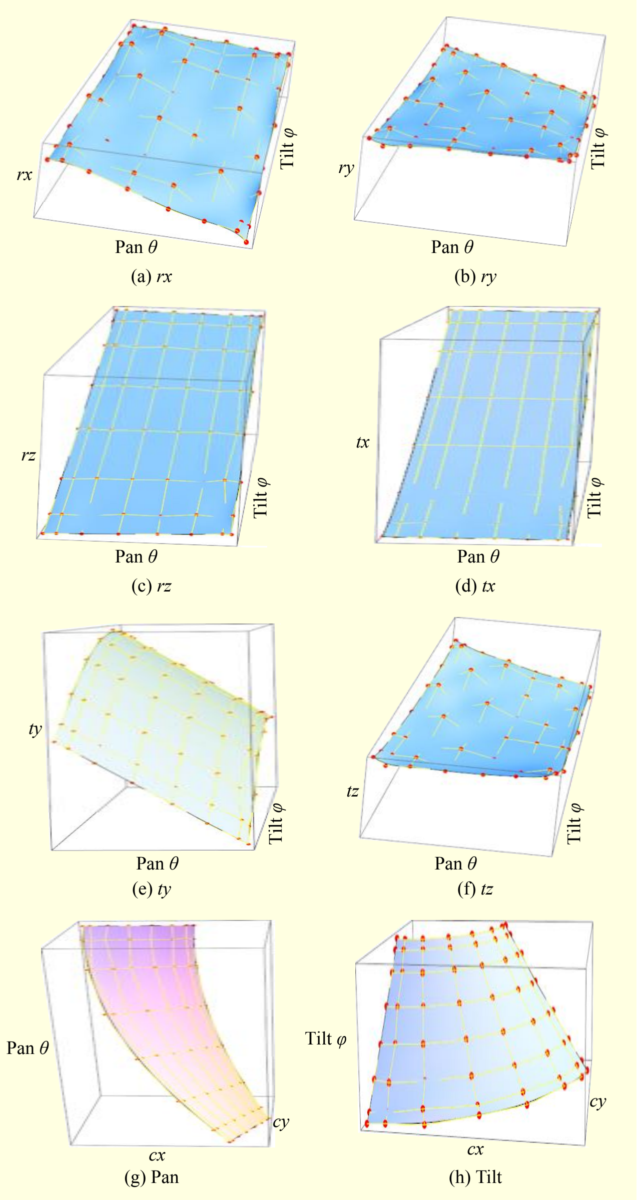


Fig. 8. Representation results of sampled data: (a) through (f) surfaces for forward kinematics, and (g), (h) surfaces for inverse kinematics.

depending on the computed position of the PCU, an augmented projection image $\mathbf{P}_\text{p}$ is rendered in Fig. 9(b) [27].

$$\mathbf{P}_\text{p} = \mathbf{M}_\text{p}\mathbf{X}_\text{cp}\mathbf{X}_\text{wc}\mathbf{P}_\text{w} = \mathbf{M}_\text{p}\mathbf{X}_\text{wp}\mathbf{P}_\text{w} = \mathbf{H}_\text{wp}\mathbf{P}_\text{w}\,. \tag{11}$$

Finally, the rendered image is projected onto the surface of an object, as shown in Fig. 9(c).

IV. Experiments

The experiments were conducted on a desktop computer with a 2.67 GHz Intel® core™ i7 CPU, using a USB 2.0 camera (Logitech C920 with a 640 × 480 resolution), a projector (Optoma P320 with a 1,280 × 720 resolution), and joints (Robotis Dynamixel MX-64). We acquired 12 × 12 sampled data for the preset rotation radii of the joints, at a pan motor angle of 17.58º to 149.45º, and a tilt motor angle of 237.36º to 263.73º. Our computer vision algorithm was programmed with OpenCV [29].

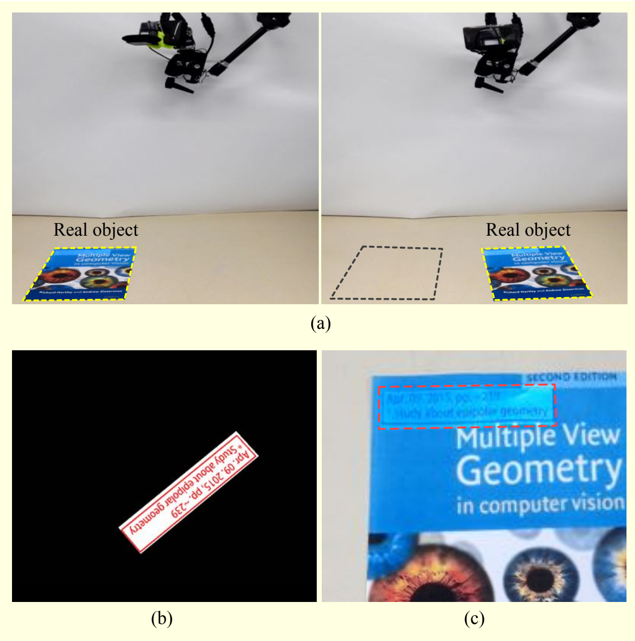


Fig. 9. Example for RSAR application: (a) change in position of PCU according to real moving object, (b) warped image source for projection, and (c) projection result.

We measured the accuracy of the proposed kinematic control in a UCR-based RSAR system without kinematic specifications by making an analytic RSAR model, as shown in Fig. 5(a). The correlation between joint angles and the position of the end-effector is presented through the contour plots in Fig. 10. The contour plots show changes in the joint angles according to the fixed values of *cx* and *cy*. To model a UCR-based RSAR system in which irregular movements occur during the system operation, we added random noise (−0.5 mm to 0.5 mm) at the position of the joints where were acquired of the sampled data.

Figure 10(b) shows a contour plot with random noise. The red and green lines represent the results through kinematic control based on the kinematic specifications. The blue points indicate the results of the proposed kinematic control. We acquired the sampled data with random noise, represented by B-spline surfaces, and computed joint angles *θ* and *φ* according to the fixed value of the end-effector (*cx*, *cy*). The distribution form of the blue points is dependent on random noise generated at the time of the sampled data acquisition.

In another experiment, we examined the accuracy of the proposed method by changing the position of the end-effector along the outline of a circle with various radii of random noises.

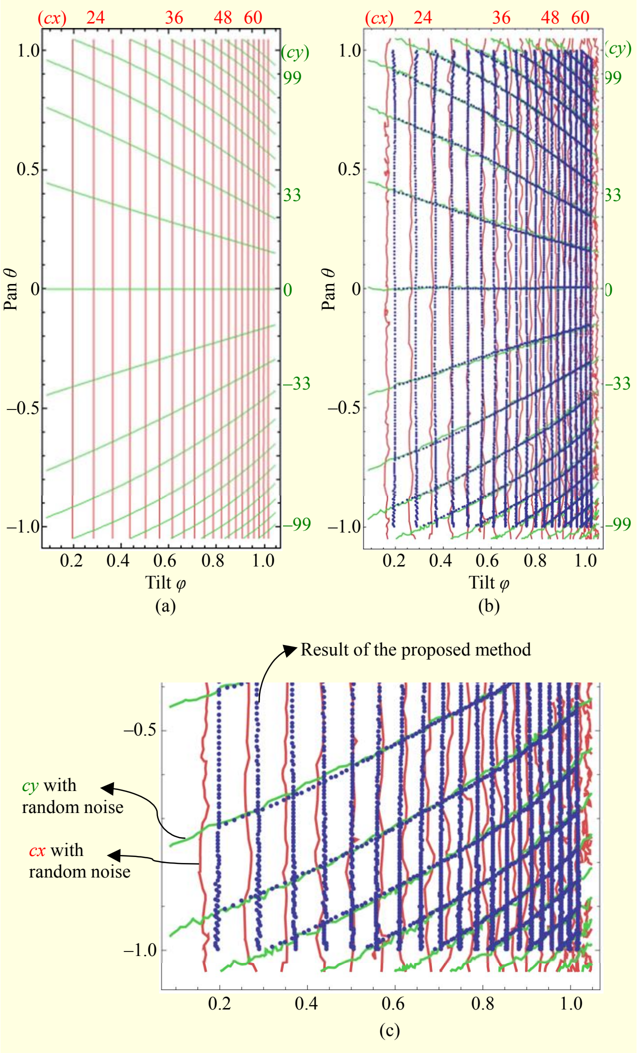


Fig. 10. Correlation between joint angles and position of end-effector with loose kinemtatic specifications through contour plot: (a) changes in joint angles according to fixed values of *cx* and *cy* in analytic RSAR model, (b) contour plot with random noise, where blue points are results of proposed method, and (c) magnified image of (b).

First, we computed 126 points of the outline of a circle in real-world coordinates, and computed the joint angles corresponding to each point through inverse kinematics using the kinematic specifications shown in Fig. 11(a). Second, we used the sampled data with various radii of random noises for modeling with loose kinematic specifications. Finally, we applied the proposed forward kinematics for computing the position of the end-effector through the computed joint angles. The position results of the end-effector are shown in Figs. 11(b) through 11(d). The distorted form of a circle is dependent on random noise generated at the time of acquisition of the sampled data.

Table 4 shows the root-mean-square error (RMSE) results of the accuracy experiment of the proposed kinematic control. The reference value for error determination is the position of the end-effector computed without random noise in the kinematic specification–based analytic RSAR model. First, we compared the proposed method without random noise. In addition, where random noise is present, we compared the results of a method using kinematic specifications with those of the proposed method. Our method was determined to have a lower RMSE than the method with kinematic specifications because we observed a noise smoothing effect owing to the use of B-spline surface fitting.

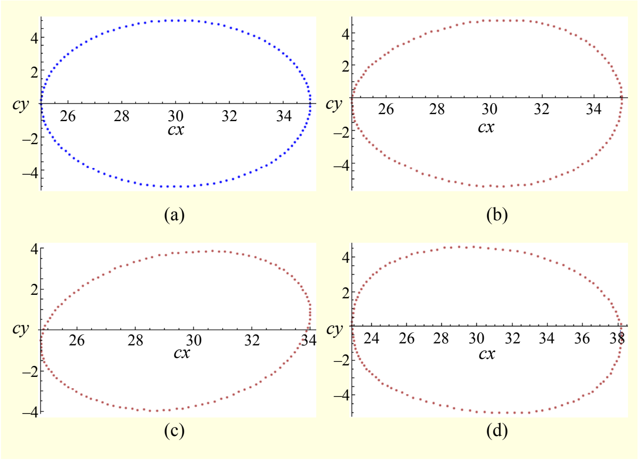


Fig. 11. Position of end-effector along outline of circle with random noises in real-world coordinates: (a) analytic RSAR model without noise and (b)–(d) inverse kinematics results using proposed method with random-noise radii (4 mm, 8 mm, and 12 mm).

Table 4. RMSE of position of end-effector between analytic RSAR model based on kinematic specifications and its model with random noise for modeling UCR-based RSAR with loose kinematic specifications.

Without noise		With random noise				
Proposed method		Noise radii (mm)	Kinematic specifications		Proposed method	
cx	*cy*		*cx*	*cy*	*cx*	*cy*
0.0296	0.2585	4	1.8144	2.4387	0.9231	1.1977
		8	3.3294	4.9929	1.9215	2.9191
		12	5.1448	7.8924	2.6224	5.7513

V. Conclusion

The traditional control method for an RSAR system based on exact kinematic specifications is inappropriate in a UCR-based scenario in which such specifications are unavailable or invalidated. The proposed data-driven kinematic control method enables an RSAR system to be controlled through loose kinematic specifications.

The proposed method exploits the sample data acquired in an off-line process and represents them in a compact B-spline surface form, which replaces the analytic kinematic equations. During the on-line process, the relevant control parameters are computed by evaluating the B-spline surfaces using the measurable joint angles as parameters. To evaluate the accuracy of the proposed method, we built an analytic RSAR model with synthetic noises to simulate an assembly error. We were able to observe a smoothing effect that mitigates the simulated error during the processes of B-spline fitting and evaluation. We believe that our approach is a highly practical solution to a UCR-based RSAR system with loose kinematic specifications.

As future work, we need to integrate better computer vision techniques to minimize the calibration and registration errors. For example, coupled-line cameras and projectors are compact geometric approaches to calibrate a projector-camera system [24], [35]. In addition, we will apply the proposed method in an RSAR system with more types of kinematic configurations and higher DOFs. We also plan to verify the feasibility of the data-driven kinematic control over a wide range of applications such as assembly guide [36], product design [37], human–vehicle interaction [38], and gesture-based interaction [39].

References

[1] O. Bimber and R. Raskar, "*Spatial Augmented Reality: Merging Real and Virtual Worlds*," MA, USA: AK Peters, 2005.

[2] R. Azuma et al., "Recent Advances in Augmented Reality," *Comput. Graph. Appl*, *IEEE*, vol. 21, no. 6, Aug. 2001, pp. 34–47.

[3] J.-H. Lee et al., "Issues in Control of a Robotic Spatial Augmented Reality System," *Trans. Soc. CAD/CAM Engineers*, vol. 16, no. 6, Dec. 2011, pp. 437–448.

[4] J.-H. Lee et al., "FRC Based Augment Reality for Aiding Cooperative Activities," *Int. Symp. Robot Human Interactive Commun. IEEE*, Gyeongju, Rep. of Korea, Aug. 26–29, 2013, pp. 294–295.

[5] J.J. Craig, "*Introduction to Robotics: Mechanics and Control*," NJ, USA: Pearson Education International, 2005.

[6] P. Corke, "*Robotics, Vision and Control: Fundamental Algorithms in MATLAB Vol 73*," Brisbane, Australia: Springer Science & Business Media, 2011.

[7] J.-H. Lee, J. Kim, and H. Kim, "A Note on Hybrid Control of Robotic Spatial Augmented Reality," *Ubiquitous Robots Ambient Intell.*, *IEEE*, Incheon, Rep. of Korea, Nov. 23–26, 2011, pp. 621–626.

[8] B.R. Jones et al., "Illumiroom: Peripheral Projected Illusions for Interactive Experiences," *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, ACM, IL, USA, Apr. 27–May 2, 2013, pp. 869–878.

[9] P. Oswald, J. Tost, and R. Wettach, "The Real Augmented Reality: Real-Time Game Editor in a Spatial Augmented Environment," *Proc. Conf. Adv. Comput. Entertainment Technol.*, Funchal, Portugal, Nov. 11–14, 2014, no. 32.

[10] A.D. Wilson et al., "Steerable Augmented Reality with the Beamatron," *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, Cambridge, MA, USA, Oct. 7–10, 2012, pp. 413–422.

[11] A.D. Wilson, "PlayAnywhere: a Compact Interactive Tabletop Projection-Vision System," *Proc. Annual ACM Symp. User Interface Softw. Technol.*, Washington, DC, USA, Oct. 23–27, 2005, pp. 83–92.

[12] F. Echtler and R. Wimmer, "The Interactive Dining Table, or Pass the Weather Widget, Please," *Proc. ACM Int. Conf. Interactive Tabletops Surfaces*, Dresden, Germany, Nov. 16–19, 2014, pp. 419–422.

[13] S. Houben, P. Tell, and J.E. Bardram, "ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space," *Proc. ACM Int. Conf. Interactive Tabletops Surfaces*, Dresden, Germany, Nov. 16–19, 2014, pp. 119–128.

[14] Fast Company Content Studios Staff Writer, "Slideshow: from Seed to Sprout, a Story of Innovation Inside the Walls of HP," HP, USA, *HP Matter: The Transp. Issue*, 2014. Accessed July, 2015. https://sprout.hp.com/

[15] K. Hosoi et al., "VisiCon: a Robot Control Interface for Visualizing Manipulation Using a Handheld Projector," *Proc. Int. Conf. Adv. Comput. Entertainment Technol.*, Salzburg, Austria, June 13–15, 2007, pp. 99–106.

[16] N. Linder and P. Maes, "Luminar: Portable Robotic Augmented Reality Interface Design and Prototype," *Adjunct Proc. Annual ACM Symp. User Interface Softw. Technol.*, New York, UUSA, Oct. 3–6, 2010, pp. 395–396.

[17] J. Park and G.J. Kim, "Robots with Projectors: an Alternative to Anthropomorphic HRI," *Proc. ACM/IEEE Int. Conf. Human Robot Interaction*, La Jolla, CA, USA, Mar. 9–13, 2009, pp. 221–222.

[18] K. Maegawa et al., "Ubiquitous Display 2.0: Development of New Prototype and Software Modules for Improvement," *Ubiquitous Robots Ambient Intell.*, Jeju, Rep. of Korea, Oct. 31–Nov. 2, 2013, pp. 102–107.

[19] A. Lee, J.-H. Lee, and J.-H. Lee, "Sampling-Based Control of SAR System Mounted on a Simple Manipulator," *Trans. Soc. CAD/CAM Engineers*, vol. 19, no. 4, Dec. 2014, pp. 356–367. (in Korean)

[20] Z. Zhang, "A Flexible New Technique for Camera Calibration," *Pattern Anal. Mach. Intell.*, *IEEE Trans.*, vol. 22, no. 11, Nov. 2000, pp. 1330–1334.

[21] R.Y. Tsai, "A Versatile Camera Calibration Technique for High-

Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. Robot. Autom.*, vol. 3, no. 4, Aug. 1987, pp. 323–344.

[22] J.-H. Lee et al., "Calibration Issues in FRC: Camera, Projector, Kinematics Based Hybrid Approach," *Ubiquitous Robots Ambient Intell.*, Daejeon, Rep. of Korea, Nov. 26–28, 2012, pp. 218–219.

[23] R. Sukthankar, R.G. Stockton, and M.D. Mullin, "Smarter Presentations: Exploiting Homography in Camera-Projector Systems," *Int. Conf. Comput. Vis. IEEE*, Vancouver, Canada, July 7–14, 2001, pp. 247–253.

[24] J.-H. Lee, "An Analytic Solution to Projector Pose Estimation Problem," *ETRI J.*, vol. 34, no. 6, Dec. 2012, pp. 978–981.

[25] T. Okatani and K. Deguchi, "Autocalibration of a Projector-Camera System," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 27, no. 12, Dec. 2005, pp. 1845–1855.

[26] R.T. Chadalavada et al., "That's on my Mind! Robot to Human Intention Communication through on-board Projection on Shared Floor Space," *European Conf. Mobile Robots*, Lincoln, UK, Sept. 2–4, 2015.

[27] A. Lee, J.-D. Suh, and J.-H. Lee, "Interactive Design of Planar Curves Based on Spatial Augmented Reality," *Proc. Companion Publication Int. Conf. Intell. User Interfaces Companion*, Santa Monica, CA, USA, Mar. 19–22, 2013, pp. 53–54.

[28] I.-W. Park and J.-O. Kim, "Philosophy and Strategy of Minimalism-Based User Created Robots (UCRs) for Educational Robotics-Education, Technology and Business Viewpoint," *Int. J. Robots, Education Art*, vol. 1, no. 1, Feb. 2011, pp. 26–38.

[29] G. Bradski and A. Kaebler, "*Leaning OpenCV: Comput. Vision with the OpenCV Library*," CA, USA: O'Reilly Media, 2008.

[30] K. Shoemake, "Euler Angle Conversion," in *Graphics Gems IV*, PA, USA: Morgan Kaufmann, 1994, pp. 222–229.

[31] G. Farin, "*Curves and Surfaces for CAGD*," AZ, USA: Morgan Kaufmann, 2001

[32] H.-J. Park and J.-H. Lee, "B-Spline Curve Fitting Based on Adaptive Curve Refinement Using Dominant Points," *Comput. Aided Des.*, vol. 39, no. 6, June 2007, pp. 439–451.

[33] H.-J. Park and J.-H. Lee, "Adaptive B-Spline Volume Representation of Measured BRDF Data for Photorealistic Rendering," *J. Computational Des. Eng.*, vol. 2, no. 1, Jan. 2015, pp. 1–15.

[34] H. Bay, T. Tuytelaars, and L.V. Gool, "Surf: Speeded up Robust Features," *Comput. Vision–ECCV*, Austria, May 7–13, 2006, pp. 404–417.

[35] J.-H. Lee, "A New Solution for Projective Reconstruction Based on Coupled Line Cameras," *ETRI J.*, vol. 35, no. 5, Oct. 2013, pp. 939–942.

[36] A. Lee, J.-H. Lee, and J. Kim. "[POSTER] Movable Spatial AR on-the-Go," *IEEE Int. Symp. Mixed Augmented Reality*, Fukuoka, Japan, Sept. 29 – Oct. 3, 2015, pp. 182–183.

[37] M.-K. Park et al., "Spatial Augmented Reality for Product Appearance Design Evaluation," *J. Computational Des. Eng.*, vol. 2, no. 1, Jan. 2015, pp. 38–46.

[38] H.-S. Park et al., "In-vehicle AR-HUD System to Provide Driving-Safety Information," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 1038–1047.

[39] H. Ha and K. Ko, "A Method for Image-Based Shadow Interaction with Virtual Objects," *J. Computational Des. Eng.*, vol. 2, no. 1, Jan. 2015, pp. 26–37.

Ahyun Lee received his MS degree in digital imaging from Chung-Ang University, Seoul, Rep. of Korea. From 2011 to 2012, he was an engineer at LG electronics Inc. Pyeongtaek, Rep. of Korea. He is currently working toward his PhD degree in computer software at the University of Science and Technology, Daejeon, Rep. of Korea. His research interests include computer vision, augmented reality, and robotics.

Joo-Haeng Lee received his BS, MS, and PhD degrees in computer science from Pohang University of Science and Technology, Rep. of Korea in 1994, 1996, and 1999, respectively. He joined ETRI in 1999 and is a principal research scientist with the Human–Robot Interaction Lab. He has also been teaching graduate students (as an associate professor) at the University of Science and Technology, Daejeon Rep. of Korea, since 2008. Based on geometric modeling and processing, his research interests include computer graphics, HCI, computer vision, robotics, and their convergence.

Jaehong Kim received his PhD degree in computer engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2006. He has been a research scientist at ETRI since 2001. His research interests include socially assistive robotics for the elderly, human–robot interaction, and social HRI frameworks.

http://dx.doi.org/10.4218/etrij.16.0115.0610