

Regular Paper

Guidance and Visualization of Optimized Packing Solutions

NATTAON TECHASARNTIKUL^{1,a)} PHOTCHARA RATSAMEE² JASON ORLOSKY²
TOMOHIRO MASHITA² YUKI URANISHI² KIYOSHI KIYOKAWA³ HARUO TAKEMURA²

Received: June 7, 2019, Accepted: November 29, 2019

Abstract: Packing optimization is a challenging and time-consuming task for a number of industry and logistics applications. Efficient packing can reduce the cost of storage and shipping and also guarantee that damage will not occur during shipping. To help address this problem, we propose a spatial augmented reality-based support system for assisting workers with packing optimization. Our packing support system first uses an RGB-D camera to acquire color and depth information of the items to be packed and the destination container. Then, object segmentation and dimension estimation are simultaneously carried out, and the position and orientation of packing items inside the container are calculated using a bin-packing algorithm. Finally, the optimized packing instructions are projected onto the user's work area. We then developed and tested two user interfaces (UI) for visualizing instructions called *Rotation* and *Object Movement*. Experimental results showed that both methods help reduce packing time up to 57.89% in *Rotation* and 55.63% in *Object Movement*, compared to a non-UI method.

Keywords: Spatial Augmented Reality, Packing Support System

1. Introduction

Packing, defined simply for the purposes of this paper as the process of arranging smaller items into a larger container, is both a complex and prevalent issue in logistics management due to the fact that it is an NP-hard problem [8]. Ensuring proper packing solutions allows workers, and hence the company, to optimize shipping and transportation costs [1]. In mathematics, many heuristic algorithms [9], [11], [14] have been developed to solve for the optimal solution of packing problems. These algorithms have sometimes been applied for large-scale packing operations, for example when loading thousands of containers onto ships or airplanes. For smaller scale packing problems such as those faced by consumer-oriented delivery or local shipping carriers [16], fewer methods are currently available for logistics and optimization. Moreover, methods for real-time presentation or display of instructions have not been applied or tested.

Overlaying a task solution visualization onto the workspace helps reduce both completion time and the number of errors [19], [23]. Such a real-time support or worker training system would be useful to individual workers and scalable across the entire supply chain. As a potential solution, we have developed a real-time packing guidance system using Spatial Augmented Reality (SAR), a branch of Augmented Reality (AR) that involves overlaying computer-generated information onto the real world using

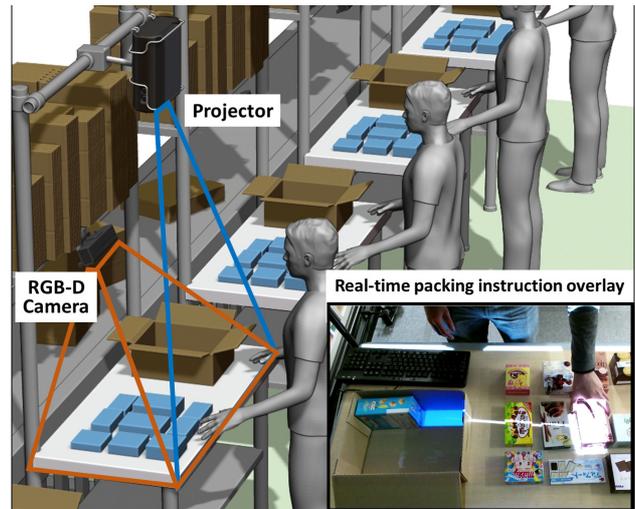


Fig. 1 Model of the use case for our packing support system in a typical fulfillment center (top). Projection onto a sample workspace with virtual overlays provides real time instructions to guide users to the optimal packing solution (lower right).

projection techniques [7]. The image in **Fig. 1** provides an example of the system's functionality and operation for assisting a warehouse shipping task.

The hardware for the system primarily consists of two parts, the first being an RGB-D camera that can ascertain and segment packing items. This camera is coupled with a projector that projects packing guidance onto the user's workspace. The software includes several primary methods to facilitate support. First, color and depth information of the items to be packed and a target container are acquired using the RGB-D camera. Next, object segmentation and dimension estimation are computed using the

¹ Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565-0871, Japan

² Cybermedia Center, Osaka University, Ibaraki, Osaka 567-0047, Japan

³ Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

^{a)} nattaon@lab.ime.cmc.osaka-u.ac.jp

segmented point cloud, and finally the optimal packing solution is found using the bin-packing algorithm proposed by Baltacioglu et al. [5]. We also designed two packing overlays (virtual instructions) called Rotation Instruction (RI) and Movement Instruction (MI). These overlay techniques are then combined with the packing optimization algorithm into a single real-time system. When the user begins packing, step-by-step visualizations of package placements and orientations are projected onto the workspace.

To test how well these visualizations could assist users, we conducted experiments to evaluate the performance of our system by comparing two overlay methods and the unsupported packing process. Results showed that overlays reduced packing time and the number of moves required per object versus a baseline, but that no single type of instruction significantly outperformed any other.

Our contributions in this paper include:

- the hardware setup and implementation of our packing support system as a whole,
- algorithms that obtain and segment objects and output an optimized packing solution in real time, and
- the design and evaluation of two types of virtual overlays that guide the user during the packing process.

2. Related Work

2.1 The Packing Problem

In mathematics and computing, the packing problem is typically described as the general goal of packing smaller items into a larger container, and the problem can often be multi-dimensional. For example, a simple one-dimensional packing problem is disk partitioning on traditional computing systems [18]. Two-dimensional packing problems would include tasks such as efficient cutting of wood or metal plates [22]. Three-dimensional packing problems are most typically found in logistics, and the goal is almost always to reduce the cost of packaging materials, vehicles, and fuel, hence the overall cost of transportation. This class of problems is typically categorized by input minimization and output maximization [34] based on the type of logistics problem being solved. Input minimization is an approach that takes into account the packing all of the items using the smallest number of containers possible (as e.g., bin-packing), whereas output maximization seeks to pack the largest possible subset of items into a single container (e.g., the knapsack problem).

Packing algorithms are usually executed in two steps, including container selection and packing position determination. A sorting step is sometimes included since the ordering can potentially increase packing efficiency [9]. In the case where more than one container is used, a method such as branch and bound [25] or heuristics such as first-fit-decreasing or best-fit-decreasing [9] are often used for selecting a container. In order to decide an item's position inside a container, placement heuristics such as Wall-building [14], Layer-building [5], Corner points [25], and Extreme points [9] have been developed in the past.

Though wall-building and layer-building approaches pack items using a "guillotine" partitioning method, this does not necessarily utilize the entire volume of the container. Both the cor-

ner point and extreme point methods are generally able to utilize a container space more effectively, but have a non-rotation constraint. Therefore, for cases with the same input, the wall building and layer building methods with a rotation constraint would provide a more optimal packing solution. Because packing problems are inherently NP-hard, none of the existing methods are always guaranteed to result in a perfect solution. At the same time, all of them have been applied to real packing problems by constraining the problem to the requirements for the particular application.

Most of the packing solutions are typically provided in a format such as a text file [9], [25], which normally contains a percentage of bin utilization and details of each packing item that are listed by their dimensions, position, orientation, and container number. By providing these data, it is easy to compare efficiency between different packing methods by listing them in a table [34]. However, in practical use, it is hard for humans to comprehend what a particular packing solution looks like or how it was ordered or packed. Erick et al. introduced a method using OpenGL to draw three-dimensional computer graphics for simulating a packing solution [11]. Observing a graphical simulation not only allows a user to better identify free space in a container or how to pack the item, but also helps developers and researchers check their packing algorithm's precision and efficiency. Moreover, with the visual and interactive capabilities of a simulated model, the user is able to understand and learn a more disciplined approach to solve 3D packing problems.

Nowadays, 3D simulation of packing solution services are available in both non-commercial [28] and commercial [21], [27]. However, existing systems display their solutions on a flat monitor. Thus, the user needs to switch his or her attention between the screen and the workspace, which can be time-consuming and prevent spatial learning. Alternatively, displaying the solution onto the workspace can potentially be more efficient for both time, effort, and learning of the packing task.

2.2 Information Visualization

Augmented Reality (AR) is a technology that displays virtual information onto the real world environment [20]. A variety of display devices are used in AR to display visual information, such as a hand-held display, a head-mounted display (HMD), or a projector. By using a smartphone or a tablet device with a built-in camera, people can access richer information from the real environment through AR applications such as real-time signboard translation [13] or a touring guide [33]. In the case of an HMD, the user does not need to hold the device so that one can get guidance information and work at the same time. For example, a user can assemble objects while viewing the assembly solution through an HMD [19], machine maintenance [10], or manipulate objects while seeing an instruction [4].

Although the user is able to work hands-free by wearing the HMD, the field of view is limited and may not cover the whole work area, causing potential issues like eye fatigue or unnecessary head movement. These issues can often be mitigated by using projection-based AR, or Spatial Augmented Reality (SAR), which has a great potential to increase efficiency by allowing people to view the entire work space efficiently without wearing any

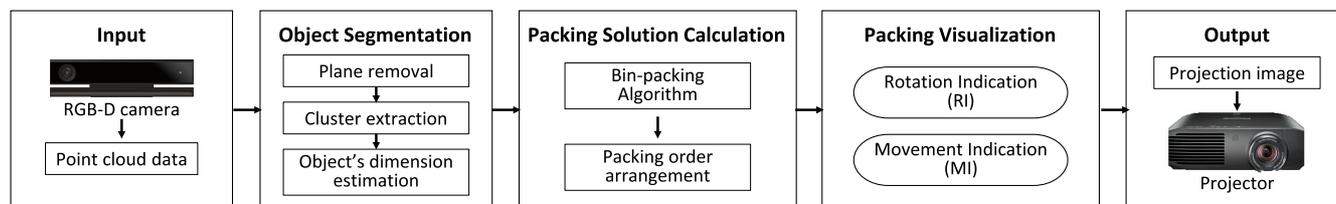


Fig. 2 Flowchart of each step of the system from raw input to final output.

device. One example includes interior architecture design systems [24]. Projecting a mock cabinet layout onto a potential room space can enable designers to modify the layout to fit the room environment or a potential client's preferences in real-time, and also ensure that drawers or doors do not interfere with normal operation of other objects. In the case of manufacturing, SAR has been applied in a spot welding inspection support system [35]. Projections of a target position for inspection and a path to the next position help a worker avoid misses and prevent duplication of work for welding position inspections.

Another advantage of SAR is the ability to indicate or guide a remote user. For example, Tsimeris [30] developed a number of SAR visual cues that convey translation and rotation information to instruct object arrangement. Adcock et al. [2] used semantic information of object physical properties, by drawing a shape of the manipulating object at the work area. Uranishi et al. [31] used a grid-pattern to help both instructor and user and to indicate and identify the position of the object. However, these systems generally use translational or 2D rotation cues, and require manual operation on the part of an instructor or professional user. This motivated us not only to propose 3D rotation cues for manipulating objects using SAR, but also to automate the process of optimizing a packing solution.

3. System Design

Our projection-based packing support system is designed to help users pack items into a container as densely and quickly as possible without the assistance of a remote collaborator. A flowchart outlining the entire process flow is depicted in Fig. 2. The system first starts by using the overhead RGB-D camera to acquire workspace and object information as inputs. We then use an object segmentation algorithm to estimate each object's dimensions and calculate a packing solution. Finally, the packing solution is displayed to the user in a step-by-step fashion by projecting the instructions (visualizations) directly onto the work area and updating after every completed step.

When testing our prototype system, we selected items and a container that were rectangular solids, which are often found in a number of real packing scenarios because they can be used to minimize open space in a container in comparison with other shapes. During experimentation, all items were initially placed orthogonally to the table and with enough space between them to facilitate segmentation. We did this so that the scanning process only has to happen once, as opposed to requiring the user to scan each item individually.

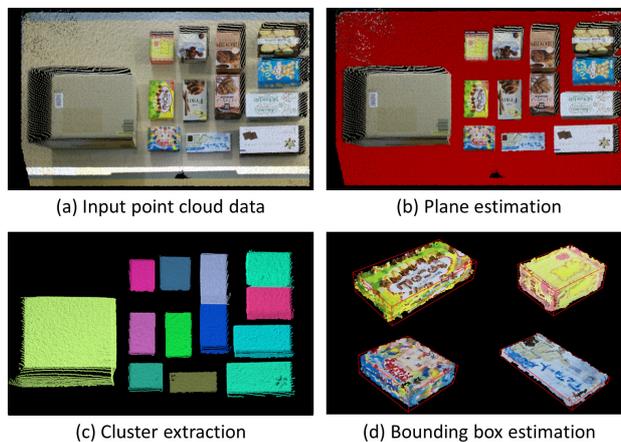


Fig. 3 The object segmentation process. Showing the capture of point cloud data, estimation of the workspace plane, segmentation of the container and packing items, and bounding box estimation with overlaid textures.

3.1 Object Segmentation

In this step of the process, we extract information about each packing item from an input point cloud image, divide these into individual point cloud segments, and calculate dimensions through the Point Cloud Library [26].

We first estimate the table surface from the input point cloud, as shown in Fig. 3 (a), by using the Random Sample Consensus (RANSAC) algorithm [12]. RANSAC is an iterative method that estimates parameters of a mathematical model from the data, which in our case is used to fit all points from the point cloud into possible plane models. After a number of iterations, it returns the plane model of best fit as well as the points fitted to that model, referred to as inliers. A sample result of the planar model from the RANSAC algorithm is shown in Fig. 3 (b), where red points are the group of inliers included in the plane model and all other points are outliers. The estimated plane parameters a , b , c , and d of this inliers group in $ax + by + cz + d = 0$ form are -0.035 , -0.377 , 0.925 , and -0.8263 , respectively.

We then remove the plane inliers (table) from the point cloud so that only objects that align on table remain. This allows us to use the Kd tree search method [6] to find clusters of point cloud data that represent individual objects. The results of a sample cluster extraction are shown in Fig. 3 (c), where different groups of clusters are shown in different colors. In each cluster, the biggest point cloud segment is selected as the container. Other remaining clusters are labeled as packable items, and we store each item and container cluster with a reference (x, y, z) position relative to the workspace. Finally, we assign the smallest bounding box possible to each cluster and set each object's dimensions equal to its bounding box size, as shown in Fig. 3 (d). In order to correctly

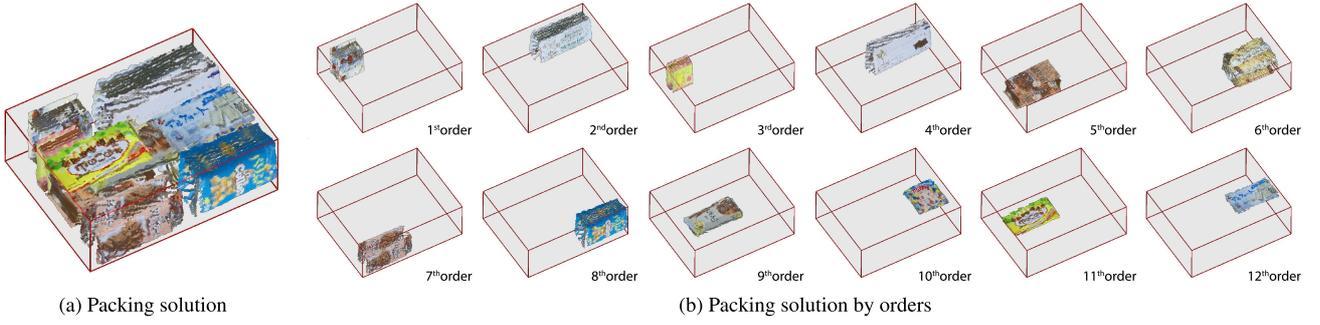


Fig. 4 Simulated visualizations of a packing solution showing segmented point cloud data fit into the container (left) and the sequential packing solution at each step (right), with the red bounding box signifying the outer regions of the container.

orient and display visualizations, we also extract Eigen values and vectors for each object, and use them to rotate the object to align the axis in the target container. Note that in the experiments, we found 1–2 mm of error in the estimation of each bounding box dimensions from the real object, therefore we adjusted the dimensions manually to more accurately represent the items being tested.

3.2 Packing Solution Calculation

To find the densest packing solution for a particular container, we apply the bin-packing algorithm by Baltacioglu et al. [5], which is a heuristic that employs a layer packing approach and packs rectangular boxes in any orthogonal orientation. We input the x , y , and z dimensions of the container, total number of items, and lists of the item's dimensions into the algorithm. Outputs from the algorithm include the x , y , and z positions and dimensions of each item to be packed in the container. A visual representation of one of these packing solutions along with each packing step is shown in **Fig. 4**.

The process behind the packing algorithm is described as follows. First, we define n as the total number of items and $\{d_{i1}, d_{i2}, d_{i3}\}$ as the three dimensions of a particular item i . We then define DIM as the set of unique dimensions for all items as calculated by the following formula:

$$DIM = \cup_{i=1}^n \cup_{j=1}^3 d_{ij} \quad (1)$$

The variable m represents the total number of members in DIM , and k is the index of members ranging from 1 to m . We then define VAL as a set of the average free space values for each DIM_k . The variable c_i is set to the dimension from item i that is closest to the DIM value, and the value of c_i is from $\{d_{i1}, d_{i2}, d_{i3}\}$. Each member of VAL is calculated by using the following formula:

$$VAL_k = \sum_{i=1}^n |DIM_k - c_i| \quad (2)$$

We then construct, $LAYER()$, an array of candidate layers height using the following formula:

$$LAYER() = \cup_{k=1}^m (DIM_k, VAL_k) \quad (3)$$

Afterwards, $LAYER()$ members are arranged by increasing the value of VAL . A small VAL value represents lower remaining free space between packing item and layer height DIM . We use the

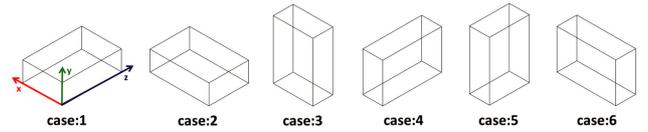


Fig. 5 The six possible rotations for any packing item, which determine how the algorithm handles the instruction at a particular step.

first DIM of candidate layers that has the smallest VAL as the first layer height. The algorithm then repeatedly tries packing solutions by selecting items that can fit into the layer along X dimension and repeats placement operations into the current layer along Z dimension until the layer surface is full. When no space remains to pack items into the current layer, a new upper layer along Y dimension is created, and the process is continued for the current layer. This bin-packing algorithm also tries to pack items with six different orthogonal rotations of the container as depicted in **Fig. 5**, then compares each rotated container utilization and selects the container rotation that has the best packing utilization as the optimal solution.

After obtaining the optimal packing solution, we arrange the order of packing items as determined by the algorithm. Simply put, we follow the rules stating that lower item positions should be packed before upper item positions and that packing order should start from one side to opposite side of the container.

Now that we could automatically obtain an optimal packing solution, we needed an intuitive way to present this to the user. In order to effectively convey the item's destination packing position and orientation, packing instructions should be both easy to understand and at the same time indicate sequential progress for the item to pick up. We then went on to design two different types of virtual overlays to convey information, as described in the following section.

4. Virtual Instructions and Overlay

As a user progresses with the packing task, steps toward reaching the optimal solution need to be displayed item by item, in real time, from start to finish. Our previous work [29] uses point cloud images that were captured during object segmentation process to visualize a packing solution. However, this type of interface has many demerits. Due to its pixel-based image, the projected image is hard to see, especially when it is overlaid on a textured item. Moreover, if the packing solution shows the rotated point cloud image, the user will see a sparse point cloud because it was only

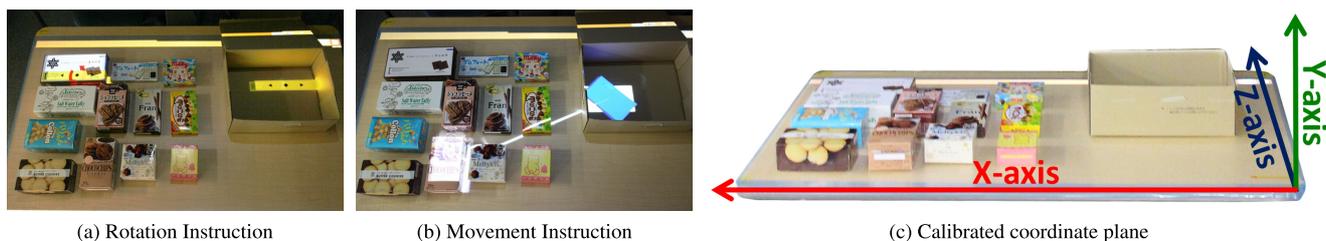


Fig. 6 Images of the packing visualizations in-use, showing the two overlays we designed (left and center) and a calibrated coordinate plane system for reference (right).

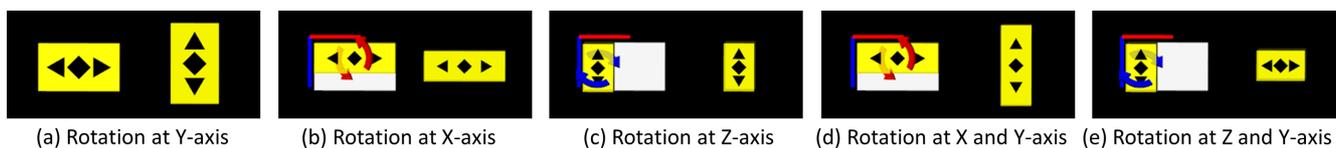


Fig. 7 Rotation Instructions: in each sub-image, the yellow box with blue/red target rotation markers on the left is projected over the physical object's original position, and the right image is projected in the destination container at the target orientation and position.

captured from a single view of the RGB-D camera.

In this study, we proposed two user interfaces, called Rotation Instruction (RI) and Movement Instruction (MI). Representative images of each of these techniques are shown in **Fig. 6** (a) and 6 (b), respectively. These methods are interactive that the user can advance or return via keyboard.

Prior to rendering, we obtain item and container positions and dimensions to locate the physical positions on the user's workspace. Container positions and dimensions in the real world are then used as reference positions for the final packing solution obtained from packing solution calculation described previously. We use the coordinate system depicted in **Fig. 6** (c) to project item positions and orientations, where the positive X -axis is a horizontal line aligned with the workspace from right to left. The positive Y -axis is a vertical line passing through the center of the workspace upwards into the air. The positive Z -axis is a horizontal line through the center of the workspace pointing away from the user.

4.1 Rotation Instruction

In some situations, packing items may have similar faces and dimensions, so it can be hard to distinguish differences in orientation simply by observing images from the virtual overlays. For example, consider a plain, square box of a single color in which one side is only slightly longer than the other. By only looking at the item image itself, users may feel uncertain as to which orientation is actually being indicated. To solve this potential ambiguity, we designed the Rotation instruction, which uses graphical images to help identify the entire transformation of the packing item. For any given packing task, there are six possible orthogonal rotations in three dimensions of any object, which are shown in **Fig. 5**. Much like algorithmic solutions to Rubik's cubes are divided into specific rotations, We classify these six possible orthogonal rotations into two-dimension rotations between the X -, Y -, and Z -axes as follows:

- case:1 no rotation
- case:2 rotate 90 degrees at Y -axis

- case:3 rotate 90 degrees at X -axis
- case:4 rotate 90 degrees at Z -axis
- case:5 rotate 90 degrees at X -axis and Y -axis
- case:6 rotate 90 degrees at Z -axis and Y -axis

In other words, a combination of rotations on any two axes can cover six different possible target orientations for any packing object.

The visualization of RI technique consists of 2 steps of rotation. The first step uses a red or blue arrow to indicates a rotation along X -axis (to flip an item up/down) or Z -axis (to flip an item left/right). The second step uses 2 comparable images to indicates a rotation along Y -axis. If there is a rotation only at Y -axis, the instruction will show only the second step image as shown in **Fig. 7** (a).

The first rotation step is images that show on the left side. A white rectangle indicates an initial orientation of an item. A yellow rectangle (left side) which overlays on the white rectangle position that indicates a final orientation of the item, a red and blue line at the corner of the rectangles show X and Z -axis direction, and a curved arrow indicates which direction to rotate the item. The left sides of (b) and (d) in **Fig. 7** show a red curved arrow that guides the user to rotate the item 90 degrees around the red line or X -axis, from the white rectangle position to the yellow rectangle position. The left sides of (c) and (e) in **Fig. 7**, which have a blue curved arrow, prompt the user to rotate the item 90 degrees around the blue line or Z -axis.

The second step indicates a rotation along Y -axis, which shows two yellow rectangles on the left and right side of the image together with the first step images. Both yellow rectangles have black arrows in the middle to signify an orientation. The left side of yellow rectangle shows an orientation before rotating the second step and the yellow rectangle on the right side shows the final orientation of the item. **Figure 7** (a), (d) and (e) which have 90 degrees different orientation of the yellow rectangles between the left and right side, prompt the user to turn the item clockwise or counter-clockwise (90 degrees rotation around Y -axis).

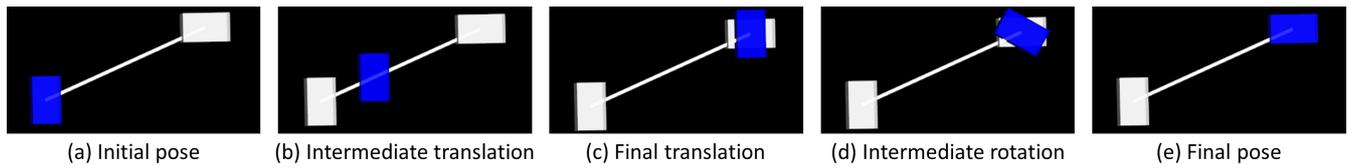


Fig. 8 Object Movement Instructions: (a), (b), and (c) show item in an image is moving from a current position to a target position. (c), (d), and (e) show item in an image is rotating from a current orientation to a target orientation.

4.2 Movement Instruction

In the RI technique, the user needs to infer intermediary steps between the start and finish, which lacks motion cues. We developed one final technique called Movement technique that in contrast shows smooth movement of the item from its current position to the target position. For this overlay, we hypothesized that users would be able to follow the instructions without additional cognitive effort to interpret translations or rotations. Frames from the sequence of one of these moving objects is shown in **Fig. 8**.

White rectangles show the current object's initial and final packing positions, and a blue rectangle shows the object's movement and rotation. This type of instruction first shows the object moving along a white leader line to the packing position. After the virtual object reaches the target position, it then gradually rotates 15 degrees along each orthogonal axis all the way to the correct orientation. After observing this movement, the user can replicate the same translation and orientation into the target location.

The blue rectangle will start moving from the initial position after 0.5 second of an input key pressed by the user. Every 0.1 second it will update its position which is 5 cm away from its last position along the white leader line. When it reaches the final packing position, it will start rotating 15 degrees every 0.1 seconds from Y-axis following with the X- or Z-axis. One-axis rotations take 0.6 seconds and two-axis rotations take 1.2 seconds. On average, items are 50 cm away from the container, so the movement overlay takes an average time of around 2 seconds to show the packing cues for one item.

Finally we wanted to conduct a user evaluation to determine the efficiency, trade-offs, and subjective ratings of each technique.

5. Evaluation

To evaluate the effectiveness of real time support and compare our two visualizations, we conducted a series of experiments to test packing performance. Our goal was to assess the overall performance and subjective preferences of the two types of proposed instructions and to compare the results with unassisted packing.

5.1 Experiment setup

To select compact packing tasks, we collected various sizes of 50 snack boxes and several parcel boxes. The size of these boxes was measured manually. We then input the size information of each parcel box and all snack boxes into the bin-packing algorithm. The two most compact packing parcels were selected as our experiment tasks (refer as Task:1 and Task:2).

Task:1 had a filling rate of 87.60% and consisted of 12 individual items to be packed into a container with dimensions of



Fig. 9 Image of the packing items at the start of Task:1.



Fig. 10 Image of the packing items at the start of Task:2.

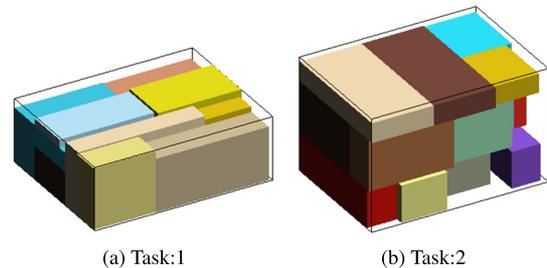


Fig. 11 Simulated images of each optimized packing solution.

32.0 cm \times 11.5 cm \times 25.0 cm, as shown in **Fig. 9**. Task:2 had a filling rate of 84.02% and consisted of 17 individual packing items to be packed into a container with dimensions of 32.5 cm \times 21.0 cm \times 23.0 cm, as shown in **Fig. 10**. A single optimal solution provided by the system for each task is shown in **Fig. 11**, and lists of packing items are shown in **Table 1**.

Our experiment was conducted in two parts: the first was a pilot experiment to create a baseline data of no-assistance packing, and the second was a within subjects design that was used to evaluate the two user interfaces. In the first run, 12 participants (average age of 22.33, ranging from 21 to 24) were asked to complete packing for Task:1 and Task:2 without any assistance. The initial box positions were the same between participants, and the task was complete when all items were packed into the destination container. In the second run, we included 24 participants (average age of 23.67, ranging from 20 to 29) to conduct packing for Task:1 and Task:2 with the RI and MI overlays in random

Table 1 List of packing items.

Task:1		Task:2	
Item	Dimensions (mm)	Item	Dimensions (mm)
1	158 × 98 × 63	1	214 × 70 × 56
2	159 × 96 × 63	2	152 × 70 × 80
3	160 × 60 × 95	3	140 × 65 × 127
4	157 × 60 × 95	4	220 × 68 × 53
5	230 × 114 × 37	5	75 × 69 × 85
6	84 × 106 × 35	6	38 × 65 × 100
7	203 × 97 × 53	7	160 × 95 × 60
8	111 × 111 × 52	8	160 × 95 × 60
9	163 × 25 × 90	9	158 × 95 × 60
10	116 × 35 × 95	10	150 × 95 × 121
11	150 × 30 × 89	11	160 × 90 × 24
12	170 × 15 × 77	12	60 × 95 × 157
		13	49 × 95 × 133
		14	230 × 37 × 114
		15	230 × 37 × 114
		16	116 × 35 × 95
		17	106 × 35 × 84

order per participant.

5.2 Procedure

Before starting the experiment, we explained to each participant how to interpret the RI and MI instructions and how to select the next or previous instruction by pressing the right or left arrow keys. The times used for object segmentation, which automatically determine items' longer and shorter side of tasks 1 and 2, were 67.10 seconds and 82.12 seconds, respectively. The packing solution calculation time was 0.48 seconds and 0.26 seconds for tasks 1 and 2, respectively. To make a more fair comparison, we added these pre-processing times to the packing time for each task. During the experiments, we measured time and move count for each item. After completing the two packing tasks, we asked the participants to fill out a subjective questionnaire.

5.3 Hypotheses

In general, we hypothesized that packing time would be reduced when assisted with the visualizations since trial and error would not be necessary to find a solution. On the other hand, we thought that the total number of moves required would be relatively similar despite taking more time. With respect to visualizations, we thought that the RI technique would be most efficient on shortening the packing time because it provides a 3D visualization of the item rotation, whereas users need to wait for the animation with the MI technique.

Regarding preference, we thought that the MI technique would be most preferred since it provides a packing example closest to the real world objects, but with the other technique the user has to take some effort to interpret the visualization before insertion. Accordingly, we formulated the following hypotheses:

- **H1:** The proposed packing support system and visualizations will result in reduced packing time and number of item moves compared to the no-guidance condition.
- **H2:** The Rotation Instruction (RI) will result in the lowest packing time when compared to other visualizations.
- **H3:** The Movement Instruction (MI) will receive the highest subjective ratings for ease of understanding, usefulness and satisfaction level when compared to other visualizations.

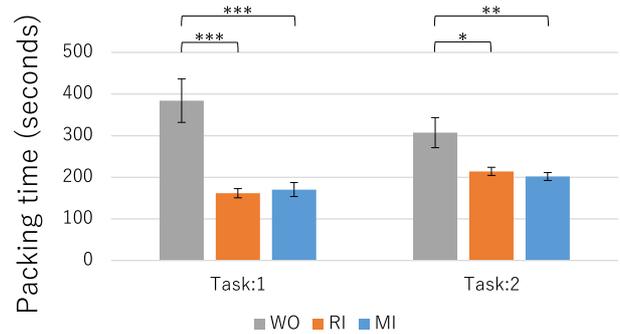


Fig. 12 Average packing time by method (in seconds). WO indicates without instruction, RI indicates with rotation instruction, and MI indicates with movement instruction.

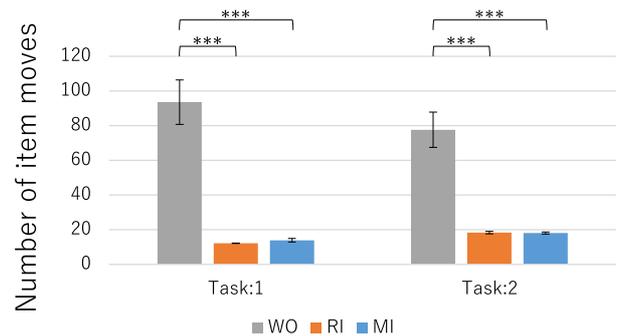


Fig. 13 Average number of item moves by method.

5.4 Experiment results

This section presents a comparison of results for packing without support from the system (WO) and each of the two packing interface techniques (RI and MI). We analyzed packing time and the number of moves per item. We also gathered subjective user data for each type of virtual instruction, which included ease of understanding, instruction usefulness, and satisfaction level.

5.4.1 Packing Time

We combine measured time to completion of each set of instructions with the time used in Object Segmentation and Packing Solution Calculation process. The total times are presented in **Fig. 12**, which shows that our proposed techniques (RI and MI) reduced packing time when compared to the non-assisted (WO) condition. Reductions in time by using RI and MI were 57.89% and 55.63% in Task:1, respectively, and 30.32% and 34.26% in Task:2, respectively.

We conducted a parametric measures Analysis of Variance (ANOVA) test [15], followed with post-hoc analysis using pairwise t-tests with Holm's adjustment [17] to evaluate time to completion results. In Task:1, we found significant differences between the WO, RI, and MI overlays with $F = 15.21$, $p < 0.001$, and in Task:2, we also found significant differences between the WO, RI, and MI overlays with $F = 6.65$, $p = 0.003$. These results are summarized in Fig. 12, where the asterisk ***, **, * symbols between each set of bars indicates a significance level of $p < 0.001$, $p < 0.01$, and $p < 0.05$, respectively.

5.4.2 Number of Item Moves

Afterwards, we analyzed the number of item moves as presented in **Fig. 13**. The chart shows that both RI and MI visualizations significantly reduced the number of total item moves, by 86.91% with RI, and 85.12% with MI in Task:1, and by 76.39%

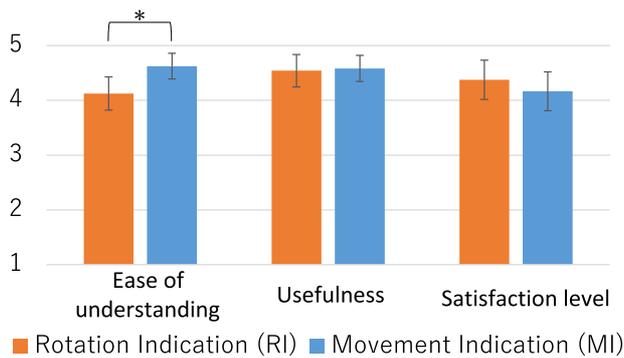


Fig. 14 Subjective ratings from the questionnaire in the main evaluation.

with RI, and 76.71% with MI in Task:2. The same analysis used for packing time was also conducted to evaluate item moves. We found significant differences of $F = 39.19$, $p < 0.001$ for Task:1 and $F = 33.83$, $p < 0.001$ for Task:2.

5.4.3 Questionnaire

Finally, we conducted a questionnaire with a 5-point Likert scale [3] to rate subjective opinions about the interface and overlays. A 1 corresponds to the most negative response and a 5 to the most positive response. Regarding each packing instruction, we asked 3 questions: “1. Is this packing indication easy to understand?”, “2. Is this packing indication useful?”, and “3. How much were you satisfied with this packing indication?”. The summary of ratings are shown in Fig. 14.

We conduct a Wilcoxon signed rank test [32] to evaluate questionnaire results. We found a significant difference only on the ease of understanding ($p = 0.045$), which is indicated by the asterisk * symbol between each set of bars in the graph.

6. Discussion

In the evaluation, we found that packing without support from the system took much longer on average than with the system. This evidence supports our hypothesis (H1), which suggests that our system is overall beneficial for packing support tasks, and can already be used practically.

Although Task:1 (12 items, filling rate 87.60%) had fewer items to pack than Task:2 (17 items, filling rate 84.02%), unassisted packing time for Task:1 was longer than Task:2 because of the higher fill rate. When more compact packing is needed, the packing is more difficult and thus the system would efficiently help reduce both packing time and number of item moves.

On the other hand, no significant results for the comparison of packing times between the two proposed visualization techniques were found. Therefore, the hypothesis (H2) is only partially supported, which suggests that the relative benefits of each individual method are quite similar.

Regarding user preference, even though rating on ease of understanding was significantly higher MI compare with RI, no other significant data was found on the other ratings. Accordingly, the hypothesis (H3) was only partially supported.

During the experiment, some participants took more time than others for the same condition due to mistakes and the time taken to think about how to rotate items. As such, in the RI condition, when looking at the overlay some users did not immediately

know the direction to rotate, so they had to try rotating 2–3 times to make sure the direction was correct. Also in both conditions, some participants did not carefully follow the instructions, so they incorrectly rotated the object. When continuing to pack later on, the remaining items could not fit in the container, so they had to go back several steps to correct the mistake.

We also asked opinions from each participant about general use during the experiment. With the RI instructions, participants took their time with about the first half of the items to understand the method, though with the latter half, participants had already started to become familiar with the instructions. In the MI condition, some participants said that seeing the image of the items moving was fun, while some said the movement was too slow, so they had to wait before they could pack the item. Also, some participants mentioned that the projections did not exactly fit the packing position, so they had to adjust the packing position by themselves. This feedback suggests that one primary improvement we need to focus on is the reduction of errors for the calibration of the projection surface. Sometimes the projected image was slightly distorted when the projection was mapped to a non-planar surface and color information was skewed due to projection on an existing colored/textured surface. This was especially true when objects of different heights were present in the workspace.

6.1 Future Work

To help automatically check for individual insertions and removals before proceeding to the next packing object, individual object tracking may provide a way for the system to enable backtracking and perhaps more intelligent instructions. Though manual interaction gives the user more control to some extent, more adaptive visualizations could help alert the user that their packing was not correct, facilitate training, develop better packing habits, and further reduce mistakes. Future work includes improving the calibration of the surface area, updating the RI instruction, and an object tracking approach rather than mapping point cloud data.

7. Conclusion

In this paper, we propose a new projector based packing support system and three types of visualizations to help convey instructions. The system works by recognizing an external container and packing items in a flat workspace, and then performing object segmentation to extract both the items and container dimensions. Afterward, the system uses all input items to calculate a packing solution, which consists of packing positions and orientations, and the order and placement of each item are projected onto the workspace as a user completes the packing task.

To help evaluate spatial augmented reality (SAR) as an assistive tool, we tested two types of visualizations, including the Rotation and Movement based instructions, and compared these to non-assisted packing. Experimental results showed that the proposed packing techniques significantly reduced average packing time and movement requirements for packing tasks, with average decreases in task time of up to 57.89% with Rotation instructions and 55.63% with Movement instructions. We hope that this research will bring packing support systems one step closer to

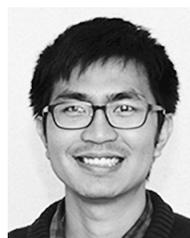
practical use in the transportation industry and pave the way for further iterations of spatial augmented reality research.

References

- [1] Abramov, A. and Kovtaniuk, A.: Decision Support Systems in Transport Logistics (2010).
- [2] Adcock, M., Ranatunga, D., Smith, R. and Thomas, B.H.: Object-based touch manipulation for remote guidance of physical tasks, *Proc. 2nd ACM Symposium on Spatial User Interaction*, pp.113–122, ACM (2014).
- [3] Allen, I.E. and Seaman, C.A.: Likert scales and data analyses, *Quality progress*, Vol.40, No.7, pp.64–65 (2007).
- [4] AutoID, T.: Projekt SmARPro - Verpackung mit Augmented Reality (2016), available from (<https://www.youtube.com/watch?v=GU3IM07MNss>) (accessed 2019-09-25).
- [5] Baltacioglu, E., Moore, J.T. and Hill Jr, R.R.: The distributor's three-dimensional pallet-packing problem: A human intelligence-based heuristic approach, *International Journal of Operational Research*, Vol.1, No.3, pp.249–266 (2006).
- [6] Bentley, J.L.: Multidimensional binary search trees used for associative searching, *Comm. ACM*, Vol.18, No.9, pp.509–517 (1975).
- [7] Bimber, O. and Raskar, R.: *Spatial augmented reality: Merging real and virtual worlds*, AK Peters/CRC Press (2005).
- [8] Bramel, J. and Simchi-Levi, D.: *The logic of logistics: Theory, algorithms, and applications for logistics management*, Springer New York (1997).
- [9] Crainic, T.G., Perboli, G. and Tadei, R.: Extreme point-based heuristics for three-dimensional bin packing, *Inform. Journal on Computing*, Vol.20, No.3, pp.368–384 (2008).
- [10] De Crescenzo, F., Fantini, M., Persiani, F., Di Stefano, L., Azzari, P. and Salti, S.: Augmented reality for aircraft maintenance training and operations support, *IEEE Computer Graphics and Applications*, Vol.31, No.1, pp.96–101 (2010).
- [11] Dube, E., Kanavathy, L.R. and Woodview, P.: Optimizing Three-dimensional Bin Packing Through Simulation (2006).
- [12] Fischler, M.A. and Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, Vol.24, No.6, pp.381–395 (1981).
- [13] Fragoso, V., Gauglitz, S., Zamora, S., Kleban, J. and Turk, M.: TranslatAR: A mobile augmented reality translator, *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pp.497–502, IEEE (2011).
- [14] George, J.A. and Robinson, D.F.: A heuristic for packing boxes into a container, *Computers & Operations Research*, Vol.7, No.3, pp.147–156 (1980).
- [15] Girden, E.R.: *ANOVA: Repeated measures*, No.84, Sage (1992).
- [16] Hartmeier, D.: A novel packing heuristic based on rigid body simulation (2016), available from (<http://www.benzedrine.ch/3D-ODRPP.html>) (accessed 2019-09-25).
- [17] Holm, S.: A simple sequentially rejective multiple test procedure, *Scandinavian Journal of Statistics*, pp.65–70 (1979).
- [18] Karmarkar, N. and Karp, R.M.: An efficient approximation scheme for the one-dimensional bin-packing problem, *23rd Annual Symposium on Foundations of Computer Science, SFCS'08*, pp.312–320, IEEE (1982).
- [19] Khuong, B.M., Kiyokawa, K., Miller, A., LaViola Jr, J.J., Mashita, T. and Takemura, H.: Context-related Visualization Modes of an AR-based Context-Aware Assembly Support System in Object Assembly ((Special Issue) Mixed Reality), *Trans. Virtual Reality Society of Japan*, Vol.19, No.2, pp.195–205 (2014).
- [20] Kipper, G. and Rampolla, J.: *Augmented Reality: An emerging technologies guide to AR*, Elsevier (2012).
- [21] LLC, O.L.: LoadPlanner - Advanced Load Planning and Optimization Software Solution (2004), available from (<http://www.loadplanner.com>) (accessed 2019-09-25).
- [22] Lodi, A., Martello, S. and Monaci, M.: Two-dimensional packing problems: A survey, *European Journal of Operational Research*, Vol.141, No.2, pp.241–252 (2002).
- [23] Marner, M.R., Irlitti, A. and Thomas, B.H.: Improving procedural task performance with augmented reality annotations, *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.39–48, IEEE (2013).
- [24] Marner, M.R. and Thomas, B.H.: Spatial augmented reality user interface techniques for room size modelling tasks, *Proc. 15th Australasian User Interface Conference-Volume 150*, pp.39–45, Australian Computer Society, Inc. (2014).
- [25] Martello, S., Pisinger, D. and Vigo, D.: The three-dimensional bin packing problem, *Operations Research*, Vol.48, No.2, pp.256–267 (2000).
- [26] Rusu, R.B. and Cousins, S.: 3D is here: Point cloud library (pcl), *2011 IEEE International Conference on Robotics and automation (ICRA)*, pp.1–4, IEEE (2011).
- [27] SERVICE, D.: 3D Bin Packing - Be Intelligent with Packing and Money (2012), available from (<http://www.3dbinpacking.com>) (accessed 2019-09-25).
- [28] Sharif, A.: Packit4me API (2014), available from (<http://www.packit4me.com/api>) (accessed 2019-09-25).
- [29] Techarathul, N., Mashita, T., Ratsamee, P., Uranishi, Y., Kiyokawa, K. and Takemura, H.: A Projection-based Packing Support System (in Japanese), *IEICE Technical Report*, Vol.116, No.245, pp.63–67 (2016) (online), available from (<https://ci.nii.ac.jp/naid/40020992271/en/>).
- [30] Tsimeris, J.A.: Visual Cues for the Instructed Arrangement of Physical Objects Using Spatial Augmented Reality (SAR), *University of South Australia* (2010).
- [31] Uranishi, Y., Yamamoto, G., Asghar, Z., Pulli, P., Kato, H. and Oshiro, O.: Work step indication with grid-pattern projection for demented senior people, *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp.4698–4701, IEEE (2013).
- [32] Woolson, R.: Wilcoxon signed-rank test, *Wiley Encyclopedia of Clinical Trials*, pp.1–3 (2007).
- [33] Yovcheva, Z., Buhalis, D. and Gatzidis, C.: Smartphone augmented reality applications for tourism, *E-review of Tourism Research (eTR)*, Vol.10, No.2, pp.63–66 (2012).
- [34] Zhao, X., Bennell, J.A., Bektaş, T. and Dowsland, K.: A comparative review of 3D container loading algorithms, *International Transactions in Operational Research*, Vol.23, No.1–2, pp.287–320 (2016).
- [35] Zhou, J., Lee, I., Thomas, B., Menassa, R., Farrant, A. and Sansome, A.: Applying spatial augmented reality to facilitate in-situ support for automotive spot welding inspection, *Proc. 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pp.195–200, ACM (2011).



Nattaon Techarantikul is a Ph.D. student at the Graduate School of Information and Science, Osaka University. She received her M.S. degree in the same faculty in 2017. Her interests include Augmented Reality, Computer Vision, and 3D User Interfaces.



Photchara Ratsamee received his M.E. and Ph.D. degrees in engineering from the Graduate school of Engineering Science, Osaka University in 2012 and 2015, respectively. Currently, he is an assistant professor at Cybermedia Center, Osaka University. He was a visiting researcher at Vienna University of Technology from 2012 to 2013. His research interests include Robot Vision, Human Robot Interaction, Augmented Reality and Rescue Robot. He is a member of RSJ, VRSJ and IEEE.



Jason Orlosky received a bachelors degree in Computer Engineering from the Georgia Institute of Technology in 2006. He then worked in information technology for three years in the United States before returning to school in 2010 to study Japanese language and literature. In 2011, he studied abroad as a research student at

Osaka University, where he graduated with his Ph.D. in 2016. He has received a fellowship from the Japan Society for the Promotion of Science from 2015–2017, and is currently a specially appointed Assistant Professor at Osaka University and adjunct Faculty at Augusta University. His topics include Augmented Reality, Artificial Intelligence, and Vision Augmentation.



Tomohiro Mashita graduated from Osaka University in 2001 and completed his M.S. and doctoral programs in 2003 and 2006, respectively. He was a post-doctoral fellow at Osaka University from 2006 to 2008. He was a senior research fellow at Graz University of Technology from 2012 to 2013. He is currently an

associate professor at Cybermedia Center, Osaka University. His research interests include Computer Vision and Pattern Recognition. He is a member of the IEICE, IPSJ, and IEEE.



Yuki Uranishi received his M.E. and Ph.D. degrees in engineering from Nara Institute of Science and Technology in 2005 and 2008, respectively. He is an associate professor at Cybermedia Center, Osaka University since 2016. He was a JSPS research fellow from 2008 to 2009.

He was an assistant professor at Nara Institute of Science and Technology from 2009 to 2012, Osaka University from 2012 to 2014 and Kyoto University Hospital from 2014 to 2016. He was a visiting research professor at University of Oulu from 2011 to 2012. His research interests include computer vision, augmented reality and human-computer interaction. He is a member of the IEICE, the ISCIE, the VRSJ, the JSMBE, and the IEEE.



Kiyoshi Kiyokawa received his M.E. and Ph.D. degrees in information systems from Nara Institute of Science and Technology (NAIST) in 1996 and 1998, respectively. He is currently a Professor at Graduate School of Science and Technology, NAIST. He worked for

Communications Research Laboratory from 1999 to 2002. He was a visiting researcher at Human Interface Technology Laboratory at the University of Washington from 2001 to 2002. He was an Associate Professor at Cybermedia Center, Osaka University from 2002 to 2017. His research interests include virtual reality, augmented reality, human augmentation, 3D user interface, and CSCW. He is a member of the IEICE, the VRSJ, the HIS, and the ACM.



Haruo Takemura received his B.E., M.E., and Ph.D. degrees from Osaka University in 1982, 1984, and 1987, respectively. In 1987, he joined Advanced Telecommunication Research Institute, International. In 1994, he joined Nara Institute of Science and Technology, as an associate professor in the Graduate

School of Information Science and Technology. From 1998 to 1999, he was a visiting associate professor at the University of Toronto, Ontario, Canada. He is a Professor at Cybermedia Center, Osaka University since 2001. His research interests include interactive computer graphics, human-computer interaction, and mixed reality. He is a member of the IEICE, the IPSJ, the VRSJ, the HIS, the IEEE, and the ACM.