FabNavi: Support system to assemble physical objects using visual instructions

Koji Tsukada, Keita Watanabe, Daisuke Akatsuka, and Maho Oki

Koji Tsukada and Maho Oki, Future University Hakodate
Hakodate, Japan
{tsuka,okimaho}@acm.org

Keita Watanabe, Meiji University
Tokyo, Japan
watanabe@fms.meiji.ac.jp

Daisuke Akatsuka, Mozilla Japan
Tokyo, Japan
daisuke@mozilla-japan.org

Abstract

In this FAB age, people can easily duplicate physical parts through personal & social FAB technologies. However, assembling these parts is still troublesome especially when creating complicated items. To solve this problem, we propose a new system, *FabNavi*, which helps people to assemble physical objects in remote places using visual instructions. FabNavi mainly consists of a pair of a camera and a projector set above the tabletop. People can easily record the assembly processes by pictures and share these instructions on the web. Then other people can re-assemble them just by following the visual instructions projected on the remote tabletop. The goal of FabNavi is to create a framework for recording and sharing new universal instructions through the FabLab network. We had many workshops and exhibitions using FabNavi to explore its possibilities and limitations in Tokyo, London, and Taipei. In this paper, we report the concept, implementation, experiences and findings in the workshops.

Keywords

Personal Fabrication, assembling, visual instruction, interactive tabletop, web service

1 Background

Recently, the digital fabrication tools (e.g., 3D printers, laser cutters, and CNC routers) become more popular and inexpensive. Moreover, there have been appeared many open facilities that provide these tools (e.g., FabLab, FabCafe, and MakerSpace) and web services that support people to share data for these tools (e.g., Thingiverse[1] and Cubify[2]). Thus, the personal and social fabrication technologies lead the FAB age, in which even common people can engage manufacturing in daily lives. In the FAB Age, a person can easily duplicate physical objects through these fabrication technologies. For example, he/she can download a compatible model data of a LEGO block from Thingiverse and create a real block using a desktop 3D printer. However, assembling these parts is still troublesome. For example, it is still difficult for most people to create complicated objects using the 3D-printed blocks. To solve these problems, we proposed *FabNavi*, a support system to assemble physical objects with visual instructions.

[1] http://www.thingiverse.com/

[2] http://cubify.com/

2 Related Work

There are some web services that support people to assemble physical objects. For example, Instructables[3] provides step by step instructions using texts & compliment images. Niconico Douga[4] is a popular video sharing service in Japan, which has more than 30 million users. Many makers used the service to share instructions to make their original works as videos with compliment texts. However, people often have difficulty to create these contents since these systems provide only few supports for creators.

There are many research projects that support people to create data for digital fabrication tools. For example, Saul et al. proposed a system to support users to create chair parts suited for a laser cutter [9]. Lau et al. proposed a system to support creating complementary data from a single photo [5]. There are also support systems to design beadworks [3], metallophones [10] or garments [11]. Although these systems are useful to create data for specific purposes, most of them did not focus to support assembly processes.

There are also many research projects that proposed interactive tabletop/surface systems, which support various tasks on the tabletop using computers, cameras, and projectors. The earliest project might be the DigitalDesk [12] and a famous product is the Microsoft Surface [1]. There are also many research projects that have various focuses on gesture interactions [8], tangible interactions [7], water-top interactions [4], and kitchen interactions [6]. However, there are few systems that focus on practical supports of personal manufacturing.

Thus, there are many web services for sharing assembly processes, research on support systems to create fabrication data, and research on tabletop interactions. Our approach is unique to integrate these approaches for practical direction: supporting people to assemble objects created by digital fabrication tools using tabletop systems.

3 FabNavi

FabNavi is a tabletop system integrated with a web service to help people assemble physical objects in remote places using visual instructions. The FabNavi hardware mainly consists of a pair of a camera and a projector set above the tabletop (Figure 1). People can easily record the assembly processes with pictures and share these visual instructions on the web. Then other people can re-assemble them just by following the visual instructions projected on the remote tabletop (Figure 2).

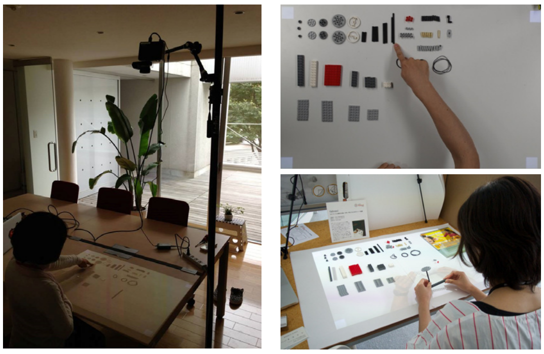

Figure 1: The basic concepts of the FabNavi: a pair of camera & projector can record assembly processes with pictures and project them on the tabletop to assist re-assembly processes.

[3] http://www.instructables.com

[4] http://www.nicovideo.jp

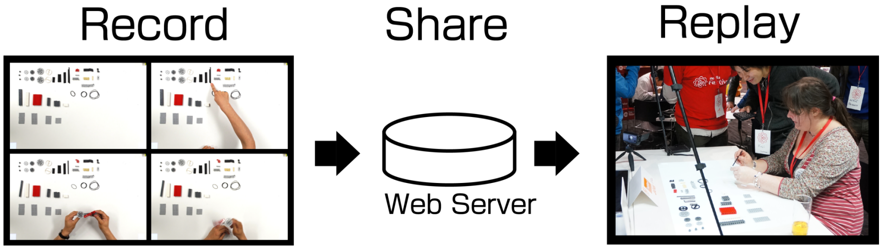

Figure 2: The basic flows of the FabNavi: the sets of recorded pictures, *visual instructions*, are shared on the web, and then other people in remote places can re-assemble it just by following the instruction projected on the tabletop.

We think the FabNavi has potential merits compared to existing services to share assembly instructions: (1) language-independent and (2) capture-skill-independent. First, since most instructions of exiting services consist of texts and compliment images/videos, common people often have difficulty to read/write them in foreign languages. The FabNavi might reduce this problem since it mainly treats visual instructions. Second, the quality of the contents in exiting services is often changed based on camera/computer skills of users. The FabNavi might reduce this problem since it offers fixed conditions of capturing/projecting environments. Thus, we believe our approach is possibly suited for universal use.

4 Implementation

In this section, we explain the implementation of the FabNavi in two aspects: hardware and software.

4.1 Hardware

The FabNavi hardware consists of a pair of camera & projector set above the tabletop, an input device for the trigger mechanism, a client computer to control the above devices, and a server to store/share data on the web (Figure 3).

We selected VIVITEK QUMI Q5 as the projector since it is quite lightweight (490 g) and has adequate resolutions (720p) and brightness (500 lumen).

We selected Sony NEX-5R and DSC-QX10 as the camera. Both cameras equip high-resolution image sensors and high-quality zoom lenses, and can be controlled by Open APIs through Wi-Fi network. First we used the NEX-5R, then replaced it with the QX10 when it became on sale since QX10 was more compact and inexpensive, and had more flexible APIs (e.g., zoom in/out became supported). We did not select web cameras, since they did not have enough image quality and adequate angle of views.

We considered several options of input devices for trigger mechanisms: keyboards / mice / gestures / voices. We selected a cordless number pad by Logitech as a start point since it can provide simple & stable operations and can be set on any positions of the table. We did not select gestures or voices in this stage since we need to fix the basic functions of the system before considering the interaction possibilities.

As the client computer, both Windows and Mac are supported since the software mainly works on the server as a web-based system. The client computer requires 2 network adapters: a Wi-Fi adapter is used to communicate with the camera, and another Wi-Fi or Ethernet adapter is used to communicate with the server.

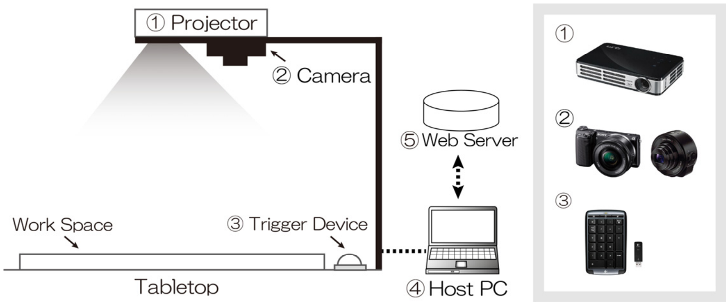

Figure 3. The system architecture of the FabNavi.

After we started the project, we have developed four prototypes that have different structures to fix the camera and projector: (1) auto-pole, (2) tripod, and (3-4) original frames to set on the table or the ceiling (Figure 4).

The prototype (1) uses Manfrotto 032B Single Autopole, Manfrotto 244RC Magic Arm, and our original mount (described later). Although it requires only small spaces, it cannot be set in the room with high ceilings.

The prototype (2) uses Manfrotto MKCOMPACTACN-BK and our original mount. Although it can be carried and set on a table easily, it requires space much larger than the projection area and it is difficult to adjust the height of the camera & projector.

The prototype (3) is designed by ourselves with MISUMI aluminum frames to solve the limitations of prototype (1) and (2): we can easily assemble/disassemble each frame with volts and screws, the system can be set on the small space (almost same as the projection area) and the height of the camera and projector can be easily adjusted.

The prototype (4) is basically a part of the prototype (3). We customize it to set on the rail of the ceiling.

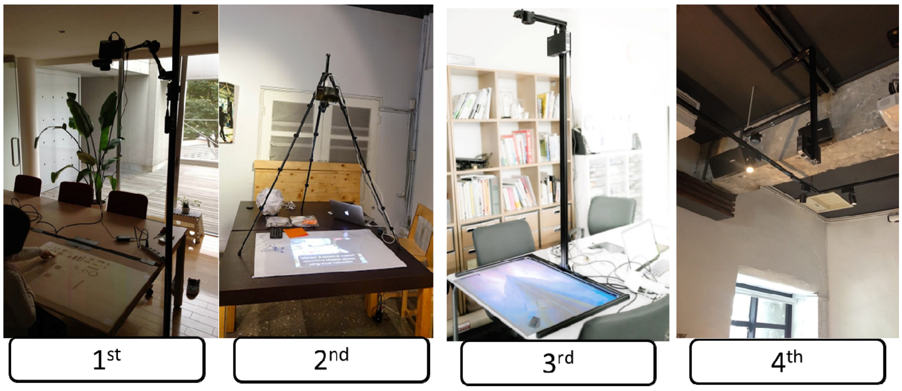

Figure 4: Hardware prototypes of the FabNavi.

The relative positions of the camera and projectors should be fixed to keep the mappings of the projection area and captured area. To archive this purpose, we developed the original mounts to fix the devices: the first mount was developed for the prototype (1) and (2), and the second mount was developed for the prototype (3) and (4) (Figure 5). The first mount was cut from an acrylic board using a laser cutter. The board equips holes to fix the camera, projector and a camera platform of the auto-pole/tripod using nuts and screws. The second mount is designed to fix the camera and projector to the MISUMI frames with nuts and screws. They were also built from an acrylic board.

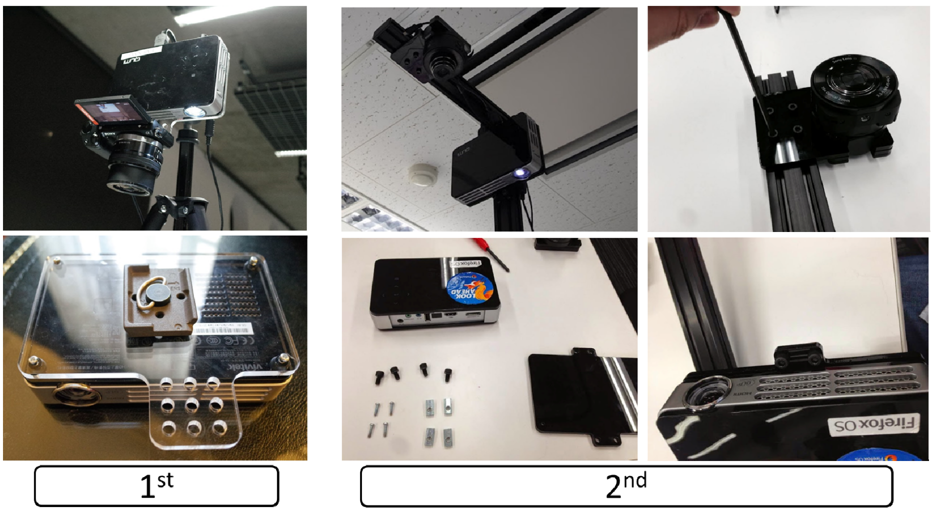

Figure 5: The original mounts to fix the camera and projector.

4.2 Software

Next, we explain the software architecture of the FabNavi. The system is basically designed as a web service, which consists of (1) server program, (2) client program, and (3) plugin of a web browser.

The server program renders pictures and navigation interfaces using HTML and communicates with client program using AJAX[5]. The program is written in Ruby and developed on the Padrino[6] framework. The program runs on a service platform called Heroku[7] .

The client program is developed using JavaScript, which runs on a web browser of a client computer. The client program can take pictures in JPEG format using the browser plugin (described latter) and transmit these pictures to the server through AJAX.

The browser plugin controls the camera and receive pictures through Wi-Fi network. The plugin was developed as a browser add-on of the Mozilla FireFox. The plugin waits for the load of the client program and then pass its reference to the client. Thus, the client program has the access to control the camera through Wi-Fi network. We explain the detailed procedures as follows:

1. The plugin sends the “startRecMode" command to the camera to prepare taking pictures.
2. When the client program detects a trigger input, the plugin sends the "actTakePicture" command to take a picture.
3. The camera returns the URL of the taken picture as a HTTP Response. The plugin then sends the URL to the client program.
4. The client program uploads the picture to the server. The server program then saves the picture to the database.

5 Usage

This section explains usage of the FabNavi in three processes: (1) capture process, (2) replay process, and (3) initialization process. We define a user who captures new instructions as “creator”, and a user who re-assembles objects by following instructions as “assembler”.

5.1 Capture Process

The capture process supports users to capture the assembly processes of their works as visual instructions. In this process, the system also supports a wireless full-keyboard along with the key pad in some operations. The detailed procedures are shown as follows (Figure 6):

1. First, a creator starts a new project by selecting the “create a new project” icon.
2. The creator enters a project name by the full-keyboard. This process can be skipped and he/she can modify the project name later on a common web browser.
3. The creator performs a calibration process if required (described later).
4. The creator can capture assembly processes just by pressing the “enter” key. The captured pictures are immediately shown on the tabletop for the validation. There are no restrictions to take pictures; however, we recommend the creator to capture 3-step instructions: (1) layout all parts, (2) select parts used in the next step, and (3) assemble the parts (Figure 7).
5. The creator can finish the project just by pressing the “esc” or “7” keys. Then the system asks him/her to confirm the project name and privacy setting of the project. If the privacy setting is changed to “public”, the project will be shown in the top page of the service.

[5] Asynchronous JavaScript + XML.

[6] http://www.padrinorb.com/

[7] http://heroku.com/

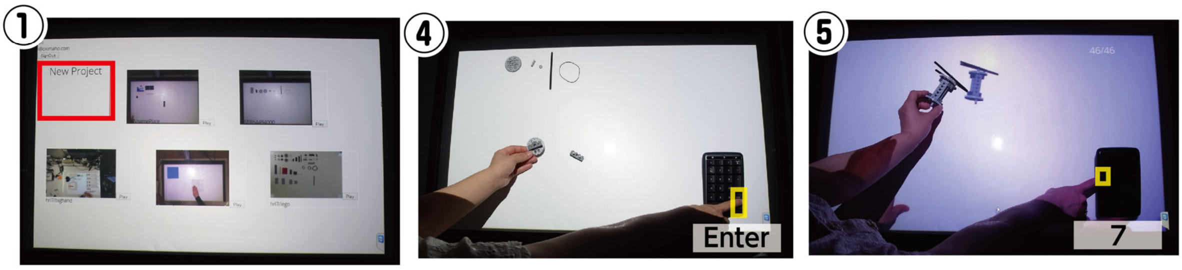

Figure 6: Usages of the capture process. The numbers indicate the steps described in body texts.

Figure 7: Recommended procedures of visual instructions.

5.2 Replay Process

The replay process supports users to assemble projects by following visual instructions. The detailed procedures are shown as follows:

1. First, an assembler selects a project from the project lists shown on the top page of the service.
2. The assembler performs a calibration process if required (described later).
3. The system then presents the first picture. The assembler starts to assemble the project by following the instructions. He/she can move forward/back the instructions just by pressing the "1" and "3" keys.
4. The assembler can finish the project just by pressing the "7" key.

5.3 Calibration Process

The system requires a user to perform a calibration process in several situations: creating/replaying a project first time and replaying a project created at remote places first time. Since the calibration setting is saved on the client browser, the user basically needs the process first time. There are three variables that should be calibrated: position, size, and aspect ratio. We describe the detailed procedures as follows (Figure 8):

1. The system shows a long crossline on the table. The user sets a calibration sheet as its center matches the crossline.
2. The system asks the user to capture the calibration sheet. The picture of the sheet is then projected on the table.
3. The user adjusts the position of the projected sheet as the center of the projected/real sheets match each other. [position calibration]
4. The user adjusts size of the projected sheet, and matches the vertical positions of four small cross lines of projected/real sheets. [size calibration]
5. The user adjusts horizontal size of the projected sheet, and matches the horizontal positions of four small cross lines of the projected/real sheets. [aspect ratio calibration]

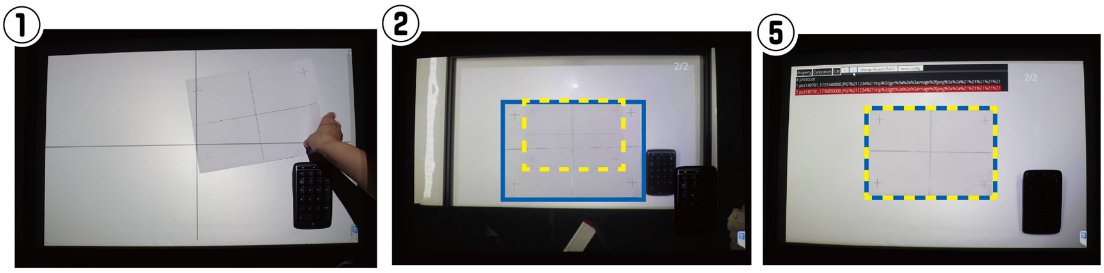

Figure 8: Usage of the calibration process. The numbers indicate the steps described in body texts.

6 Workshops & Exhibitions

We had exhibitions and workshops 6 times to explore the possibility of the FabNavi and obtain feedbacks from users. All exhibitions and workshops are listed in the Table 1. The workshop means an organized workshop with a few attendees (4 to 5 at each time), while the exhibition means a mixture of demonstration and unorganized workshop with many numbers of attendees. In both cases, some of our team members explained the basic usage of the system, and then observed the assembly processes of attendees with minimum disturbance.

The exhibition 1 & 2 were held to confirm possibilities of the concepts and basic functions of the FabNavi. We selected the projects, such as LEGOs, paper crafts, and wood crafts, which can be assembled with simple techniques like “inserting parts each other”.

The workshop 3 & 4 were held to confirm practical possibilities of the FabNavi to help workshop attendees assemble projects. We selected the IQLight lampshade project in considering following points: it is suited for the FabCafe setting, since the whole parts can be cut with a laser cutter; the assembly process is rather confused, so attendees might need system supports; the finished lampshade looks beautiful.

The exhibition 5 and 6 were held to mainly confirm the possibilities of the third prototype. Since these exhibitions were held in quite large events and most people did not stay at an exhibition long time, we selected simple LEGO projects again.

	Date	**Event**	**Contents**
1	Sep, 2013	Exhibition at Mozilla Festival Tokyo	LEGO1[8]
2	Oct, 2013	Exhibition at Mozilla Festival London	LEGO1, Paper Craft, Wood Craft
3	Dec, 2013	Workshop at FabCafe Tokyo	IQLight
4	Mar, 2014	Workshop at FabCafe Taipei	IQLight, Origami
5	Apr, 2014	Exhibition at Niconico Chokaigi (Tokyo)	LEGO2[9]
6	May, 2014	Exhibition at Maker Faire Taipei	LEGO2

Table 1: Workshops & Exhibitions of the FabNavi

The figure 9 shows the attendees and the figure 10 shows the example projects in each workshop & exhibition. Since most workshops & exhibitions focused on the re-assembly processes, the capture processes were performed by lab members of the project team.

[8] LEGO Crazy Action.

[9] A name card assembled with LEGO blocks.

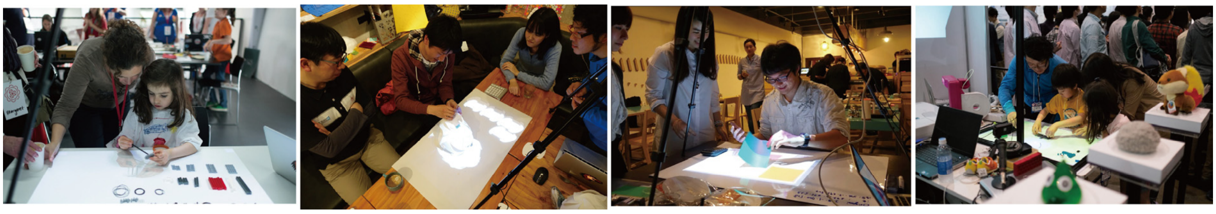

Figure 9: Attendees in the workshops & Exhibitions of the FabNavi. Mozilla Festival London (Left), FabCafe Tokyo, FabCafe Taipei, and Niconico Chokaigi Tokyo (Right).

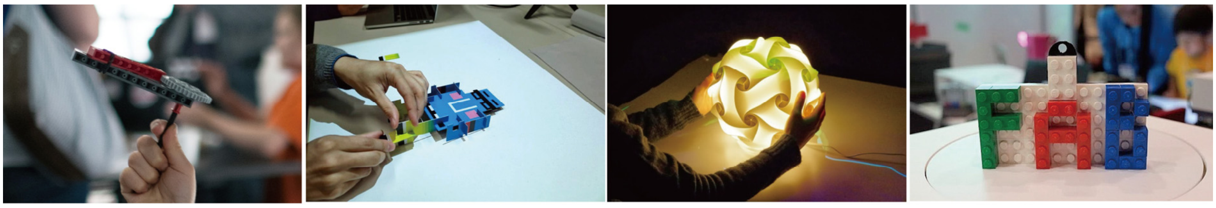

Figure 10: The example projects used in the workshops. LEGO Crazy Action (Left), paper craft, IQ Light, a name card built with LEGO (Right).

6.1 Findings from the workshops

In this section, we describe the findings from the workshops & exhibitions in several aspects. All findings were obtained from subjective data including observations and oral interviews.

6.1.1 System Performance on different assembly methods

The system performance seemed to be changed depending on the complexity of the assembly methods. For example, most people could easily assemble LEGOs, paper crafts, and wood crafts with the FabNavi support. In these cases, the assembly methods are quite simple: inserting parts (e.g., blocks or paper pieces) each other. In the meanwhile, some people have difficulty to assemble IQLight even when they have the FabNavi support. This problem caused from the assembly method of IQLight is quite complicated: twisting a piece, inserting it to other pieces, and rotating it to be fixed. Therefore, the support performance seems to be changed depending on the attendees' assembling skills. Moreover, since IQLight consists of only white pieces, some users have difficulty to check detail of projected images.

6.1.2 System performance on different language

The system performance did not seem to be changed depending on the language. For example, we held similar workshops to create LEGO projects in Tokyo and London. In these cases, most people in both countries can create the projects just with the FabNavi support. Meanwhile, we also held similar workshops to create the IQLight in Tokyo and Taipei. In these cases, the team members sometimes need to help the attendees by words or gestures when the attendees looked confused long time. For example some members said "Please bend it then insert!" and "You should keep the assembled parts with left hand since you will attach another part with it". In other cases, some member pointed the difference of the current/previous instructions with her finger.

We think these problems might be reduced by improving the quality of visual instructions. Moreover, we also consider developing intelligent functions (e.g., automatic emphasis of difference between pictures) as future work.

6.1.3 Observations of attendees

The attendees did not overlay real parts to the projected parts every time. In most case, they compare the state of real parts and projected parts by aligning them. This situation was especially observed in the IQLight project. In the meanwhile, the attendees sometimes overlay the real parts and projected parts to confirm the total shape. This situation was especially observed in the LEGO name-card project, in which they attached the color parts on the white base.

6.1.4 Feedbacks of creators

As mentioned above, the capture processes were mainly performed by lab members of the project team. So, we could obtain many feedbacks from them through oral interviews. We list their typical feedbacks as follows:

- Wireless key pad was easy to use, but sometimes I felt difficulty to push a key when working with two hands. Automatic capturing at regular intervals may solve the problem, but it becomes difficult to select required pictures.
- I think pointing or another annotation might be necessary to point which parts/points are focused in some cases. For example, I want to point the position of a new part in the IQLight project; I want to emphasis something in the picture such as technical points of assembly methods (e.g., direction of the IQLight parts)
- It is sometimes hard to take massive numbers of pictures.
- I sometimes need to consider the pictures are taken from directly above. For example, I leaned assembling objects to the camera to capture the location of new parts.
- I want a new function to remove pictures at once or save it as “mistaken examples” when I mistook to assemble parts and already captured them.
- The feedbacks of assemblers are very important for me. When I record contents just by myself, I am worried that these contents may become difficult to be understood by assemblers. I hope the mechanism to brush up instructions with the feedbacks of assemblers. Especially, the instruction step at which assemblers had difficulty is quite important.

7 Conclusion

This paper proposed a support system, *FabNavi*, which helps people to assemble physical objects in remote places using visual instructions. People can easily record the assembly processes by pictures and share these instructions on the web. Then other people can re-assemble them just by following the visual instructions projected on the remote tabletop. We implemented the whole system and performed workshops and exhibitions among the world. As next steps, we try to solve the problems found through the workshops. We are also discussing other design points of the system such as possible interaction techniques, browsing contents in common web browsers, using video instead of pictures, and annotation methods suited for the FabNavi.

The final goal of FabNavi is to create a framework for recording and sharing new universal instructions through the FabLab network. We plan to start from Hakodate, Tokyo, and Taipei as a first step, and then try to deploy the system among the world.

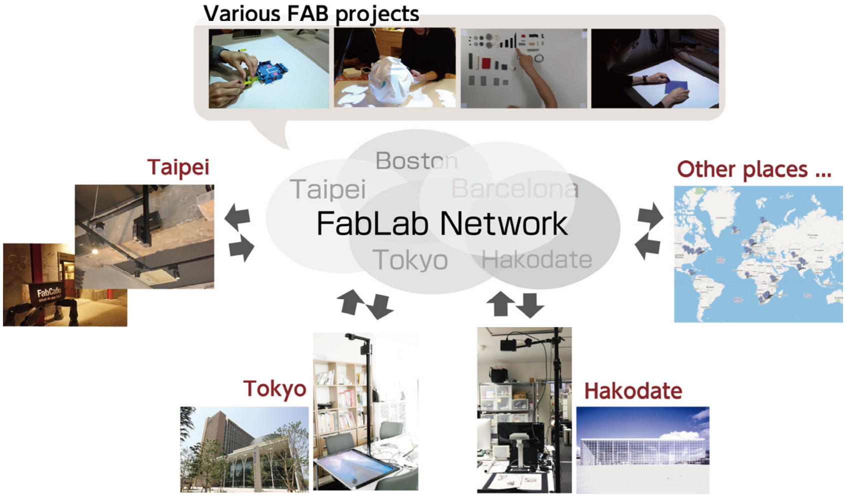

Figure 11: The goal of the FabNavi: Creating a framework for recording and sharing new universal instructions through FabLab network.

Acknowledgement

We appreciate Ken Teramoto who supports development of the web-based system. This work is partly supported by the JST PRESTO program and COI-T program.

References

[1]David Banes, Microsoft Surface - a new approach to access and technology, Journal of Assistive Technologies, Vol. 3 Iss: 1, pp.29 - 31 (2009)

[2]Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann, DemoCut: generating concise instructional videos for physical demonstrations. In Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13), pp. 141-150 (2013)

[3]Yuki Igarashi, Takeo Igarashi and Jun Mitani, Beady: Interactive Beadwork Design and Construction, Proceedings of SIGGRAPH 2012, Volume 31, Issue 4, Article No. 49 (2012)

[4] Hideki Koike, Yoichi Takahashi and Yasushi Matoba, AquaTop Display: Interactive Water Surface for Viewing and Manipulating Information in a Bathroom, ACM Int'l Conf. on Interactive Tabletops and Surfaces (ITS '13), pp.155-164 (2013)

[5] Manfred Lau, Greg Saul, Jun Mitani, and Takeo Igarashi, Modeling-in-context: user design of complementary objects with a single photo, In Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium (SBIM '10), pp.17-24 (2010)

[6] Ayaka Sato, Keita Watanabe, and Jun Rekimoto, MimiCook: a cooking assistant system with situated guidance, In Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction (TEI '14), pp.121-124 (2014)

[7] Toshiki Sato, Haruko Mamiya, Hideki Koike and Kentaro Fukuchi, PhotoelasticTouch: Transparent Rubbery Tangible Interface using an LCD and Photoelasticity, Proc. on the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST'09), pp.43-50 (2009)

[8] Toshiki Sato, Haruko Mamiya, Hideki Koike and Kentaro Fukuchi, "An Augmented Tabletop Video Game With Pinching Gesture Recognition", SIGGRAPH Asia 2008 Emerging Technologies (2008)

[9] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi, SketchChair: an all-in-one chair design system for end users, In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11).pp.73-80 (2011)

[10] Nobuyuki Umetani, Jun Mitani, Takeo Igarashi and Kenshi Takayama, Designing Custommade Metallophone with Concurrent Eigenanalysis,New Interfaces for Musical Expression++ (NIME++), pp.26-30 (2010)

[11] Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun, Sensitive couture for interactive garment modeling and editing, In ACM SIGGRAPH 2011 papers(SIGGRAPH '11), Article 90 , pp. 12 (2011)

[12] Pierre Wellner, Interacting with paper on the DigitalDesk, Commun. ACM 36, 7 , pp. 87-96 (1993)