

## SIGHT DIVE: DIRECTING FOR VIEWING THROUGH OTHER DEVICE'S CAMERA

Takashi Ohta  
School of Media Science  
Tokyo University of Technology

### ABSTRACT

We developed a camera application that enables a user to take pictures from another device's viewpoint. This is done simply by directing the user's own mobile device so that it catches the other device within its camera-view. This interface dynamically gives a user's device an ad-hoc access to another device's camera, and that connection continues for as long as the user's device remains pointed in the same direction. This research aims to demonstrate an attempt to use multiple devices in relation to each other. We laid particular importance on the interface design to create a novel approach for establishing a dynamic connection between multiple devices. We implemented a prototype application to prove the concept and tested its validity.

### 1. INTRODUCTION

Since the digital camera was integrated into the smartphone, many people have become able to take pictures under any circumstances because smartphones are now ubiquitous. While digital cameras are very easy and convenient to use, they have a limitation in that they can take pictures only from the viewpoint of the device's physical position. Often there are cases where the camera's zoom function is not sufficient to produce an ideal image or the line-of-sight to the object becomes blocked. It would be more convenient to the user if they could set the camera's viewpoint to the ideal location from any position. With this in mind, we came to consider that an ideal solution to such a situation would be borrowing the camera of another device that is positioned at a more convenient angle and location. If such a function were realized, it would be possible to scatter multiple devices around a place in order that anyone could get an ideal view by borrowing one of these cameras' viewpoints at any moment.

Technically, it would not be difficult to achieve such a function. It is a matter of pairing two devices dynamically and transmitting the captured video in real-time. A typical user interface for such a function would be one that displays thumbnails of all the captured video available at that moment, and lets a user choose the desired one by tapping it. However, we do not want bothersome steps, such as selecting menus or icons, because these stages would harm the usability of the

device as a camera. What we believe would be preferable in such a function is an intuitive interface to maintain the ease of taking a picture; namely, directing a camera and pressing the shutter button. We designed the interface so that one can use another device's camera simply by directing one's own device to face the other, allowing it to take a picture from the other device's view as if it were its own.

In this article, we report our first attempt at building a prototype for proving this concept. This prototype intends to demonstrate the interaction of the user interface and the basic function.

### 2. CONCEPT OF SIGHT DIVE

Sight Dive is implemented as a camera application. Its interface allows users to access another device's camera and use its viewpoint as if it were their own camera's image. Fig.1 illustrates the concept of Sight Dive. The upper image illustrates a case where the target is too far away to be captured at an ideal size in a picture. Sight Dive lets users take a photo by using a secondary device's camera that is placed nearer to the target than their own, as demonstrated in the lower illustration of Fig.1. With such a function being realized, people could take their ideal photo by choosing a better view and borrowing a camera at an appropriate position from among the devices scattered around a location.

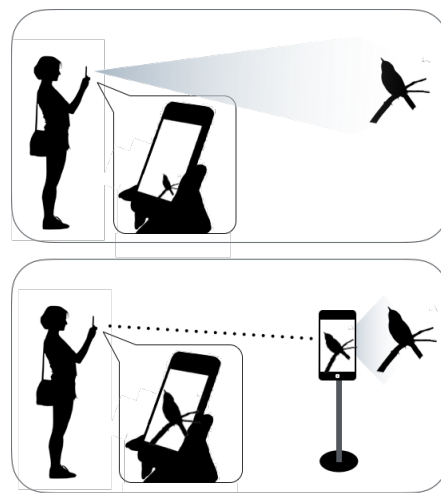


Fig. 1 Concept of Sight Dive

This work attempts to use multiple devices in this way by establishing an ad-hoc relation between them. Many attempts have been made in this research field so far. Various mechanisms have been introduced to establish a connection between multiple devices. There are approaches that use an extra sensor to detect the physical contact of devices [1,2] and use it as a trigger for establishing a network connection between the devices. Google's "Tone" utilizes the detection of a tone as a trigger [3]. These works focused on creating a temporary co-working environment and file sharing system; therefore, it is sufficient for these applications to only have a signal for deciding which devices to pair up. Some studies aimed to form a larger display using multiple devices, and so required information on each device's relative position. Hinckley's work used trails drawn by a stylus pen that were applied across multiple devices' screens [4]. Another major approach is to use an external camera for detecting the devices' placements [5,6]. The common objective of these projects is to realize a function of each research theme. Therefore, the mechanisms or the user interfaces were designed to provide support for using that function.

We also have been carrying out research on utilizing dynamic relations between multiple devices [7,8]. With these researches, we put more weight upon the design of the user experience than on the realization of a function. Rather, the user experience design is our main motivation for undertaking this research. The same function could be offered by various different interface designs. We especially focused on designing a user interface that would be pleasing to use and fulfill the expectations of the user's imagination regarding such a function. With our Pinch interface, we imagined the action of pinching, and hence merging, two clay objects together as being analogous to the connection of applications running on different devices. With the Sight Dive application, we took inspiration from the point of view that seeing leads to a connection.

### 3. USER INTERFACE DESIGN

A typical interface for selecting a camera's view from multiple candidates would display thumbnails of all the videos being captured, and let the user choose from among them. However, we do not want laborious steps such as selection menus or on-screen icons, because being required to maneuver through these would harm the usability. We intend, with such a function, to have an intuitive interface to maintain the ease of taking a picture, an interface as simple as directing a camera and pressing the shutter.

This project requires that there be no interruption in switching from the user's own camera to that of another device. To achieve this, we designed the interface so that one can access the other device's camera merely by directing one's own device towards it. Fig.2 illustrates the basic design of the interface for Sight Dive. In an ordinary state, the application works as a standard

camera application. When the device is directed so that its camera captures another device on screen, the application dynamically links to the other's camera, and the video being taken by it will appear on the screen of the device in hand. In both cases, you can take a photo by tapping the screen. When more than two devices exist in the same direction, you can control the selection by applying a pinching in/out gesture to the screen, as shown in Fig.3.



Fig. 2 Selecting a device by direction

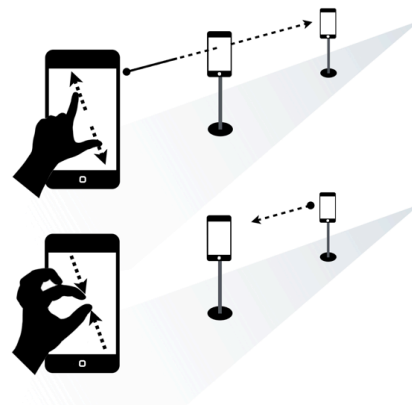


Fig. 3 Selecting from devices exist in same direction

We prepared a visual effect so that the user will be alerted to the detection of another device. The screen's color will fade to monochrome when the camera is directed closer to another device; the closer the direction of the camera comes to the location of another device, the more the color of the screen fades (Fig.4). If the camera is then held in the position where the screen turns monochrome, a graphic of a waiting circle appears

and the image on the screen will then switch to the image being received by the detected device's camera.

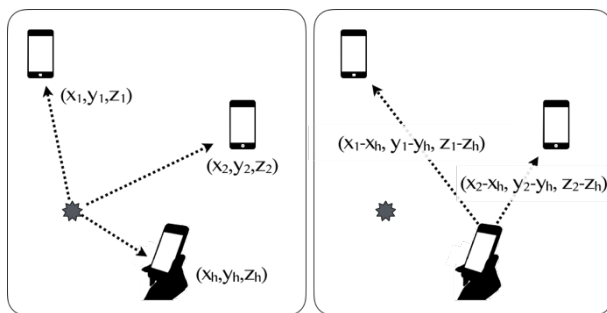


**Fig. 4 Visual effect for switching a camera**

#### 4. SYSTEM IMPLEMENTATION

We implemented the system as a mobile device application. When the application starts, the user can set the mode to one of two distinct roles. The first is to allow the device to work as a video provider. The second allows the device to work as a browser, which is the mode that allows the user to detect devices in the video provider mode, and so, is used by the device in the user's hand.

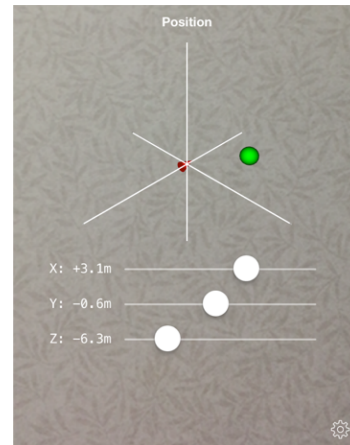
Because a browser should be able to locate another device by direction, the application needs to know the devices' geometrical relative positions. We decided to set every device's location as a position relative to an origin (Fig.5). This data is assembled on an external server, which the browser can access to calculate the direction and distance to each device. With the current implementation, we let a user set each device's location manually by a configuration panel, shown in Fig.6.



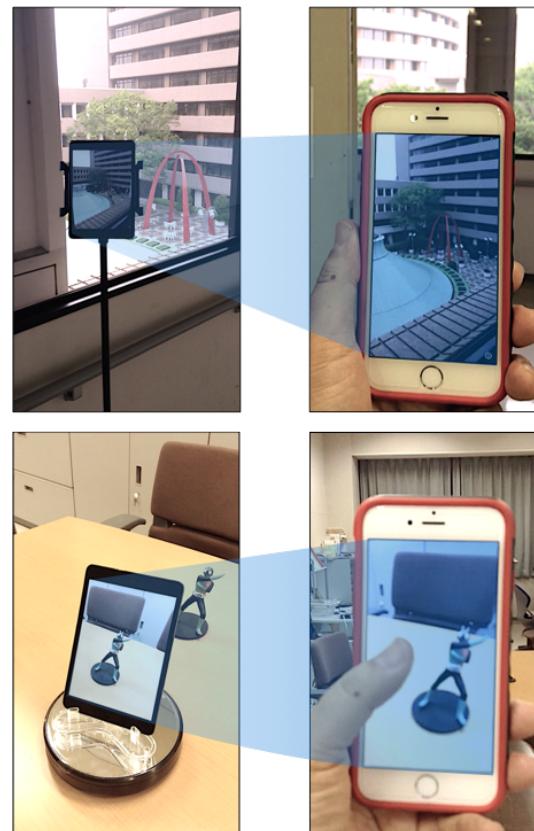
**Fig. 5 Calculation of relative positions**

A target device is found by comparing the direction obtained by the built-in gyro-sensor to the direction calculated using the preset geometric data. When there are multiple devices in the same direction, the zooming factor is considered to determine which one to connect to. When a browser device finds an appropriate video provider device, the application establishes a network connection between these devices, and starts to transmit video. The video image taken by the provider device appears at the browser device's screen as it is received

in real-time (Fig.7), and the user can save a picture by tapping the screen. The action for taking a picture is the same whether it uses the device's own camera or that of another device.



**Fig. 6 Configuration panel for setting location**



**Fig. 7 Prototype application working**

#### 5. EVALUATION

We tested the function and usability of our concept with a prototype application. We placed four video provider

devices at the corners of a conference room. A user stood at the center of the room and directed his browser device towards each provider to examine whether a connection was established and a video started to transfer when these devices were captured within the browser's camera view. The application worked sufficiently well overall. However, we observed that the recognition of the direction deviated when the application was used continuously, and needed to be calibrated periodically. We assume that this deviation originated from the accumulated sensor error. We need to evaluate further the magnitude of the error's effect under various conditions, in order to improve the reliability of this aspect of the device's function.

Regarding usability, the application provided very easy access to use the camera functions of other devices, especially comparing the process to an interface that employs a configuration panel to choose a device and establish a connection. It was also sufficiently clear when another device had been detected through the color-fading effect described above. This interface does not interfere with the screen and therefore is very suitable for being employed as a camera application. One issue that arises is the disorientation of the user that can occur due to the image on screen not responding to the movement of the user's device after it is attached to another device.

## 6. DISCUSSION

Even though there are some aspects to improve, we believe that the prototype application succeeded in demonstrating the concept of Sight Dive. We expect this research can provide a different viewpoint in research fields, such as ubiquitous computing and intuitive interface design. Most research in the field of ubiquitous computing or the IoT (Internet of Things) focuses on the creation of services or on management systems. What we intend to present through this research is a more casual method of connecting digital functions and making them a part of a living environment.

Regarding interface design, we propose a different approach in terms of designing an intuitive interface for digital functions. With concepts such as TUI (Tangible User Interface [9]), the intuitiveness of an interface is based on knowledge of the usage of existing physical tools. This concept is put in practice by mapping a digital function to a physical object, the usage of which users are expected to be familiar with, and letting the object act as the interface. We did not follow the approach of borrowing an idea for user interface design from an existing tool, but instead tried to realize what a user would expect and imagine from the interface of a device with this function.

We believe that Sight Dive would be useful in locations such as sport stadiums. Such venues are generally too large for a user to take a clear picture of scenes happening on the ground. If multiple provider devices were equipped around the stadium, an audience

member would be able to take a picture by directing their own device towards the most appropriate provider (Fig.8), and could switch to use an appropriate device at any moment. The same setup would be useful for a parade event at a theme park, where audiences are crowded, making it difficult to take an ideal picture.



**Fig. 8 Usage scenario for a sport stadium**

Some may say that displaying thumbnails has a certain advantage over our approach, because a user can choose the most desirable view by browsing all the available scenes. With our interface, a user can find a desired view by directing own device towards other device by device. We expect this would not harm the usability much because it only requires moving one's device around. Besides of such comparison, we would like to insist that our intention is not in providing the function with a convenient interface. What we pursue in this research is a design of a novel interaction for using multiple devices in relation.

## 7. CONCLUSION

We designed an interface that enables users to borrow another device's camera function casually by using an easy and simple interaction. The interface works exactly as a standard camera application, directing the camera to a target and tapping the screen to save an image. The prototype works sufficiently well to demonstrate the validity of the idea. At the same time, it revealed aspects needing improvement, especially in that the method by which the devices' relative geometry is set needs to be automated. We also have to tackle the problem with the gyro sensor's accumulated error in estimating a device's direction. As for future development, we plan to design a system that enables multiple users to access the same device simultaneously.

## 8. REFERENCES

- [1] K. Hinckley: "Synchronous gestures for multiple persons and computers," In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, UIST '03, 149–158 (2003).
- [2] D. Merrill, J. Kalanithi, and P. Maes: "Siftables: Towards sensor network user interfaces," In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, ACM, New York, NY, USA, TEI '07, 75–78 (2007).
- [3] Tone: An experimental Chrome extension for instant sharing over audio,  
<https://research.googleblog.com/2015/05/tone-experimental-chrome-extension-for.html> (2015)
- [4] K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith: "Stitching: Pen gestures that span multiple displays," In *Proceedings of the Working Conference on Advanced Visual Interfaces*, ACM, New York, NY, USA, AVI '04, 23–31 (2004).
- [5] R. Rädle, H.-C. Jetter, N. Marquardt, H. Reiterer, and Y. Rogers: "Huddlelamp: Spatially-aware mobile displays for ad-hoc around-the-table collaboration," In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ACM, New York, NY, USA, ITS '14, 45–54 (2014).
- [6] MIT CENTER FOR CIVIC MEDIA, Junkyard jumbotron,  
<https://civic.mit.edu/blog/csik/junkyard-jumbotron> (2011)
- [7] T. Ohta and J. Tanaka: "Using pinching gesture to relate applications running on discrete touch-screen devices," *Int. J. Creat. Interaces Comput. Graph.* 4, 1 (Jan.), 1–20 (2013)
- [8] T.Ohta: "Intuitive Gesture Set on Touch Screen for Ad-hoc File Sharing among Multiple Mobile Devices", *IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pp.751-752 (2014).
- [9] H. Ishii and B. Ullmer: "Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms," In *Proceeding of CHI '97*, pp.234-241 (1997).