

# Ad-Hoc Multi-Displays for Mobile Interactive Applications

Arne Schmitz<sup>†</sup>, Ming Li<sup>‡</sup>, Volker Schönefeld<sup>§</sup>, Leif Kobbelt

Computer Graphics Group, RWTH Aachen University, Germany

---

## Abstract

We present a framework which enables the combination of different mobile devices into one multi-display such that visual content can be shown on a larger area consisting, e.g., of several mobile phones placed arbitrarily on the table. Our system allows the user to perform multi-touch interaction metaphors, even across different devices, and it guarantees the proper synchronization of the individual displays with low latency. Hence from the user's perspective the heterogeneous collection of mobile devices acts like one single display and input device. From the system perspective the major technical and algorithmic challenges lie in the co-calibration of the individual displays and in the low latency synchronization and communication of user events. For the calibration we estimate the relative positioning of the displays by visual object recognition and an optional manual calibration step.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Distributed/network graphics—I.3.6 [Computer Graphics]: Interaction techniques—I.3.1 [Computer Graphics]: Input devices—

---

## 1. Introduction

There is currently a large class of mobile devices emerging which have several properties in common. First, those devices are essentially general purpose computers, i.e., they use adapted desktop operating systems and are programmable using standard programming languages. Second, they all have relatively large, touch sensitive screens. Third, they have advanced graphics capabilities, often supporting 3D graphics via the use of OpenGL ES. Lastly, all those devices have some sort of wireless communication system built in, either GSM or 3G based, WiFi or Bluetooth. Those devices are mostly mobile phones and media players and they are receiving a huge popularity.

Until recently, those devices and especially the applications running on them barely utilized the fact that peer to peer communication between multiple devices can be used to build some kind of ad hoc network. Such a network of devices can in turn be used for distributed rendering purposes and exploration of new interactive systems consisting of multiple, touch sensitive devices.

In this work we have designed a system that utilizes those emerging smart mobile devices to build an ad hoc network

of displays and to explore possible settings for multimedia applications and interaction scenarios. Our contributions are first an extension to traditional multi-touch interfaces to support multiple mobile devices, where gestures can span many screens. Our second contribution is the development of a simple and efficient calibration workflow which includes an automatic calibration step and a simple, additional manual calibration step for the imaging device, that has not been included in the automatic setup. We developed an intuitive pattern that effectively supports the manual calibration. See Figure 1 for an overview of the calibration workflow. The use of a wireless network for such an interactive system gives rise to possible problems such as loss of packets or latency. This is tackled by our method as well.

### 1.1. Possible Applications

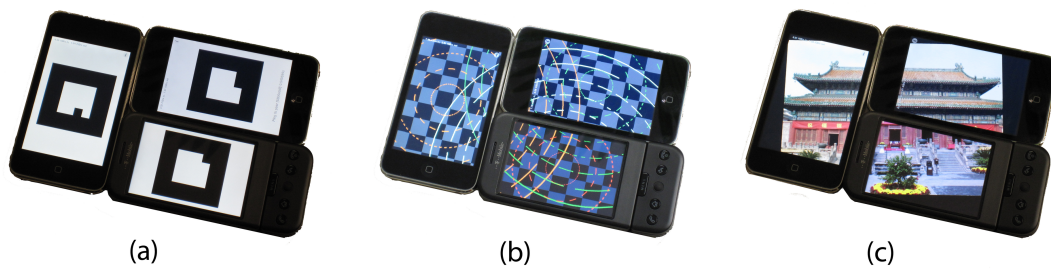
The purpose of our proposed system is to provide the functionality for building a larger, interactive display from multiple, individual devices. This type of display is useful in a scenario where a group of people meets spontaneously and wants to share or present information in a visual manner. For example our system makes it possible to show a slideshow presentation or view pictures and photographs, although no projector is at hand, while not being limited to the small screen size of a single mobile phone. We call this mode of operation a *distributed rendering canvas*. It consists of one large rendering context with different segments of the viewport assigned to individual devices.

---

<sup>†</sup> Supported by the UMIC Research Centre, RWTH Aachen.

<sup>‡</sup> Supported by the B-IT Research School.

<sup>§</sup> Supported by Deutsche Forschungsgemeinschaft (IRTG 1328).



**Figure 1:** The three steps of getting our multi display to run. (a) The clients display markers for automatic calibration. (b) An optional manual calibration step can be performed. (c) The display is calibrated and ready for interactive use. The image shows a heterogenous setup consisting of an iPhone, iPod Touch and an HTC Dream.

This mode also enables the display of high resolution pictures with more details. Although current handheld devices already have displays with pixel densities of about 200 pixels per inch, the overall display resolution is often limited to  $320 \times 480$  pixels. Our framework enables devices with different pixel densities to work together, see Fig. 1 (c). In order to make out details in photographs one usually has to zoom in really closely on such handheld devices. With a distributed rendering canvas as we propose, one can zoom in on details, without getting lost, since the display size becomes much larger.

## 2. Related work

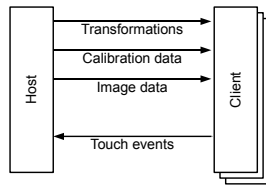
There are already several display systems with multiple screens available on the market. The Nintendo DS portable gaming system uses a clam-shell design with two smaller displays so that when open, the two displays can be used to increase the available display area. Similarly, Lenovo released a dual-screen laptop, the ThinkPad W700ds, which offers a 17-inch screen along with a secondary 10.6-inch slide-out display. Intel demonstrated a concept laptop, Tangent Bay, that integrated an extra three OLED multi-touch screens above the keyboard. The user can organize files and control the media player in the three mini-screens to save room in the main screen. Between mini screens, the laptop supports multi-touch and file-dragging gestures. This shows that multi-display technology is growing, especially in the area of mobile devices, but still they are relatively static in their setup, whereas our work allows for a very dynamic setup.

Merrill et al. [MKM07] designed a platform, Siftables, that uses several very small displays to form a tangible user interface. Although they allow for very novel input metaphors, their devices are very small, and have very limited computational power and interaction capabilities. Also the shown devices are not off-the-shelf hardware, in contrast to the systems used in our approach. Tandler et al. [TPMT\*01] presented ConnecTables which links

two LCD displays together dynamically to build a shared workspace for cooperation. Hinckley et al. [HRG\*04] introduced Stitching, an interaction technique which allowed users to combine two pen-operated mobile devices with a wireless network. They applied pen-gestures to interact between different screens. To connect more than 2 devices, Lyons et al. [LPR\*09] presented a multi-display composition system for collocated mobile devices over a wireless network. Both works use larger mobile devices, and in parts restrict the user to regular screen layouts, like grids. Our work on the other hand allows for non-standard screen layouts on small handheld devices, i.e. non-rectangular displays, see Fig. 3.

The principle of multi-touch input metaphors has been explored for a long time. In 1982, Mehta [Met82] introduced the first multi-touch display which could detect more than one contact point on the screen simultaneously. Then Nakatani and Rohlich [NR83] presented a “Soft Machine” that combined the touch screen input with the real-time graphical response for human-computer interaction. In 1991, Wellner published his “Digital Desk” [Wel91] which supported a tangible manipulation of presented data using fingers, e.g. pinching for scaling a picture. Of relevance is also the work by Westerman [Wes99], who introduced chordic manipulation. We utilize and extend these works by allowing the user to use multiple finger gestures on arbitrary devices in our system.

To construct a scalable, high-resolution display from many individual ones, a fundamental challenge is to avoid visible seams due to misalignment among the displays. To tackle this, many researchers proposed methods for automatic alignment of multi-projector displays using cameras for registration. Chen et al. [CCF\*00] presented an automatic alignment method using an uncalibrated camera to measure the relative mismatches between neighboring projectors, and then correct the projected imagery to avoid seams. Raskar et al. [RBY\*99] introduced a technique to build a 3D visualization system for panoramic display envi-



**Figure 2:** The clients send touch events to the host, who in turn computes transformation matrices, and sends image as well as calibration data to the clients.

ronments. They proposed a surface mesh unification method to ensure the geometric continuity across projector boundaries. Chen et al. [CSWL02] addressed a vision-based geometric alignment system for aligning the projectors in an arbitrary large display wall. They utilized a camera homography tree to automatically register any number of uncalibrated camera images. These works allow for dynamic and irregular display arrangements, but rely on projectors, while our work concentrates on common handheld mobile devices, which are more ubiquitous.

### 3. System Structure

The system presented here is based on a client-server architecture. One of the mobile devices acts as a *host*, while all the other devices act as *clients*. The host is selected by the user at the beginning of the assembly of the ad hoc display, while clients can join at any later time. The information stored on the client are mainly its relative location in the ad hoc display, i.e. a viewport transformation, which defines a window within the larger, distributed rendering canvas. The information stored on the host are the current position of fingers on the different screens. The host collects this information and computes the gestures and the resulting transformations. It then broadcasts the transformations back to the clients, as well as calibration data and image data, see Figure 2.

#### 3.1. Supported platforms

Our system supports devices running the iPhone OS and Android, e.g. iPod Touch, iPhone, HTC Dream, etc. Both systems can be used in a heterogenous setup. Moreover, the software could be ported to any other device, supporting a touch screen, sufficient rendering capabilities and either a Bluetooth or WiFi connection (either LAN or Internet based).

#### 3.2. Network Subsystem

With iPhone OS 3.0, it is possible to set up an ad hoc network between supported devices using the GameKit framework. However, while it has the big benefit of not requiring additional hardware, such as access points, this Bluetooth based method also has drawbacks. It only allows up to three

devices in a network with a maximum throughput of around 60 KiB/s. Furthermore this approach is still somewhat experimental and does not offer much control, since it introduces an abstraction layer which is not very flexible and does not support the Android platform.

Hence we implemented a second approach, based on WiFi that offers higher throughput (up to 600 KiB/s), a more reliable connection, and a large number of devices. Up to seven devices have been tested without any negative impact on the performance.

The latency in both modes depends heavily on the local conditions of the wireless channel. Packets might get dropped or resent because of varying quality of the wireless link. For our test setups the round-trip time for both approaches averaged at around 40ms, implying a one to two frame delay between input and reaction, assuming a 25 frames per second update rate.

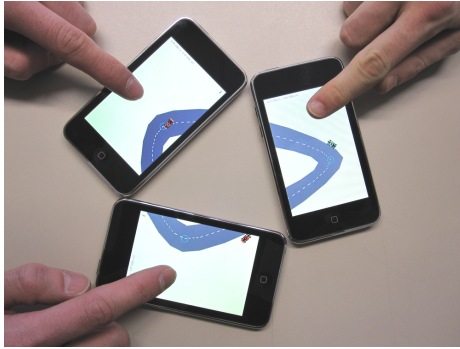
### 4. Calibration Process

Since the ad hoc display consists of multiple screens, to keep the visual continuity at the boundary, both the orientation and position of each participating device need to be calibrated, so that every individual display shows the correct part of the global viewport. Therefore after calibration, every device taking part in the multi-display stores a local viewport transform. There are two different coordinate systems: the local, device dependent coordinate system, and the global device independent coordinate system. To reduce the complexity of this task and to adjust the alignment quickly, our calibration process includes two steps, one is an automatic pose estimation based on optical marker detection, the other is a manual fine tuning based on the users' assessment.

#### 4.1. Pose Estimation

Marker based tracking has been studied for over a decade. Rekimoto [Rek98] first introduced a 2D matrix code using camera-based 6DOF tracking of artificial 2D markers. He used a square, planar shape for pose estimation, and an embedded 2D barcode pattern for marker identification. After that, Kato et al. [KB99] developed a tracking library called ARToolKit, which is widely used in AR applications.

In our ad hoc display, we utilize the marker based detection algorithm by Kato et al. to estimate both the orientation and position of each participating device. Every client renders a unique marker that it receives from the host. The host phone is used to take a photo of the entire setup. It then processes this photo, detects all markers, and returns the global coordinates and orientation of each marker. Therefore, the local viewport transform of each client can be computed from their relative positions. Note that a viewport transformation in this case only consists of rotation and translation. The scale of the device is known for each device individually, since these values are device-dependent and static.



**Figure 3:** We designed a car-racing game to show the flexibility of our ad hoc multi-display system.

The transformation matrix is then sent to the corresponding client. Once the client receives its new viewport, it updates its view and displays the pattern for the following, optional manual fine tuning. The automatic calibration might fail if bright reflections obstruct the client display's visibility.

The automatic calibration enabled us to create cooperative, mobile applications with arbitrary display layouts. To demonstrate the flexibility of our ad hoc multi-display, we designed a car-racing game (see Figure 3). This game allows random multi-players to participate. After the auto calibration, the race track will be automatically generated by the placement of the devices. In this way, the users can intuitively customize their own track.

#### 4.2. Manual Fine Tuning

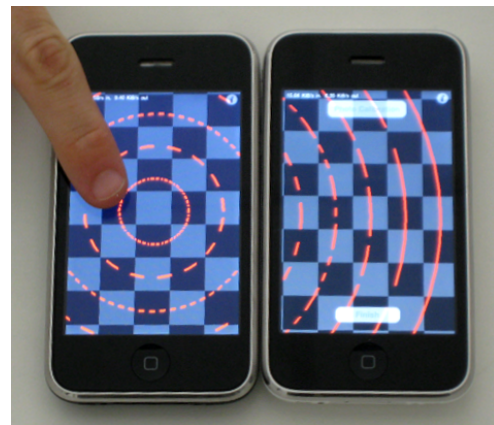
Since the host phone is used as a camera in the pose estimation step, its viewport is not reconstructed yet. The user needs to align the host to the neighboring clients, so that it gets the correct local viewport. Furthermore, the user has the opportunity to adjust the alignment of all other devices as well, in case there was a misalignment by the automatic registration, or if the user wishes to move some of the phones after the initial registration. The user can also skip the automatic calibration step completely and use only the manual tuning. This is done by aligning the calibration patterns manually through drag and rotate gestures. In the evaluation section, we will compare the efficiency of these two calibration methods.

#### 4.3. Calibration Patterns

For the manual tuning step, we utilize several visual patterns to help users align adjacent displays. A good calibration pattern needs to be able to tell the user where the center of the neighboring device is, and if the alignment is already sufficient. On each display, we render a checkerboard pattern and several concentric circles in different styles (dashed line, dotted line, short dashed line, etc.). The checkerboard is well suited to show the orientation of the viewport of the current



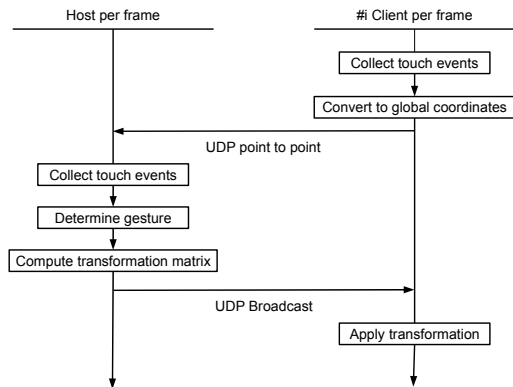
**Figure 4:** The checkerboard grid shows the global canvas. The circles denote the distance from the device's viewport center.



**Figure 5:** During the manual calibration, only the concentric circle of the currently touched screen is shown on the multi-displays

device. Due to their strong symmetry properties, the circles allow the user to precisely set the relative position of the devices (see Figure 4).

During the development, we invited some users to try out our prototype. This revealed that the method of displaying the calibration patterns were confusing. During the manual fine tuning step, all the calibrating patterns were shown on the whole display surface at the same time, so it was hard for the users to tell which calibrating circle belongs to which screen. Therefore aligning the adjacent displays became a difficult task. Hence we modified the display of the calibration patterns, so that during the manual fine tuning only the concentric circle of the currently touched screen is shown on the whole display surface (see Figure 5). In the following user test, the users gave that method a much better ranking, saying that it helped them to concentrate on the current tuning task.



**Figure 6:** The packet flow between host and clients for the multi-device multi-touch metaphor.

## 5. Synchronization

The main aspect of our system is the ability to show large images or visuals on a multitude of smaller viewports, forming a tiling display. Two aspects of this system need to be synchronized in order to get a good user experience: The input and the interactive feedback of the data to be displayed. For this to happen both the clients and the hosts run an event loop. Also care has to be taken when to use TCP and when to use UDP transport, to keep latency minimal, but still maintain reliability. We decided to send all user interactions via UDP, and all image and calibration data via TCP, to ensure reliability.

### 5.1. Multi-Device Multi-Touch

As an intuitive interaction metaphor multi-touch input has been researched in the last decades, and is now being widely employed on mobile devices. However, for patent reasons not every touch sensitive mobile device supports multi-touch gestures (such as the HTC Dream). For this purpose our multi-device multi-touch (MDMT) metaphor enables touch gestures spanning multiple devices. Our system enables the user to use one or two finger gestures. One finger is used for translating objects. Two fingers are used for scaling and rotating objects.

In our system, the touch events of all devices are collected individually, converted into the global coordinate system and sent periodically via an unreliable connection to the host. The host then accumulates all those touch events, and decides upon which gesture is currently exerted by the user. Depending on the gesture, an affine transformation on the displayed object is performed, like translation, rotation, and scaling. The resulting transformation is broadcast from the host to the clients again via an unreliable connection (see Figure 6), since it is very important that the latency between touch event registration, gesture recognition and transformation broadcast is as low as possible. Reliable TCP con-

nections guarantee the in order delivery of the data stream. However, in the case of packet loss this may lead to massive backlogs and latencies. Since we send transformations with a rate of about 60 Hz, it is not important if some packets are dropped or if they arrive out of order. A packet counter simply drops packets that were received in the incorrect order.

### 5.2. Tiling Display

The main application that we propose is an ad hoc tiling display. Since the devices used in this display are somewhat heterogeneous, every device must know about its own physical screen dimensions. So there has to be some mapping from physical, real world coordinates to the canvas' coordinates, since screen size, screen resolution and pixel density may vary from device to device.

Thus every device has knowledge about its screen dimensions both in pixels as well as in millimeters. This allows us to compute the pixel density and set up a mapping from real world and device dependent coordinates to coordinates in the reference frame of the multi-display. See Figure 7 for example screen layouts.

## 6. Evaluation

In the evaluation of our system we concentrated on the usability of the system, the usefulness of the calibration pattern we developed, and the perceived visual disturbance caused by the device borders.

### 6.1. Participants

We invited 30 participants to our user study. Of those 24 were male and 6 were female. The average age was 25.6 years ( $\sigma = 2.6$  years). Only one participant did not own a mobile phone. The other subjects did own phones with an average age of 3 years ( $\sigma = 1.6$  years). So our sample group is rather young, and technology-aware, but because of the average phone age, they most probably do not own a phone with a large touchscreen. Some test subjects explicitly mentioned that they never have used an iPhone-like device before. The participants were divided into two equally sized groups, which were well mixed according to gender and age. The test subjects in the first group calibrated the display manually, while the people in the second group did this using the automatic pose estimation. The purpose was to evaluate the efficiency and accuracy of these two calibration methods without bias. The users' satisfaction with each method was also an important criteria for our evaluation.

### 6.2. System Configuration

For the evaluation, we used multi-touch devices running iPhone OS 3.1 and Android 1.6. All the devices have a resolution of 320×480, and their actual display sizes are 3.5

**Table 1:** Results of the questionnaire concerning quality of calibration, difficulty of the task and the perceived influence of the device borders. The valid range for answers was 1 (worst) to 5 (best) on a discrete scale. In the table  $\bar{x}$  denotes the average value,  $\tilde{x}$  the median and  $\sigma$  the standard deviation.

Configuration	Quality (User)			Quality (Observer)			Difficulty			Borders			Avg. Time	
	$\bar{x}$	$\tilde{x}$	$\sigma$	$\bar{x}$	$\tilde{x}$	$\sigma$	$\bar{x}$	$\tilde{x}$	$\sigma$	$\bar{x}$	$\tilde{x}$	$\sigma$	m:s	$\sigma$ (s)
Linear Auto	3.8	4	1.01	4.5	5	0.64	1.8	2	0.68	3.1	3	1.28	01:28	93.5
Matrix Auto	3.4	3	1.13	4.1	4	0.74	2.1	2	0.83	3.5	4	1.25	01:45	92.0
Linear Manual	3.3	4	1.11	3.5	4	1.25	2.7	2	1.23	2.4	2	1.12	02:11	116.6
Matrix Manual	2.8	3	1.08	3.0	4	1.41	3.5	4	0.92	2.7	2	1.45	03:05	102.8



**Figure 7:** The two configurations used in our user study. The top one is the  $4 \times 1$  linear configuration. The bottom one is the  $2 \times 2$  matrix configuration.

and 3.2 inches. The user was asked to calibrate two configurations. A linear with a  $1 \times 4$  layout of the devices, and a matrix configuration with a  $2 \times 2$  layout, see Fig. 7. We chose these two setups, since they exhibit the smallest and the widest spatial gaps between neighboring displays. We asked the users how noticeable and disturbing these gaps were while using the system.

### 6.3. Tasks

Each participant had to perform two tasks in our study: calibrating a multi-display in a linear configuration, and a matrix configuration, as shown in Figure 7. One group did this by using the manual calibration method, the other used the automatic calibration method. The primary purpose of these tasks was to test how much easier the automatic calibration is for the user, and also how the display borders in the different configurations affected the user's experience of the multi-display.

### 6.4. Procedure

Each experiment took 15 minutes on average, including the tutorial and the execution. At first the user filled out a form

with his personal background information, then we gave him or her an introduction to the system, which took about three minutes. We showed him once how to calibrate the display using either method. Then the user would try to calibrate the display without help. Half of the users did the linear setup first, and then the matrix configuration. The other half did the tasks the other way around. For each task we measured the time needed to calibrate the display. Afterward we asked the user several questions. In particular we asked how difficult the task seemed, what the quality of the resulting calibration seemed to be, and if the borders had any influence on the alignment and the viewing experience. All questions in the interview could be rated on a Likert scale between one (least agreement) to five (most agreement). Additionally, the calibration quality was also judged by a neutral observer, who supervised all the experiments. This was done to make it possible to compare the results of the individual experiments. We did not measure the accuracy in pixels, because due to the device borders it is not noticeable if the devices are not perfectly aligned.

### 6.5. Results

Three aspects of the experiment were of great interest to us in the performed user study. First, the perceived quality of the calibrated display, second the time needed for the process, and third how intuitive it is to use the multi-device multi-touch metaphor. The results of the study are presented in Figure 8.

Concerning the display quality, the users themselves answered differently, depending on if they were in the manual or the automatic calibration group. The automatic calibration was judged significantly better than the manual calibration both for the linear and for the matrix setup, as can be seen in Table 1 and Figure 8 (a) and (e). This is also true for the rating that the external observer gave, as can be seen from the same table and Figure 8 (b) and (f). The difficulty was rated much easier for the automatic calibration than for the manual method, but also the linear setup was rated easier than the matrix configuration (Tab. 1, Fig. 8 (c) and (d)).

The second focus of the study was the time consumption to set up the display. As can be seen in Table 1, the automatic calibration method was 32% faster on average than the manual calibration. The time consumption of the linear

and the matrix setup in the automatic case differed by only 4% on average. On the other hand for the manual method, the difference was 30%. Also, four users did not complete the manual matrix setup at all, since they thought it was too complicated. Thus an automatically guided setup is helpful and makes arbitrarily shaped display configurations possible. In the case of the automatic setup, the user only has to manually adjust one master device, irrespective of the total number of devices.

Furthermore we asked the users to judge the influence of the display borders in the assembled multi-display. Table 1 shows that the borders did not matter as much to the users of the manual setup group than to the other users. However the standard deviation for both groups is quite high, which means that the borders were perceived very differently for individuals.

In the last part of the questionnaire, we asked the users to answer some questions on the usefulness and acceptance of the proposed system. The users found the system to be quite intuitive ( $\bar{x} = 4.3$  for both groups). The auto group was more willing to use the system for viewing pictures ( $\bar{x} = 3.5$  versus  $\bar{x} = 2.7$ ). Overall, the users found the system relatively useful ( $\bar{x} = 3.7$  versus  $\bar{x} = 3.5$ ), albeit this question had a greater variance compared to the other questions. This might be due to the fact that the users were only shown one application, i.e. the picture viewer. Lastly, the auto users were more likely to use this application in a meeting or at a social occasion ( $\bar{x} = 3.5$  versus  $\bar{x} = 2.8$ ), which is due to the smaller amount of work needed for setting up the display manually.

## 7. Conclusion

In this work we have shown how to build a multi-device multi-touch display, which is comprised of commodity mobile devices. In comparison to previous work, we have shown how to provide an easy calibration of the display, with both automatic and manual methods. We have provided a framework for building mobile, cooperative applications.

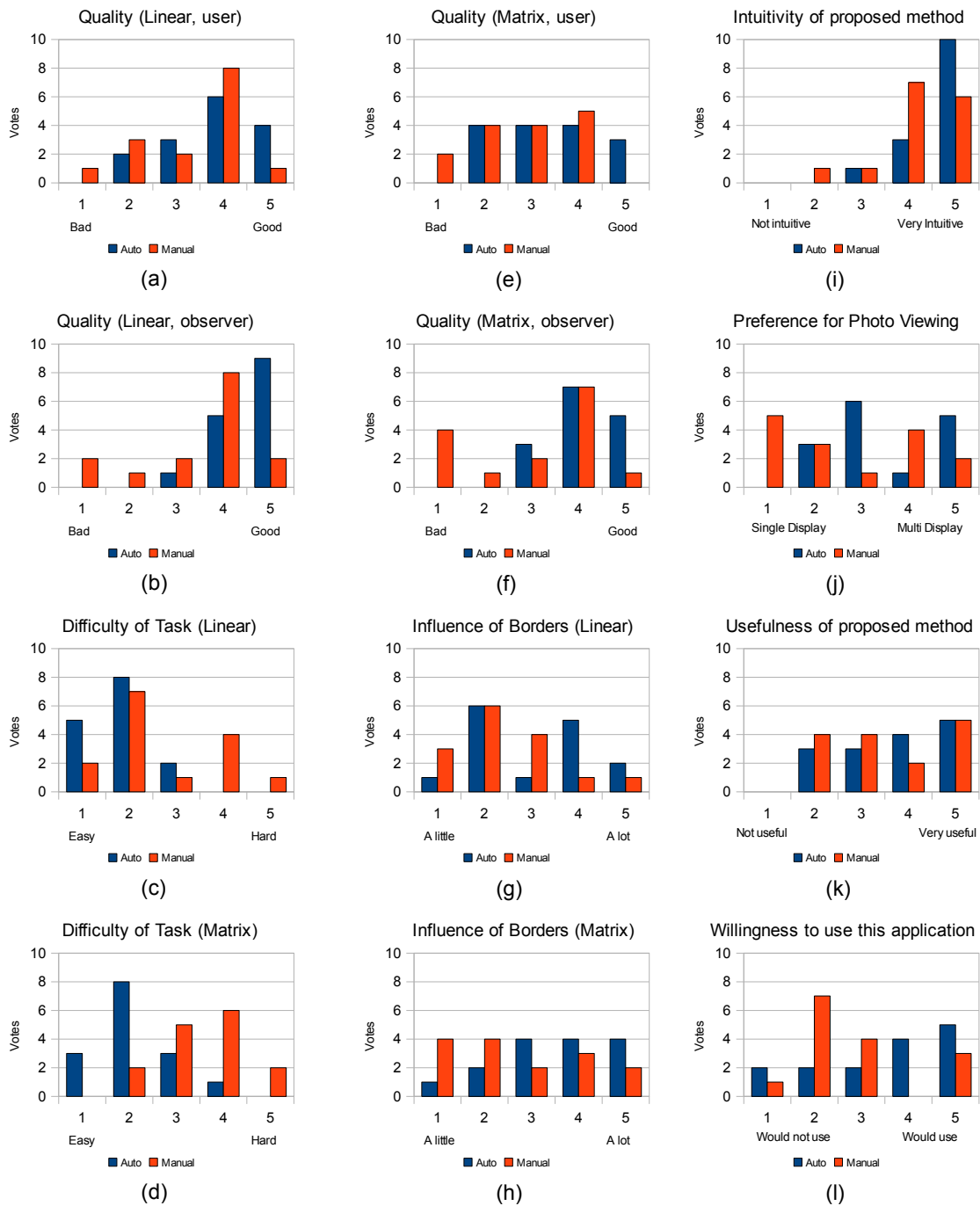
From the user study, we have found out that the multi-device multi-touch is an intuitive interaction metaphor for the ad-hoc multi-display use case. The automatic pose estimation technique can assist the user to calibrate multiple mobile displays in a relatively short time. The calibrated screens contain no disturbing visual error. To demonstrate the flexibility of our calibration method, we have presented a collaborating car racing game.

For future work, it would be interesting to explore more innovative applications of such displays. Especially since current mobile devices are equipped with a range of sensors such as cameras, touch screens and tilt or acceleration sensors this allows for novel input metaphors and applications. While our work represents a robust basis for these applications to explore, still more work is needed, for example in the area of continuously tracking the devices, if they were to

be moved around for a more interactive and even dynamic display.

## References

- [CCF\*00] CHEN Y., CLARK D. W., FINKELSTEIN A., HOUSEL T. C., LI K.: Automatic alignment of high-resolution multi-projector display using an un-calibrated camera. In *VIS '00: Proceedings of the conference on Visualization '00* (Los Alamitos, CA, USA, 2000), IEEE Computer Society Press, pp. 125–130.
- [CSWL02] CHEN H., SUKTHANKAR R., WALLACE G., LI K.: Scalable alignment of large-format multi-projector displays using camera homography trees. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 339–346.
- [HRG\*04] HINCKLEY K., RAMOS G., GUIMBRETIERE F., BAUDISCH P., SMITH M.: Stitching: pen gestures that span multiple displays. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces* (New York, NY, USA, 2004), ACM, pp. 23–31.
- [KB99] KATO H., BILLINGHURST M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality* (Washington, DC, USA, 1999), IEEE Computer Society, p. 85.
- [LPR\*09] LYONS K., PERING T., ROSARIO B., SUD S., WANT R.: *Multi-display Composition: Supporting Display Sharing for Collocated Mobile Devices*, vol. 5726/2009. Springer Berlin / Heidelberg, 18 Aug. 2009, pp. 758–771.
- [Met82] METHA N.: *A Flexible Machine Interface*. Master's thesis, Department of Electrical Engineering, University of Toronto, 1982.
- [MKM07] MERRILL D., KALANITHI J., MAES P.: Siftables: towards sensor network user interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction* (New York, NY, USA, 2007), ACM, pp. 75–78.
- [NR83] NAKATANI L. H., ROHRLICH J. A.: Soft machines: A philosophy of user-computer interface design. In *CHI '83: Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (New York, NY, USA, 1983), ACM, pp. 19–23.
- [RBY\*99] RASKAR R., BROWN M. S., YANG R., CHEN W.-C., WELCH G., TOWLES H., SEALES B., FUCHS H.: Multi-projector displays using camera-based registration. In *VIS '99: Proceedings of the conference on Visualization '99* (Los Alamitos, CA, USA, 1999), IEEE Computer Society Press, pp. 161–168.
- [Rek98] REKIMOTO J.: Matrix: A realtime object identification and registration method for augmented reality. In *APCHI '98: Proceedings of the Third Asian Pacific Computer and Human Interaction* (Washington, DC, USA, 1998), IEEE Computer Society, p. 63.
- [TPMT\*01] TANDLER P., PRANTE T., MULLER-TOMFELDE C., STREITZ N. A., STEINMETZ R.: Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In *UIST* (2001), pp. 11–20.
- [Wel91] WELLNER P.: The DigitalDesk calculator: tangible manipulation on a desk top display. In *UIST* (1991), Rhyne J. R., (Ed.), ACM, pp. 27–33.
- [Wes99] WESTERMAN W.: *Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-Touch Surface*. PhD thesis, University of Delaware, 1999.



**Figure 8:** Here we show the results of the user study interviews. In charts (a)-(d) the users judged quality of the calibration and the difficulty of the task. (e) and (f) show the respective opinion of the experiment observer. Furthermore, (g) and (h) show the perceived influence of the device borders. Finally (i) through (l) show the results of some general questions testing the acceptance of the method with the user.