Binding and registration of multiple displays

Alina Meixl
Universität Passau
Lehrstuhl für Informatik mit Schwerpunkt Eingebettete Systeme
Innstr. 43
94032 Passau, Germany
meixlali@fim.uni-passau.de

ABSTRACT

Today many people have devices with little displays such as smartphones, tablets or smartwatches. These devices provide possibilities to connect them and use them as one big display. Multiple displays can be used to enlarge contents or to exchange and copy data in an easy and natural way. These possibilities make the idea of multiple displays an interesting topic. Different options have been presented for this purpose. To create multiple displays, the devices must first be connected to each other. This process is called binding or pairing. If the technology is required in an environment where movability is important, information about the position of devices must be exchanged to share content. This is called local or spatial registration. In this paper we present a comprehensive overview of the state of the art in the field of mobile device binding and spatial registration. Furthermore we present advantages and disadvantages of the individual techniques and compare them.

Keywords

binding, registration, multiple displays

1. INTRODUCTION

2015 already 52.8 % of Germans [32] and 28.1 % of people worldwide own a smartphone, upward tendency [5]. Mobile devices are becoming more integrated into everyday life. They make life easier in many ways. For example anyone can get any information on the Internet at any time, take pictures and view them immediately, and even do work on these devices.

However, mobile devices have still some disadvantages. As an everyday device they can not be very large, because they have to fit into the pockets or bags. Watching photos for example, can be restricted, details can only be made visible by zooming. In addition, only a few people can watch the content at the same time. For example if a user wants to show some holiday photos to his friends, not everybody might see the pictures, as they could be too small on the device. If a user has been working on a smart device, he may not want to waste time for copying the edited files at the office. Again, fast transmission capabilities could make life easier even more.

Hinckley et al. call this the "spontaneous device sharing problem" [14]. There have already been introduced some solutions for easily connecting devices - especially their displays - and sharing content between them. For example this can be accomplished by gesture-driven technologies such as pinching (see chapter 2.1.5) or stitching (see chapter 2.1.4).

Connecting smart devices is also known as binding, device association, pairing, bonding or coupling [3, 4]. After the connection is established, the devices can share their position with other ones, so that content can be displayed or transferred optimally. This action is called local registration. In this paper we describe various techniques for both binding and registration of multiple displays.

2. BINDING TECHNIQUES

Rashid et al. define binding "as a way of coupling two devices by explicitly or implicitly creating a software plus network connection between them" [29]. There a many different ways to show a device that it should connect to another one. For this, both devices have to contain the special software and they have to be in any network, mostly Wi-Fi or Bluetooth. In the following a selection of various binding techniques are introduced and explained. We also want to discuss the advantages and disadvantages of every technique and compare them in the end especially for their usability, scalability and movability.

2.1 Binding by gestures

Many binding techniques use different gestures for pairing devices. Gestures are a very intuitive way to connect them [17]. In the following different binding techniques by gestures are introduced (see figure 1).

In this chapter, we will only describe binding techniques which include gestures which are directly executed on or with the device.

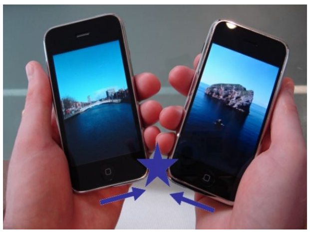
2.1.1 Shaking

Mayrhofer and Gellersen demonstrated the idea of coupling two mobile phones while holding and shaking them simultaneously [21] as well as Holmquist et al. who implemented "Smart-It Friends", some small devices that get connected when a user holds them together and shakes them [15].

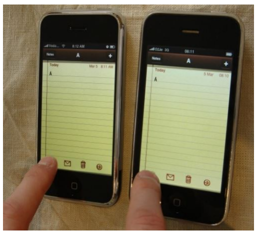
This movement is measured with acceleration sensors and sent as a broadcast message to the other devices for com-

- Alina Meixl is a master's student at the University of Passau, Germany

- This research report was written for *Advances in Embedded Interactive Systems* (2014), Volume 2, Issue 4 (October 2014). ISSN: 2198-9494

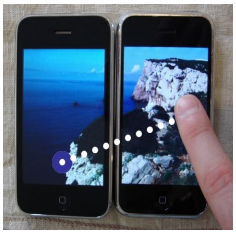


(a) Bumping [29]



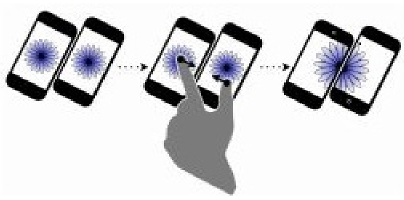
(b) Simultaneous Button Pressing [29]

Figure 1: Device binding by different gestures (continued on the next page)



(c) Stitching [29]

(d) Touching [29]



(e) Pinching [25]

(f) Shaking [29]

Figure 1: Device binding by different gestures (continued)

parison [15]. Both recommend shaking two devices with one hand as using two different hands could cause big differences in the acceleration data. Mayrhofer and Gellersen also present two different methods for the following message exchange. The first one is called ShaVe (shaking for verification). Devices exchange messages containing the acceleration data and apply a similarity measure (between the own and received data) and threshold afterwards. The second technique is called ShaCK ((shaking to construct a key). Here the devices "exchange variants of acceleration feature vectors and then use the percentage of matches found as a similarity measure" [21]. All different techniques end with establishing a connection between the shaken devices if there is an accordance of the data.

Advantages.

Mayrhofer and Gellersen sum up some advantages of shaking. First it is an intuitive and natural gesture because users are normally familiar with the gesture of shaking, so it doesn't require learning and can be performed without having to think.

They also say, that the gesture is vigorous and distinctive as the acceleration is measured over a longer period than the one of other gestures as bumping. This allows a better separation from other motion patterns.

Lastly they say, that shaking movements are very variable as it is a free gesture that normally is never two times the same. So this should allow a good detection of pairwise shaken devices without many false positives of randomly similar movements [21].

Disadvantages.

If a user wants to pair two devices and holds one in every hand it can cause problems with the motion detection, because the accordance of the movements can be too small to recognize it as a shaking gesture. So this gesture is only possible to perform with devices which can be hold in one hand or at least in two hands at the same time. Thus it is not possible to carry out this technique between static devices.

The devices also need to have a motion detector, which not every one might have. Another disadvantage is the scalability, as it could be a problem to connect more than two devices. The number of the devices that can be hold by a user is depending on the size of the hands and also on the size of the device itself. The more devices a user wants to connect, the more uncomfortable the technique is.

2.1.2 *Bumping*

In general bumping means the gesture of striking two devices together as clinking together glasses for a toast with just one corner (see figure 1a) or the whole edges [13, 29].

Hinckley presented some different connection possibilities trough bumping with tablets [13]. The first possibility is to use one tablet as a base and the other one as the connecting device. The second gets bumped to the base tablet. An acceleration sensor can detect a vibration by the bumping motion. This data is shared via a wireless network and compared. If there is an accordance a connection is established. So the devices can differentiate which edges were bumped and how the data should get shared. The connection remains as long as the devices touch each other. If one is moved, the devices will be disconnected and return to their previous state.

If two users want to connect their devices while holding them, they can also bump just the corners together, as this is the more intuitive motion while holding tablets in two hands [13].

Two users facing another can also connect their devices trough bumping the upper edges of their tablets. So they can for example share the working screen and both add changes. Therefore bumping allows display tiling, as well as sharing and pasting information and establishing face-to-face collaborations. This should also be possible for more than one device, for example for a 2x2 tiled display or also for a 1x3 tiled display, where the bumping can be performed by just one device by the domino effect. Every tablet then gets information about where it is relatively to the others. Bumping should not just be possible for tablets but for all handy devices with screens and rectangular shape [13].

Advantages.

Compared to just detecting all near devices, a hierarchy is created by selecting the devices. In addition, the edges for the division are specified. Another advantage is that the movement is very quick and natural, because it is a synchronous gesture like shaking hands, that everyone knows from everyday life [13, 29].

On the first glance the scalability seems to be a problem, as you can normally only connect two devices at a time. In a video, published by Hinckley in accordance to his paper, he mentions the idea of using the "domino-effect" to connect more than one device in a row by bumping one to another causes also a bump between the other one and a third one and so on [1]. This means, that all devices which should get connected are bumped from one side to the other. But this is just possible if all devices are lying in one row. Hinckley also shows in his video that he found a way for coupling devices on other arrangements, for example 2x2. Therefore the devices have to be added one by one.

Another advantage is, that the pairing is possible for all devices which can be bumped and have the required sensors.

Disadvantages.

As bumping means to poke one device to another, users might get unsure about how hard they have to bump them together and maybe some also have inhibitions because of this gesture and breaking something [29]. However this should not be a problem, as the hardware of the devices is designed to survive this handling. Therefore the threshold has to be set very small. It must be considered that such a small value could cause many false positives [13].

There is also the question how bumping could work with not rectangular shaped devices, such as smart-watches for example. This could prevent the advantage of the specified edges, as there are none.

Another issue occurs when many devices are added one to another. Users may get a problem if they have to remove one which is lying in the middle of the others. This could break the whole multiple display.

2.1.3 Simultaneous Button Pressing

Rekimoto et al. introduced an interaction technique called SyncTap [31]. This method is about making a network connection between two devices. A user can do this by pressing one button on each of two devices simultaneously. As a reaction to this the devices will send UDP packets to the network as a multicast containing the information about the button pressing times and the IP address. Everyone in the network will get this message and if the timestamp is the same as the own one a connection to the IP address can be established.

Rekimoto also tells about the possibility to do the same packet exchange with detecting synchronous sensor values such as sound. A user can knock with one device against the other one which causes the same captured sound for comparison [31].

Advantages.

This binding technique is not limited to devices with touch screens. It can be executed to every device which can have a programmed button interaction for this special pairing. This also means that a device has not to be handy.

Another point is that pressing buttons is a very natural gesture with a high affordance. So the user does not have to learn any new gesture for this technique.

Disadvantages.

A user normally has to use both hands to press the buttons simultaneously. This also means that he needs to focus on the interaction with two different screens, what might be a problem for an inexperienced user, especially if the buttons of the two devices are not directly side by side or have different haptics. This leads to another disadvantage. If a user needs both hands for two devices, he can just bind two at a time.

Another problem is, that a user study showed that users do perceive the simultaneous pressing of two buttons as awkward and uncomfortable as the effort to attain synchronicity was too high [29].

2.1.4 Stitching

"A stitching gesture is a [...] gesture that spans multiple displays, consisting of a continuous [...] motion that starts on one device, skips over the bezel of the screen, and ends on the screen of another device" [14] (see figure 1c). It uses the geometrical information from a pen or a finger and timestamps to automatically determine the spatial relationship between two devices.

Hinckley et al. created a prototype for this binding technique on pen-operated mobile devices[14]. They use a server which gets the stitching information from the participants' devices and sends a stitching event to both devices containing each others network address in case of matching pen traces. Because there is a short time where the pen is not touching any of the screens while it is over the devices frame they "define an *envelope* as the time interval during which the pen is in range of the screen and is moving at a speed above a predetermined threshold" [14]. To decide whether the movement was a stitching gesture some criteria have to be satisfied:

1. The envelopes have to end/start near the screens' borders and last longer than a given timespan.

2. The pause between the two envelopes is allowed to be 1.5 seconds as a maximum. This supports stitching between devices within a range of 75 cm as a maximum.

3. The direction of the pen while exiting the first screen and entering the other one must match within plus/minus 20 degree.

Hinckley also talks about the idea of cooperative stitching, where a user performs the first part of the gesture and the other one finishes it on his own device. So no one has to touch another one's device. It could be also possible that many users finish the gesture and for example everyone gets the shared file. Stitching of multiple devices is also possible to a maximum of 16 devices [14].

Advantages.

As there are two values which are compared, the timestamps and also the geometrical properties, there might be not as much collisions with other gestures as with other gestures-based techniques.

Another advantage is that for this technique no direct touching of the devices is needed, as it might be a taboo in some cultures. The binding is supported to a distance of an arm length and even longer for the idea of cooperative stitching [14].

[1] https://kenhinckley.wordpress.com/category/papers-with-youtube-videos/page/4/, accessed on 20.12.2015

Similar to the bumping technique a hierarchy is created by the direction of the gesture and also the edges for the division are specified.

Disadvantages.
The first problem is that this technique is just executable for devices with touch-screens.

It might also be a problem to detect the gestures if someone else is touching the display at the same time.

With this technique users worry about the security [29, 14]. Hinckley et al. think that this is not a problem, because the physical nature of the gesture does not allow any user to violate the rules, as those users would be noticed because of the small range. They also introduce the idea to let the user decide who should be able to pair the devices via stitching in an untrustworthy environment by passwords for example or just forbidding connections to unknown devices [14]. However as long as users think that the technique is insecure, they might not use it.

2.1.5 Pinching

Pinching normally means doing a simultaneous swiping gesture with the thumb and forefinger on two juxtaposed devices for connecting them as shown in figure 1e [25, 23]. But there are also some other ways to do the pinching. For example a user can use the forefingers of both hands [19, 23] instead of one hand. There is also the possibility of doing a two-step pinching created by Nielson et al. where "the user slides his index finger to the edge of one device and then afterwards to the edge of the other device" [23]. Especially the last shows that the pinching gesture is like two successive or simultaneous stitching gestures in two directions.

The setup works similar to the simultaneous button pressing, but with another gesture. First the devices have to be connected for example via Wi-Fi or Bluetooth [25]. If any user does a swipe gesture (see figure 1e) on a device, the device will send a message to all other connected devices. If a device gets a message with swiping information and also just sent its own information it can compare the content of the received message to the own and derive if there was a pinching gesture. For this some conditions have to be satisfied:

1. First, the timestamps are compared. They show, whether the gestures were performed simultaneously.
2. Then the devices check if the screens surfaces are directed to the same orientation.
3. Finally a check is made whether the movements were opposed.

If these three conditions are satisfied, the device will deduce that the identified swiping motions belong to a pinching gesture [25].

Advantages.
A good thing is that for pinching no extra sensors are needed [25]. Users can also arrange their devices in many ways and use devices of different sizes, for example tablets and smartphones. There is also an extensive range of mobile devices supported [25].

Similar to the bumping and Stitching techniques the edges for the division are specified by the gesture.

Disadvantages.
This technique is just executable for devices with touch screens. So the amount of usable devices is restricted.

Another problem is, that people might have a problem with letting other people handle their phone as a study showed . They were afraid of others damaging their device [19].

2.1.6 Touching

Touching can be understood in two different ways. The first is that touching means that one device touches another one for pairing. The other one would be the usage of the human body as a conductor to transfer electrical signals between devices, also called "intrabody communication" [37, 4] as shown in figure 1d. This means that a user touches two devices with his hands and this results in a connection.

The first variant is introduced by Lucero et al. They present "EasyGroups" which is a group binding method that allows collocated people easily to form a group, start an application and define the order of the devices around the table [20]. For this Bluetooth should be enabled on all devices and the application needs to be pre-installed. One user can start the application and touch the device he wants to connect with his own device. It will send connectivity information to the new group member over Bluetooth. The new device can now also start the application, connect to the WLAN network and join the group.

A very similar concept is used for the "Touch & Connect" technique of Seewoonauth et al., where an RFID tag is used to store the Bluetooth MAC address of the corresponding device. When a user touches this tag with his own device a spontaneous connection is established without a device discovery process. Both devices use this Bluetooth link to exchange data [35].

VISTouch is a technology introduced by Yasumoto and Teraoka. They put a smartphone in a special case with protuberances. If the phone touches another device, this will cause a connection [36].

The second variant is introduced by Park et al. [26]. Here the pairing between two devices is also done by touching them, but the electric signals for the data exchange are transported trough the human body.

Advantages.
Pairing devices by touching is very easy to perform for any user, as there is nothing else to do than touching the devices.

Touching between devices is also a less powerful contact then the bumping motion. So users should not have any concerns about any damages.

A user study also showed that the intrabody communication is very easy for users and also very fascinating [29].

Disadvantages.
The touching of two devices for the intrabody signalling technique can be a problem for bigger devices like tablets. People with small hands may not be able to perform this. There is also the problem that a user might touch two devices or two devices might touch each other but no connection between them is wanted. An additional authorizing of the pairing could destroy the simplicity of the technique.

A study revealed another disadvantage. Some users had concerns that such a technique may be too insecure [29].

2.2 Binding trough sounds

Devices can't just get paired by performing gestures but also by using special sound recognition. There are some different ways to do this which are described in the following.

2.2.1 Binding trough the Doppler effect

DopLink uses a well-known physical phenomena for pairing the devices - the Doppler effect. It "characterizes the change in observed frequency of a sound wave as a source moves towards or away from the receiver" [1]. When a user wants to connect to another device he presses a button to initiate an inaudible tone. Then he makes a pointing gesture towards the target. Because of the change of the velocity of the sound, a Doppler shift can be detected by all devices in the vicinity. The target device will receive the maximum frequency shift compared to other possible devices. All devices will sense a frequency shift and report it to the server. If we want to combine the device to multiple other devices, the server organizes the devices in a sequence based on their sound arrival times. Then it sends each participant the position relative to other devices.

There is also another technique that uses the Doppler effect. In this case the user has just to do a wave gesture in the air from one device to another one. "The hand movement reflects the ultrasound, causing a shift in frequency" [2].

Advantages.

This technique doesn't need any additional hardware for the devices.

Especially the second variant has the advantage that the user has not to touch the devices directly. This is something some people do not like as we saw in chapter 2.1.5.

Disadvantages.

There might be the possibility that other very loud sounds drown the connection sounds. Other sounds ,occurring in everyday life, could also cause an undesired connection especially if the sounds are inaudible for human.

2.2.2 Binding by the sound of gestures performed on a shared surface

SurfaceLink is a system where users can make natural surface gestures to control association and information transfer among a set of devices placed on a mutually shared surface (e.g. a table) [6]. A user can for example do some of the already mentioned gestures like pinching or stitching but not on the device itself but on the surface between the devices. Also other motions like clockwise gestures are possible and can be used to connect devices which are arranged in a round shape. To figure out the relative positions of devices in a 2-dimensional space, SurfaceLink combines stereo positioning with user gesture data.

Advantages.

This technique is easy to perform for the user and doesn't need any additional hardware for the devices. It also supports using some of the gestures mentioned above, e.g. the pinching gesture.

Another advantage is that the user has not to touch the devices directly, what some people don't like as already mentioned in chapter 2.2.1.

Disadvantages.

As already mentioned in chapter 2.2.1 there might be the possibility that other very loud sounds drown the connection sounds or maybe the sound can not be evaluated if more than one person is doing sounds on the surface. Another disadvantage is that an additional surface is needed and the texture of the surface should be comfortable for the user and also cause the needed sounds. This is what makes this technique not very usable in the context of mobility, because such a surface is not always available.

2.3 Binding by visible markers

Another way to pair devices is the usage of codes. Here a unique code is created on the display of the device. This code has to be detected in any way of the network environment. There are different possibilities for the code creation and recognition. These are shown in the following.

2.3.1 Binding through 2D matrix codes

One way of connecting different devices in a network is using 2D matrix codes. One example for this is the HuddleLamp which is a lamp that contains a camera in the lampshade [28]. This is connected to an additional PC which has the role of the server. To connect a device to the network there is no need to pre-install additional software. The user just has to scan an QR-Code which starts a web application that creates a code for the device to join the huddle or access the site directly. The camera in the lamp will recognize the code on the new device if you put it into the lamp's view and add the device to the network, this is called web-based pairing.

Schmitz et al. also use these codes for pairing devices. Every client renders a unique marker, then the user has to take a photo of the entire setup with the host device. This photo is then used to detect all markers, and returns the global coordinates and orientation of each marker [33]. The host uses this information to compute the viewports and send them to each client.

Advantages.

The HuddleLamp allows any user to join the huddle adhoc [28].

An advantage of the second variant is that no additional sensors are needed.

There is also one security advantage, because no information has to be shared between the clients directly, just between a device and the server.

Disadvantages.

For the construction of the HuddleLamp additional hardware is needed. The camera in the lamp and also an extra server machine. The space of the application is also determined by the range of the camera and the size of the lamp. So the user can move the devices freely inside this given area but not generally free in space.

There is also no possibility to do multi-touch gestures (e.g. pinch-to-zoom) on more than one device. Fingers still have to be on the same device for detecting the gesture and if more than one user touches the screens it could cause problems.

The second variant has also the disadvantage that the calibration might fail due to bright reflections obstructing the client's displays visibility. [33].

2.3.2 *Binding through an ID, encoded by color transitions*

The "phone as a pixel"-system consists of a target image, a collection of client display devices, a web server and a camera [34]. The name means that a phone is used to display one pixel of a large image, but it can also display more than one. Each client is first navigated to a web page containing a JavaScript application. This is for controlling all further client activities. Once the client has received a unique ID from the server, it flashes a color sequence on its screen, which encodes the ID (see figure 2).

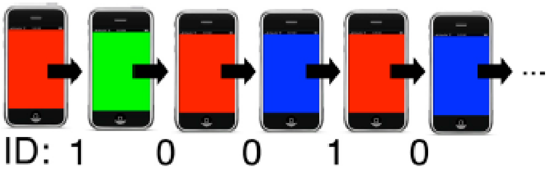


Figure 2: Encoding the device's ID using color transitions. Special color changes stand for "1" others for "0"[34]

The camera tracks the flashing from each display and determines IDs for all devices simultaneously, along with the camera coordinates. Each device receives a color value or a region of a larger image trough the web server after it finishes displaying the ID. After this the flashing ID sequence ends and the output is displayed.

Advantages.
The number of the displayed pixel is variable. One device can contain only one pixel but also many of them.

New clients can join the setup ad-hoc. They just have to be in the range of the camera and need to start the pairing process by adding the website.

Disadvantages.
Each device has first to do the flashing of a whole sequence to get detected by the camera. This requires some time.

Another disadvantage is the additional camera for the setup. Just as with the HuddleLamp (see 2.3.1) the space is limited by the range of the camera.

3. BINDING TECHNIQUES IN COMPARISON

In the last section we already discussed the advantages and disadvantages of the different binding techniques. Now the different techniques are compared in order to decide which are the most promising. Therefore we collected all information about the different techniques in table 1. The user study of Rashid and Quigley [29] and the survey of Chong et al. [4] provided most of the information.

In general we can say that a binding technique without additional equipment is more flexible as it can be done everywhere at any time. But it also depends on the different situation the binding should be performed in. Binding with additional instrumentation as for example a camera showed that there are other possibilities for sharing information as you do not need more than your own device necessarily.

Cardinality at a time.

Technique	Cardinality at a time	Mobility	Additional equipment	Practicability	Scalability	Movability after binding
Pinching	pair	yes	none	easy	yes	both
Stitching	pair/ group	yes	none	easy	lim (16)	both
Shaking	pair	yes	none	easy	n/a	dyn
Button pressing	pair	yes	none	hard	no	dyn
Bumping	pair/ group	yes	none	easy	lim	both
Touching	pair	yes	none	easy	no	stat
Touching (devices)	pair	yes	(RFID)	easy	yes	dyn
Doppler effect	group	yes	none	easy	yes	n/a
Surface sounds	group	lim	surface	easy	yes	dyn
2D matrix codes	group	var	(camera)	easy	yes	dyn
Color transitions	group	no	camera	easy	yes	stat

Table 1: Summary of the characteristics of the different binding methods (lim=limited, var=various, x = unlimited, dyn=dynamic, stat = static), information in brackets is just needed in special variants of the technique

This means how many devices can be paired at a time. As we can see most of the gesture-based techniques allow only binding two devices at a time. For stitching and bumping exist new ideas for increasing the number. Non-gesture-based techniques do normally allow an unlimited number of connections at a time, they are just limited to some physical issues, like for example the range of the camera or the surface.

Mobility & Additional equipment.
Mobility depicts if the techniques are applicable in the context of smart devices. That means that a user should be able to apply it everywhere at any time. As we can see most of the techniques provide mobility. The ones that don't provide this are mostly restricted because they do need additional equipment which is not available everywhere. The 2D matrix code technique can also use the camera of one of the devices, that's why it is variable.

Practicability.
Practicability is a combination of how easy the gesture is for the user and how big the accuracy is. Rashid and Quigley compared some of the gesture techniques (bumping, stitching, shaking, touching and simultaneous button pressing) in a user study and detected that shaking and touching were the easiest techniques to perform for the users, while simultaneous button pressing was the most difficult one as they had problems to do it simultaneously [29].

Scalability.
Scalability describes whether devices can be added easily to a multiple display or not. The value in the table shows

if it is possible to add any number of new devices to the environment. Sometimes this is possible but the number is limited.

Movability after binding.
Here the question is whether individual devices can be moved after the binding process without being disconnected. Some techniques provide static ways, as well as dynamic variants (both).

4. REGISTRATION TECHNIQUES

If content like pictures is shared between multiple displays the devices must mutually share their position to be able to display the right content. For this purpose their local position has to be registered and sometimes the rotation as well. There is the model of six degrees of freedom (DoF) which shows the number of independent ways by which a dynamic system can move, without violating any constraint imposed on it (see figure 3).

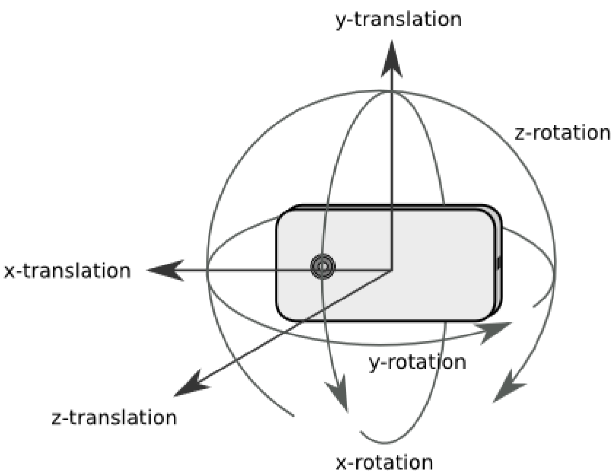


Figure 3: Six degrees of freedom: independent ways by which a dynamic system can move. 3 degrees are for translation on and 3 for rotation around the x-, y-, and z-axes.[2]

Depending on the design of the multiple display environment we need various DoFs for displaying the right content. On a predictable surface we only need two DoF: x- and y-translation. There are also some cases where we use three of the degrees (x- and y-translation and z-rotation) for a 2D environment if the devices are for example lying on a table (cf. HuddleLamp [28]).

To ensure the reception and handling of this position information about a device and especially the position relative to the other devices, both external sensors, such as cameras are used, as well as internal sensors, such as an accelerator or the built-in microphone. A selection of common technologies is described below.

4.1 External sensors

To register the position of each device in the network we can use external sensors, for example an additional external camera as explained in the following.

[2]http://vidhance.com/technology/six-degrees/, accessed on 05.12.2015

4.1.1 Position registration trough an external camera

The position of a device and its position relative to other devices can be detected by an external camera in a 2D as well as in a 3D environment.

HuddleLamp is a lamp with a camera in the lampshade [28]. When a device is paired with the server it gets an internal ID for detecting the position. This enables the vision system to track the devices movements over time. The server knows the position of a device relative to the tracked area and can send the information what to display. Registration by the light codes works similar to that. After the pairing with the code every device gets an ID to calculate and send the display information [28]. HuddleLamp uses a web-based architecture and JavaScript API. This is called a web-based tracking technology which proves to be a good method for registration trough cameras [24].

Advantages.
When a table is used as a base for the area and the devices lie on it, the position detection can be handled as a 2D system. This means that we do not have to consider all six DoF. This may result in less computational effort. As the position of each device is recognized by a camera there is no need for the individual devices to exchange messages to each other. The position is just known by the server and there is no need for evaluation of position data of the device itself.

Disadvantages.
For the HuddleLamp there is a small but noticeable delay between the movement of a screen and the reaction of the UI. The need of connection, synchronization and rendering performances of browsers is responsible for that. So the computational effort is still too high. As the position of the device is just noticed by the camera and the assigned ID of the device, it can happen, that a device gets lost. Because there is no packet exchange about position information between server and clients the device has to be removed and connected again.

4.2 Internal sensors

Instead of using external sensors to detect the position of a device we can also use the already built in sensors. The different possibilities are explained in the following.

4.2.1 Position registration by the internal camera

Many devices are equipped with a camera on the back and a front camera.

Schmitz et al. use the camera on the back for the automatic calibration [33]. Here the camera is used to take a photo of the arrangement of other devices which render matrix codes for detecting them. These codes are based on the idea of Rekimoto (Matrix codes for six DoF tracking) and contain information about the position of a device in a 3-dimensional space [30, 33].

Li and Kobbelt use the front camera to do the local registration for a device in the environment [18]. For this a matrix code [30] is positioned on the range of the devices' cameras. Then a marker detection algorithm is used to calculate their position in comparison to the marker and then between each client device and the server device. After that the latter can use this information to calculate the display information for every client.

Advantages.

A good thing is, that no additional equipment is needed for both variants.

As both variants use the marker detection, the techniques should also be doable for 3D environments, since the marker detection provides also 3D information of devices [30].

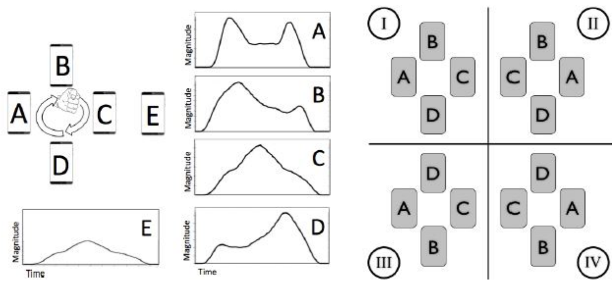
Disadvantages.

The second variant has the disadvantage that the devices can not be moved after the calibration. If they are moved, the procedure has to be repeated.

Another problem is, that one device has to be used to take the picture, so the user has to to the calibration manually on a second step.

4.2.2 Position registration trough an internal microphone

For internal sensors there are many possibilities, for example the internal microphone. SurfaceLink uses this sensor to provide the system with a much better understanding of the device arrangement. It combines stereo positioning (1-dimensional) with data of user gestures to detect the relative positions of devices in a 2-dimensional space [6]. For two-dimensional registration the user has to do a gesture on the surface. Then the timestamp of the audio peaks for each device informs about the order of devices (figure 4b). This could cause more than one possible arrangement, as shown in figure 4b.



(a) Real arrangement of the devices

(b) Possible arrangements without stereo positioning

Figure 4: This picture shows the necessity of stereo positioning [6]

That's what the stereo positioning is for. It is carried out by sending two non-audible tones of different frequencies, one from the right and from the left speaker. Other devices can thus determine whether the sending device is on the left or on the right by the heights of the observed amplitudes. This would show, that the right arrangement for the devices is the number I in figure 4b. However these algorithms don't calculate the exact distance to other devices but their position.

"Tracko" also uses the internal microphone for 3D tracking [16]. Jin et al. developed an algorithm based on the idea of round trip times which can calculate the position of devices in a 3D environment by sending and receiving inaudible sounds to and from the other devices.

Advantages.

Registration by sound doesn't need any additional hardware.

As we can see the technique is very flexible. With the corresponding algorithm every dimension detection can be accomplished (1D-3D).

Furthermore the sounds are non audible, so they do not disturb any humans.

Disadvantages.

There might be the possibility that other very loud sounds drown the connection sounds. The first variation has also the problem, that the sound could be not evaluable if more than one person is doing sounds on the surface. Another disadvantage of the second variant is that the microphones could be covered by hands. This disturbs the sound detection.

4.2.3 Position registration by multi-touch and acceleration data

The VISTouch system requires that the touching device is put in a special case which has twelve protuberances, three on each side of the case with different distances between any two protuberances. That makes each site unique. When a device inside the case touches another device providing a multi-touch function, position data can be exchanged. The relative positions of the devices can be calculated "by using the information of the spatial positions and triaxial (or biaxial) angles" [36]. First the system decides which side of the touching device is in contact with the other device's display by the distances and difference distances between the three protuberances. This information is sent to the touching device which calculates the third angle by the received information and the internal acceleration sensor. Finally, the system sends the information back to the tablet.

Advantages.

Because the data only contains coordinate information, the system can achieve a high calculation precision of the positions with a small calculation load.

Five degrees of freedom are provided for the touching device (except y-translation, y-rotation just in a 90 degree angle). This means, that the system can recognize multiple devices positions in real space (3D) as long as they are touching each other.

Disadvantages.

One problem is, that the devices have to touch each other the whole time.

Another thing is the necessity of a special case with protuberances for the touching device. This makes the system not easy to use spontaneously.

The whole side of a device has to touch the other one. That allows rotating the device around the y-axis only in a 90 degree angle.

5. REGISTRATION TECHNIQUES IN COMPARISON

In the section above we already discussed the advantages and disadvantages of the different spatial registration techniques, while in this section we compare them in order to decide which are the most promising.

We presented four ways of spatial registration for multiple displays. Most of them are just used to track the position of a device on a two dimensional surface but they also provide registration in 3D.

Technique	DoF (max.)	Fast technology	Number of traceable devices	Power consumption	Additional extensions
External camera	6	no	lim	low	camera
Internal camera	6	yes	x	high	2D matrix code*
Internal microphone	6	yes	x	high	-
Display and acceleration data	5	yes	x	high	special case

Table 2: Summary of the characteristics of registration methods by different sensors (x= unlimited, lim = limited, *there are ideas for implementations without matrix codes)

Degrees of freedom.

All sensors can be used to detect the position of a device in a 3D environment. For every sensor there is a technology which provides this registration.

Speed.

The position tracking of the technology used for HuddleLamp is complicated as many factors have to be considered. This causes a noticeable delay between the movement of screens and the reaction of the UI. The other techniques do not have to deal with this problem, as there is no detour via an additional device for the transfer of the data.

Number of detectable devices.

All techniques do not have a maximum number of detectable devices but they have maximum ranges through the limited size of the detectable area for example.

Power consumption.

The technology with the smallest power consumption is definitely the one which causes no extra effort to the devices itself. The technologies which use device internal sensors will have a higher power consumption than the registration via an external camera, as they always have to listen to other devices data and send their own.

Altogether we can say that there is no perfect technology. It all depends on the requirements of the system.

6. CONCLUSION

Altogether we can say that there are many ideas for the binding and registration of multiple displays. However there are still not many of these techniques used in real life. Even though the existing binding and registration methods work technically fine, the developers have to think more about security issues. For example what if the binding is not wanted wanted anymore even though it was requested before. What if there is a picture in the middle of a gallery not everyone should see? Or are the protocols used for the data exchange really secure? Developers have to focus on the security of the technologies as for example Mayerhofer and Gellersen did [21] for the bumping technique. Rashid and Quigley detected in a user study that the ease of use, security, promptness, appeal, originality and reliability are the key factors for using these technologies [29]. As we have seen above there is also the need of perfecting the existing technologies as there are still many disadvantages for the named key factors. There are also some other challenges which have to be considered [27, 9, 11, 7, 8]. One thing is the perceptional challenge of display switching. People might have problems with "angular coverage, content coordination, input directness and input-display correspondence" [27]. Those problems can occur because of varying display resolutions, luminance, visual interference and color or contrast differences. New technologies for multiple displays could include to set all changeable values automatically to the same. Social challenges like privacy can also be a big issue for users, similar to Augmented Reality applications [10, 12]. As we already mentioned, it is anomalous to touch another person in some cultures and also people didn't like it if their device was touched by a stranger. These issues can probably be resolved through the presented technologies as they might remove the necessity for another person's input (e.g. cooperative stitching [14]). Finally, the quality of registration techniques might have an influence on the user experience [22].

7. REFERENCES

[1] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel. Doplink: Using the doppler effect for multi-device interaction. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 583–586. ACM, 2013.

[2] K.-Y. Chen, D. Ashbrook, M. Goel, S.-H. Lee, and S. Patel. Airlink: Sharing files between multiple devices using in-air gestures. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 565–569, New York, NY, USA, 2014. ACM.

[3] M. K. Chong and H. Gellersen. Usability classification for spontaneous device association. *Personal and Ubiquitous Computing*, 16(1):77–89, 2012.

[4] M. K. Chong, R. Mayrhofer, and H. Gellersen. A survey of user interaction for spontaneous device association. *ACM Computing Surveys (CSUR)*, 47(1):8, 2014.

[5] eMarkete. Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2018 (in Milliarden)., 2015. `http://de.statista.com/statistik/daten/studie/309656/umfrage/prognose-zur-anzahl-der-smartphone-nutzer-weltweit/`, accessed on 15. November 2015.

[6] M. Goel, B. Lee, M. T. Islam Aumi, S. Patel, G. Borriello, S. Hibino, and B. Begole. Surfacelink: Using inertial and acoustic sensing to enable multi-device interaction on a surface. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 1387–1396, New York, NY, USA, 2014. ACM.

[7] J. Grubert, R. Grasset, and G. Reitmayr. Exploring the design of hybrid interfaces for augmented posters

in public spaces. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pages 238–246. ACM, 2012.

[8] J. Grubert, M. Heinisch, A. Quigley, and D. Schmalstieg. Multifi: Multi fidelity interaction with displays on and around the body. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3933–3942. ACM, 2015.

[9] J. Grubert, M. Kranz, and A. Quigley. Design and technology challenges for body proximate display ecosystems. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 951–954. ACM, 2015.

[10] J. Grubert, A. Morrison, H. Munz, and G. Reitmayr. Playing it real: magic lens and static peephole interfaces for games in a public space. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 231–240. ACM, 2012.

[11] J. Grubert, M. Pahud, R. Grasset, D. Schmalstieg, and H. Seichter. The utility of magic lens interfaces on handheld devices for touristic map navigation. *Pervasive and Mobile Computing*, 18:88–103, 2015.

[12] J. Grubert and D. Schmalstieg. Playing it real again: a repeated evaluation of magic lens and static peephole interfaces in public space. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, pages 99–102. ACM, 2013.

[13] K. Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 149–158. ACM, 2003.

[14] 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*, AVI '04, pages 23–31, New York, NY, USA, 2004. ACM.

[15] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp 2001: Ubiquitous Computing*, pages 116–122. Springer, 2001.

[16] H. Jin, C. Holz, and K. Hornbæk. Tracko: Ad-hoc mobile 3d tracking using bluetooth low energy and inaudible signals for cross-device interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 147–156. ACM, 2015.

[17] C. Kray, D. Nesbitt, J. Dawson, and M. Rohs. User-defined gestures for connecting mobile phones, public displays, and tabletops. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pages 239–248. ACM, 2010.

[18] M. Li and L. Kobbelt. Dynamic tiling display: building an interactive display surface using multiple mobile devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, page 24. ACM, 2012.

[19] A. Lucero, J. Holopainen, and T. Jokela. Pass-them-around: Collaborative use of mobile phones for photo sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1787–1796, New York, NY, USA, 2011. ACM.

[20] A. Lucero, T. Jokela, A. Palin, V. Aaltonen, and J. Nikara. Easygroups: binding mobile devices for collaborative interactions. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 2189–2194. ACM, 2012.

[21] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *Mobile Computing, IEEE Transactions on*, 8(6):792–806, 2009.

[22] A. Mulloni, J. Grubert, H. Seichter, T. Langlotz, R. Grasset, G. Reitmayr, and D. Schmalstieg. Experiences with the impact of tracking technology in mobile augmented reality evaluations. In *MobileHCI 2012 Workshop MobiVis*, volume 2, 2012.

[23] H. S. Nielsen, M. P. Olsen, M. B. Skov, and J. Kjeldskov. Juxtapinch: Exploring multi-device interaction in collocated photo sharing. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, pages 183–192, New York, NY, USA, 2014. ACM.

[24] C. Oberhofer, J. Grubert, and G. Reitmayr. Natural feature tracking in javascript. 2012.

[25] T. Ohta and J. Tanaka. Pinch: an interface that relates applications on multiple touch-screen by pinchinggesture. In *Advances in Computer Entertainment*, pages 320–335. Springer, 2012.

[26] D. G. Park, J. K. Kim, J. B. Sung, J. H. Hwang, C. H. Hyung, and S. W. Kang. Tap: touch-and-play. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 677–680. ACM, 2006.

[27] A. Quigley and J. Grubert. Perceptual and social challenges in body proximate display ecosystems. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 1168–1174. ACM, 2015.

[28] 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*, ITS '14, pages 45–54, New York, NY, USA, 2014. ACM.

[29] U. Rashid and A. Quigley. Interaction techniques for binding smartphones: A desirability evaluation. In *Proceedings of the 1st International Conference on Human Centered Design: Held As Part of HCI International 2009*, HCD 09, pages 120–128, Berlin, Heidelberg, 2009. Springer-Verlag.

[30] J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific*, pages 63–68. IEEE, 1998.

[31] J. Rekimoto, Y. Ayatsuka, and M. Kohno. Synctap: An interaction technique for mobile networking. In *Human-Computer Interaction with Mobile Devices and Services*, pages 104–115. Springer, 2003.

[32] H. Schmidt. Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2015 (in Millionen)., 2015. http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/, accessed on 15. November 2015.

[33] A. Schmitz, M. Li, V. Schönefeld, and L. Kobbelt. Ad-hoc multi-displays for mobile interactive applications. In *31st Annual Conference of the European Association for Computer Graphics (Eurographics 2010)*, volume 29, page 8, 2010.

[34] J. Schwarz, D. Klionsky, C. Harrison, P. Dietz, and A. Wilson. Phone as a pixel: Enabling ad-hoc, large-scale displays using mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2235–2238, New York, NY, USA, 2012. ACM.

[35] K. Seewoonauth, E. Rukzio, R. Hardy, and P. Holleis. Touch & connect and touch & select: Interacting with a computer by touching it with a mobile phone. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 36:1–36:9, New York, NY, USA, 2009. ACM.

[36] M. Yasumoto and T. Teraoka. Vistouch: Dynamic three-dimensional connection between multiple mobile devices. In *Proceedings of the 6th Augmented Human International Conference*, AH '15, pages 89–92, New York, NY, USA, 2015. ACM.

[37] T. Zimmerman. Personal area networks: near-field intrabody communication. *IBM Systems Journal*, 35(3-4):609–617, 1996.