Interactively Set up a Multi-display of Mobile Devices

Peter Barth(✉) and Manuel Pras

Hochschule RheinMain, University of Applied Sciences,
Kurt-Schumacher-Ring 18, Wiesbaden 65197, Germany
peter.barth@hs-rm.de, mail@manuelpras.de

Abstract. We provide a method to interactively set up a multi-display using a combination of multi and single device gestures. An initial setup provides a coarse grained model. Test pictures and user judgement based on the human visual system then guide a fine grained interactive process. This allows the user to move and rotate single screens until differences between physical and model position are no longer perceived. To this end, a central computer holds the model and connects among all participating smartphones and tablets with different physical dimensions and display resolutions. In addition, it evaluates gestures and prepares as well as distributes images on the multi-display.

Keywords: Multi-display · Interaction · Smartphone · Visual system · Network graphics

1 Introduction

The sheer number of smartphones and tablets pushed into the market makes mobile devices available in large quantities. Each device has a high resolution display and provides touch interaction. However, each individual display is still comparatively small and typically supports interaction with one user only. With a comparatively small investment or being a group of people, many devices are available. These devices may be joined to provide one large display, a multi-display, for interactive applications or just the visualisation of single screens. To set up a flexible multi-display, we need to know the position of each individual device that constitutes the multi-display.

We propose an interactive method to quickly reconstruct the position of many devices that form a multi-display. Instead of relying on cameras and computer vision techniques [8], we employ the error detection capabilities of the human visual system until perceived accuracy satisfies the user. We can then display any image on the modelled multi-display. Typical applications include image and slide shows as depicted in Fig. 1, as well as tickers, games and videos [9]. These applications all depend on having a single canvas where each devices shows just a portion of one large image.

S. Sigg et al. (Eds.): MobiCASE 2015, LNICST 162, pp. 221–238, 2015.
DOI: 10.1007/978-3-319-29003-4_13

Fig. 1. Multi-display showing an image

2 Related Work

Most often multi-displays are found in fixed installations where the location and position of the often homogeneous displays are known. Systems range from the Nintendo DS over laptops with slide out screens such as the Gscreen Spacebook to expensive commercial multi-displays to be used in fairs and exhibitions consisting of many large monitors or even many projectors joined together [8]. While these setups offer an harmonic user experience, they require dedicated hardware and do not support spontaneous scenarios or reusing a multitude of different displays at hand.

Traditionally, forming multi-displays relies on computer vision techniques [2,3,5–7]. These techniques suffer from external conditions of the surrounding environment and can most often not be used in sunlight or other direct light from above. In contrast, they do not require interacting with the device and are thus less likely to change the position of the devices during setup.

Schmitz et al. [8] were first to realise that smartphones and tablets are well suited to support multi-displays. They propose to combine a collection of heterogeneous devices to form a single screen multi-display. Their main contribution is a calibration process relying on computer vision combined with manual fine tuning and supporting any possible distribution of the devices on a flat surface. They have already realised, that a manual calibration procedure using gestures is not only useful, but may be sufficient. Although they favour the automatic camera based calibration, they support skipping it altogether. They chose communication over Wi-Fi instead of Bluetooth to achieve acceptable latency and high bandwidth. As test image during manual calibration they use a checker board which has concentric circles as overlay. User tests indicated that centring

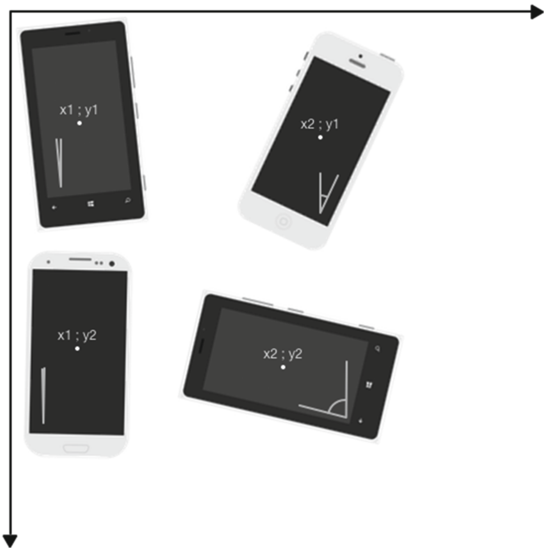

Fig. 2. Model of a multi-display with position and rotation angle

the test image on the device currently being adjusted helps the users to focus. They provide user evaluation setting up a multi-display consisting of four devices and report using up to seven mobile devices with displays using different resolutions.

3 Interactive Multi-Display Setup

We support any non-overlapping distribution of mobile devices on a flat surface. We distribute an image on the devices constituting the multi-display once the layout is known. Here, we concentrate on setting up the model. An accurate representation needs to know the exact position of each device as depicted in Fig. 2. For each device we need to know the physical width and height as well as the number of pixels in x and y direction of its screen, assuming classical portrait mode. Using any origin on the surface, it is sufficient to know the x and y coordinates of the centre of each display as well as its rotation angle. We opted for right rotation with 0 degrees for a device in upright portrait mode. We always use an origin such that all mobile devices are to the right and under the origin. Thus, each centre point has positive values in millimetres and rotation angles are between 0 and 360 degrees. We differentiate between the device model and one physical instance of it. Of the device model we know the physical dimensions and pixels. Of the physical instance we know its centre point position and rotation. Thus, to alter a model we adjust only three values per participating device instance.

Our main contribution is an interactive procedure to adjust the computer model of the position of the participating devices to the actual physical setup.

We rely exclusively on gestures and the human visual system as well as human judgement to achieve sufficient perceived accuracy. We differentiate between two modes of interaction. First, we compute a coarse grained model capturing alignment of the participating devices. We assume a row based layout, such that each device is a member of a row. With a simple series of multi-device gestures – swiping row-wise from left to right – we compute an initial model. We update the model each time the swiping finger leaves a device and refresh the test image. Then, we use a series of gestures on single devices to express movement or rotation of an individual device. We use these gestures to adjust the modelled position of this device. The user sees the result of the interaction with each currently touched device immediately. Therefore, the user can employ the human visual system to identify errors and cognitive abilities to improve the model.

3.1 Test Images

Users shall decide whether the model represents the actual setup based on test images shown on the multi-display. The test images used during both the coarse grained and fine grained calibration step serve two purposes: identify a device that needs to be adjust and correct the positional error of that device. To identify a device that needs to be adjusted, we need a global test image, that allows to spot any offset in position or rotation. We propose to use intersecting lines combined with coloured concentric circles such as in Fig. 3. The concentric circles help best to spot position offsets as even small errors are identified as oval by the visual system. In addition, the lines serve to identify rotation errors, which appear as bend. Test images are always shown on all devices and if necessary centred to the barycentre and scaled in advance to cover the entire multi-display. Furthermore, the central computer visualises the current model on its screen to mimic the setup. Thus, the user may compare again visually the current model to what is laid out on the surface as in Fig. 3. This is helpful to spot very large offsets.

In the correction phase the current position in the model of the individual devices is manually corrected. We propose an alternative test image as in Fig. 4. A collection of equidistant coloured lines is used, that intersect on the first touch point of the device to be adjusted. Thus, the user can concentrate on the device being positioned and perceives errors in its position and rotation angle more easily.

3.2 Coarse Grained Initial Position

For initial setup, we assume that all devices are members of a row, but the devices need not be aligned. For the gesture, the user swipes over all devices from left to right, row by row from top to bottom. For each device, we compute a straight line from the touch points using linear regression. Based on the entry and exit points on each device of a row, we can compute its absolute position as in Fig. 5. For this, we postulate that the user performed a straight line with

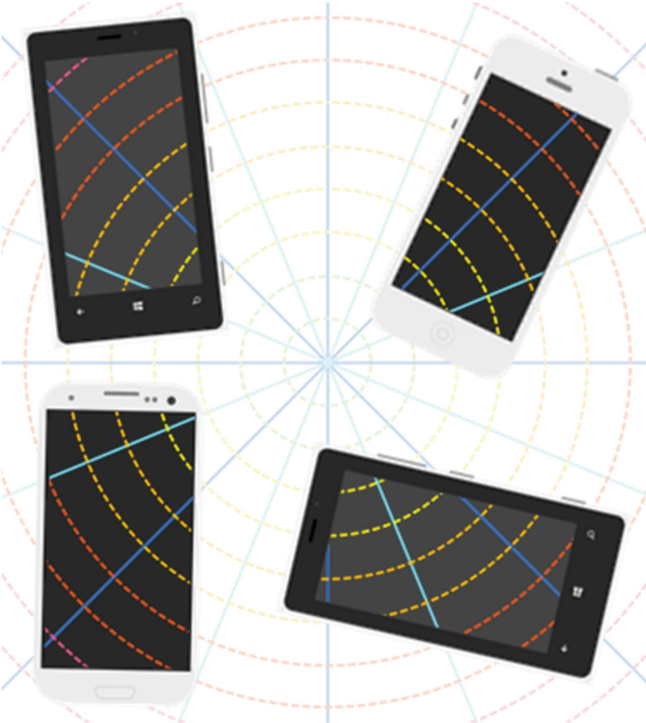

Fig. 3. Test image of coloured concentric circles and intersecting lines (Color figure online)

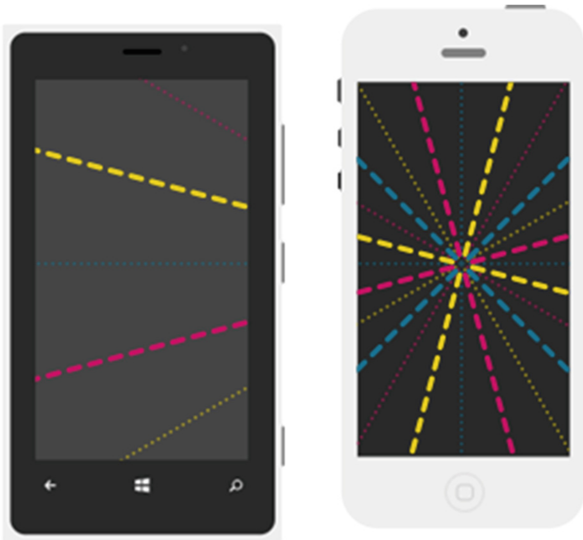

Fig. 4. Test image to adjust single device

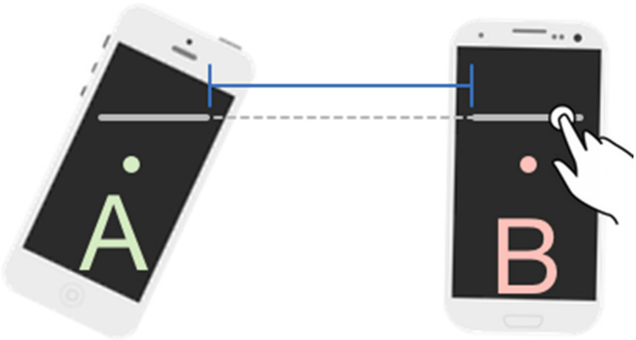

Fig. 5. Swipe across two devices

constant speed between each pair of devices. Although this assumption will not hold exactly, it gives us a near enough approximation and allows to build our initial model quickly. Thus, the devices are always in the correct relative position. Showing a test image allows to start the fine grained calibration on a good basis. The horizontal position is never off more than half the dimension of the device screen.

To guess the distance between exit point on one device and entry point on its right neighbour, we need to measure the duration between these two events. We compute time differences of each device against the central computer and thus adjust device times to an absolute time. This adjustment uses standard techniques [4] and is not visible to the user. It is performed before any multi-display calibration starts. Note, that during adjustment, we do not send images or other bulk binary data in order to keep the variance of the round trip time low. Typically, we experience maximal errors within a single digit millisecond range.

We need to detect when we reach the right end of a row and after that the next row starts. One option is to measure the (adjusted) absolute time between device exit and device entry. We only support initial layouts where neighbouring devices have no huge gaps in between, which is sensible for a multi-display. We set the maximal gap to 7 cm. This means that if two devices are more than 7 cm apart horizontally, we assume that the second device marks the start of the next row.

However, while this works in practice, the approach is not always adequate. An alternative is changing the direction at each row. Although that approach is very stable, it implicitly assumes that orientation of all devices is upright portrait mode. If a device is flipped, an incorrect row change may be detected. This may be compensated by using the compass.

With these two options we only have information about each individual row but no information about the distance between two rows. Assuming that the multi-display does not have huge gaps between rows, the left-most device of each row gets vertically aligned using the lower border of the previous row as in

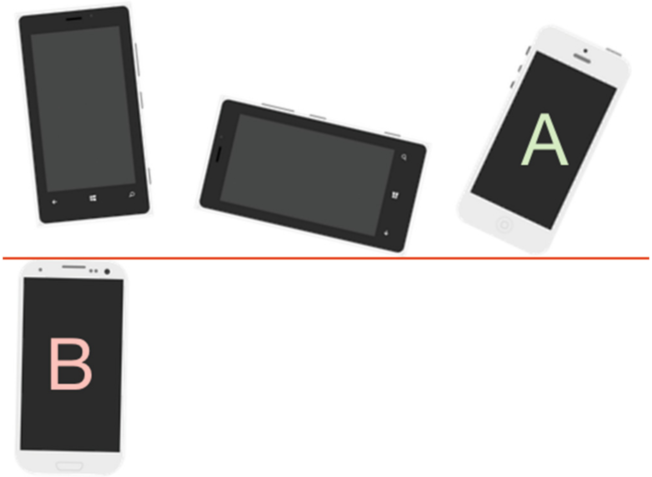

Fig. 6. Vertical alignment using row borders

Fig. 6. This means that the upper border of the left-most device equates with the lower border of the previous row. To accommodate for frames of the mobile devices, we can either assume a fixed average frame border or use dedicated device specific information. Note, that we assume the left-most devices to be left aligned in order to determine the horizontal position of the left-most device of each row. Again, we think that this is a plausible assumption for most multi-display setups.

We may relax that assumption and only require that there is a left-most column. Then we would add as final step a swipe on the left-most column of the devices from top to bottom as in Fig. 7. The computation is similar to the one we did per row. With the resulting distance between the centre points of all devices in vertical direction, the initial model can be build again.

3.3 Fine Grained Absolute Position

Given a model, we rely on the visual system of the user to spot errors and his or her cognitive abilities to manually improve the model. These two steps are repeated until the final model matches the physical setup with good enough perceived accuracy. To do so, the user can adjust the position and the rotation angle of each individual device. It is important, that the feedback of gestures happens immediately in order to employ the visual system of the user. Thus, we do not distribute images over the network. Instead, we compute the corrected image on the device that is being adjusted and show the correction effects locally as quickly as possible. Therefore, we do not suffer from lagging screen updates impairing the direct interaction cycle. We visualise a dedicated device centred test image as in Fig. 4 on the entire multi-display as soon as a manipulation gesture starts on a device. To save bandwidth, we currently only distribute the

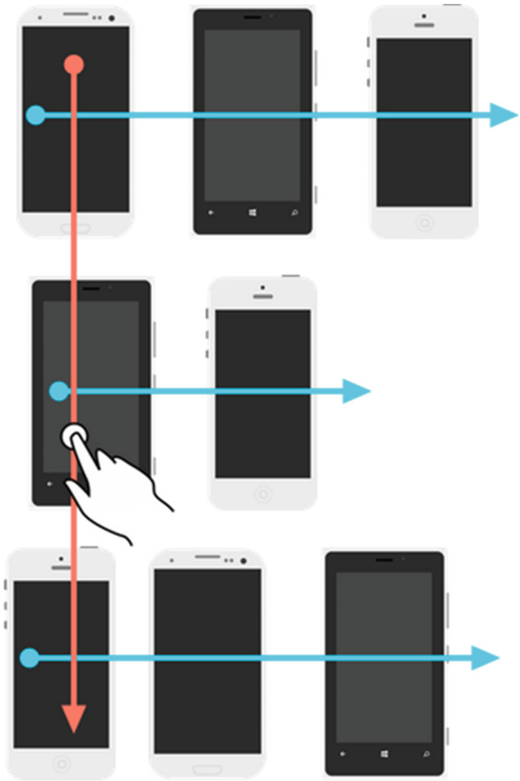

Fig. 7. Additional vertical swipe for row detection and alignment

image to the nearest four neighbours as they are the ones the visual system most likely uses as reference to detect errors. Typically, the test image with device centred lines on the four nearest neighbours allows users to spot and quantify offsets accurately.

We support changing position or rotation with different gestures as depicted in Fig. 8. We change position using a single touch point as in Fig. 8 left and rotation using two touch points as in Fig. 8 right.

To this end, we evaluate the gestures on the device and translate and rotate the image on the device. During a gesture, we need to cover the entire display with a single test image. Therefore, we use a test image that is 100 percent larger in each direction than the actual screen of the device. This allows to show complete images during all single stroke adjustments.

The translation gesture is straightforward. The user touches the device and moves the image around until the image fits the idealised image. After releasing the finger, we compute the offset and send the offset to the central computer. There, the model is updated and freshly generated parts of the test image are distributed. Note, that the model visualised on the central computer screen is

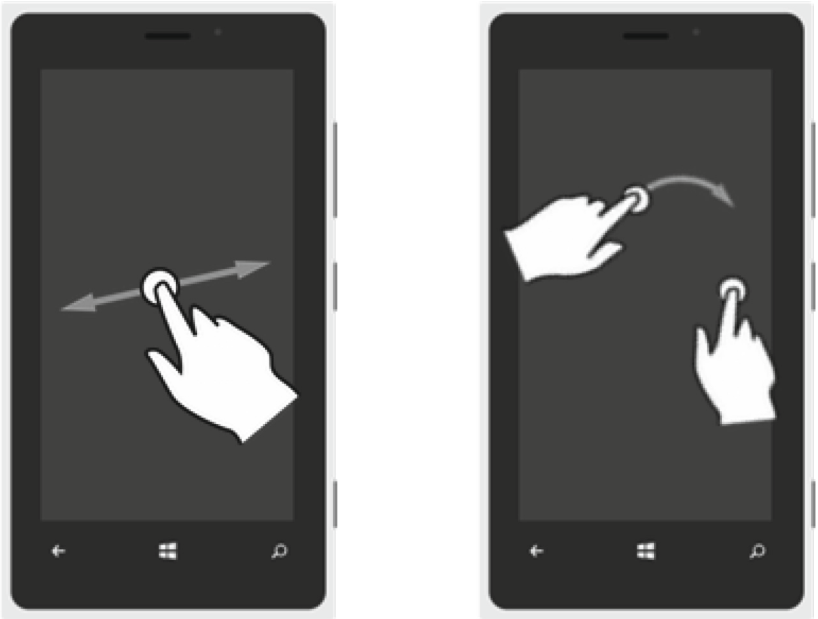

Fig. 8. Translation and rotation gesture

not updated during adjustment. This is not necessary. The user shall concentrate on the manipulated and nearest devices while having the finger on the device.

The rotation gesture is slightly more complex. We may use two fingers of different hands or two fingers of one hand as with a typical pinch gesture. To fluently switch between translation and rotation we suggest to use one finger of one hand to translate. To rotate, keep on finger on the screen and touch it with a finger of the other hand. Moving the second touch point while holding the first manipulates the rotation angle. The distance between the two fingers is used to calibrate the effect of rotation. Bringing them closer together allows for a more coarse grained rotation, while moving them apart allows for a finer grained rotation adjustment. We amplified this natural effect to allow for better control. The further apart the two fingers are, the less is the effect of the circular arc on the rotation angle.

3.4 Differentiate Among Gestures

One challenge is to differentiate among the different gestures. We have three different gestures in two phases. In the initial phase there is swiping entire devices. In the correction phase there is translation with one finger and rotation with two fingers. We differentiate between the translation and the rotation gesture using the current finger count. As long as there is only one finger touching the screen it is a translation gesture. As soon as another finger joins the device screen the gesture changes to a rotation gesture.

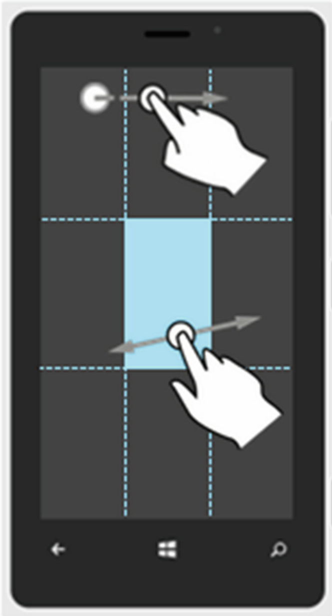

Fig. 9. Display sections to differentiate between initial and fine grained gestures

We may want to differentiate between the coarse grained initial phase and the fine grained correction phase. Note, that after the start of a fine grained phase there won't be a coarse grained phase again. This may only happen, if the user wants to start over because the device layout has changed dramatically, was wrong to start with, additional devices have joined, or the results are unsatisfactory. Based on the location of the first touch point (as shown in Fig. 9), we can reliably recognise coarse a grained setup during the entire gesture. Note, that we expect the user to swipe the complete screen of the device in the coarse grained phase. Thus, the user will start a swipe gesture on the border of the mobile device. Furthermore, the user will naturally – and if not by training – put a translation or rotation start point in the centre of the device to retain more degrees of freedom. Thus, we can partition the device screen into an inner and an outer area. We recognise a coarse grained calibration in the outer area and a fine grained calibration in the inner area. A working setup is to divide the display area into three equally sized parts in horizontal and vertical direction as depicted in Fig. 9, but that may be changed. Therefore, users can do a fine grained adjustment or start over by executing at any time any gestures they have in mind without being forced to use meta commands.

Fig. 10. Single steps for creating an image portion

4 Image Preparation and Distribution

The central computer holds the multi-display model and the image to be shown on the multi-display. The image will be shown on the bounding box of the individual displays of the multi-display. First, the bounding box and thus the individual display positions are scaled to fit the image. For example, we start with the left-most image in Fig. 10. Next, we need to construct the correctly sized and rotated image parts per individual display. To this end, we cut out rectangular pieces. We make sure that any later rotation will be covered by using the diameter as side length of the rectangular piece. This results in the second picture in Fig. 10. Next, we rotate the image by the screen's negated rotation angle to compensate for the device rotation in the physical setup. This gives the third picture in Fig. 10. Finally, we cut out the rectangular image and scale it to the resolution of the target device. We get the right-most picture in Fig. 10. This image is then sent to the device and there shown without any further processing.

We base our data communication on Blaubot [1], which provides distributing messages among a collection of mobile devices and potentially connected central machines. Blaubot supports both Bluetooth and Wi-Fi out of the box and the underlying transmission technology can be selected by configuration. During the production phase most often high resolution images are regularly distributed to all devices. Because of the high bandwidth requirements, we rely on Wi-Fi.

5 Evaluation

We evaluated the simplicity and accuracy of the proposed calibration method through user tests. With adequate test images an accuracy within a 1–2 mm error margin is routinely attainable by untrained users in 2 min on average.

5.1 Participants and Setup

We run the tests with six participants, two female and four male. All participants own a smartphone for at least two years and are between 21 and 29 years old.

The different tasks of the evaluation are performed using three different smartphone models running Android 4.0 or higher (HTC Desire, HTC Legend, and Motorola Milestone). The mobile devices were released between 2009 and 2010 and represent the low end of the currently running Android devices. While the screens of those devices have about the same dimension, they vary in resolution.

5.2 Tasks

The participants had to solve three different tasks, two of which were divided into two sub tasks. This gives a total of five sub tasks. As part of the first task the participants were asked to build a 2×3 device matrix and set up a multi-display as in Fig. 11 left. After the participants had finished the calibration, the multi-display showed a checker board pattern, which allowed them to rate the accuracy.

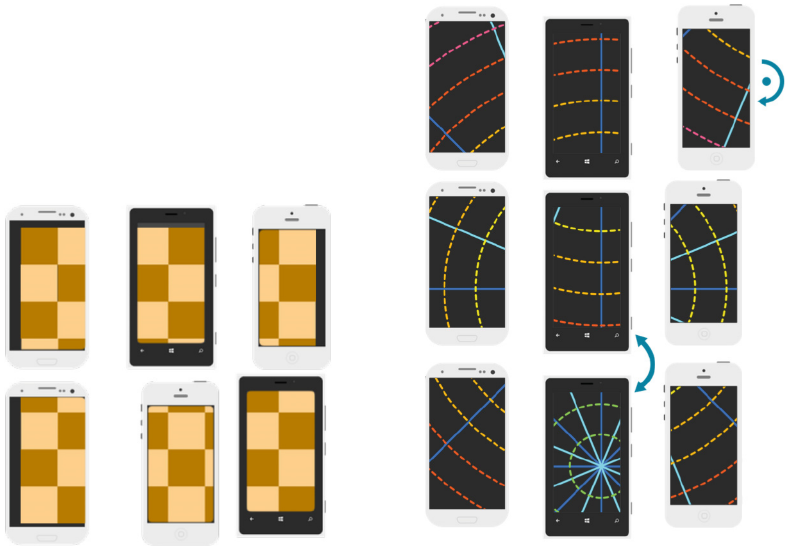

Fig. 11. 2×3 matrix for a checker board (left) and swapped and flipped devices (right)

The second task focused on the ability to detect and fix large errors between physical setup and the multi-display model using different test images. The coloured concentric circles image was compared with an alternative image, consisting of a coloured grid as in Fig. 12 left. In order to compare those two test images the second task was divided into two sub tasks. In both sub tasks the participants were asked to identify and fix two large errors in the multi-display as in Fig. 11 right. The first error consisted of a random swap of two devices in the setup. The second error was a 180° rotation of one of the remaining devices.

While the first sub task used the concentric circles as test image, the second sub task used the alternative grid image.

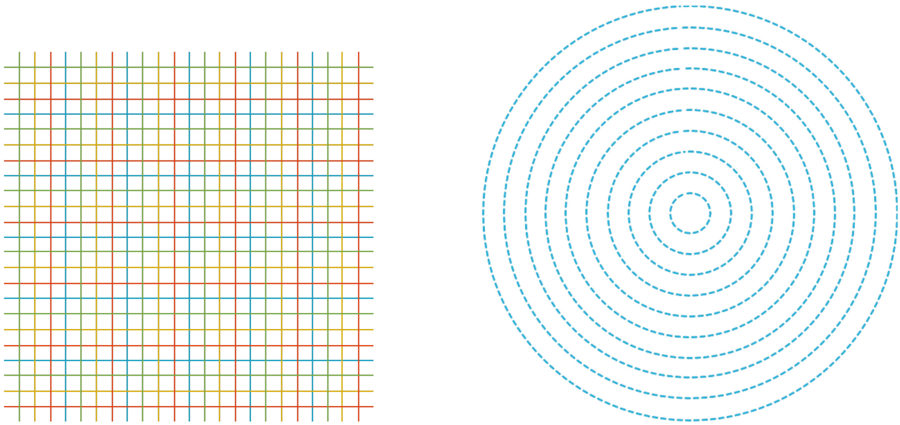

Fig. 12. The two test image alternatives used in the evaluation

The third task evaluated the adequacy of the test image for the fine grained calibration step. We compared equidistant coloured lines with monochrome concentric circles as in Fig. 12 right. The participants were asked to build a 2×2 matrix for displaying a picture showing buildings as in Fig. 13.

Due to its geometric patterns, this type of subject is suitable for judging the accuracy of the multi-display setup. Again, this task consisted of two sub tasks. One sub task used the equidistant coloured lines as test image, the other sub task used monochrome concentric circles.

5.3 Procedure

After filling in personal background information, the participants watched a tutorial video in order to learn the basic steps of the multi-display setup. To familiarise themselves with this process, they set up a 2×2 matrix under the supervision of an expert before processing the five sub tasks. Following each processed sub task the participants rated the difficulty of solving the task as well as the accuracy of the multi-display setup. The accuracy rating of the multi-display setup was done for all participants by one neutral observer. The rating was done on a Likert scale between one (least agreement) and four (most agreement).

In addition to those subjective ratings we were also interested in the measured offsets between the physical setup and the multi-display model. After the test was over, we measured the remaining offsets of the multi-display. We showed a generated test pattern consisting of horizontal and vertical lines with a distance of 1 cm as in Fig. 14. We then took a picture of the multi-display setup and used

Fig. 13. Final image shown in production phase

it to measure the offset using image processing tools between each neighbouring pair of devices.

5.4 Results

The evaluation of the first and the third task shows that the resulting model of the actual physical layout can be adjusted to satisfy users for most common visualisation tasks. To achieve such a result, it is crucial to use adequate test images for both the coarse as well as for the fine grained calibration phase. This claim is supported by the results of the second (Fig. 15) and the third task (Fig. 16). The test image consisting of equidistant coloured lines is best suited for

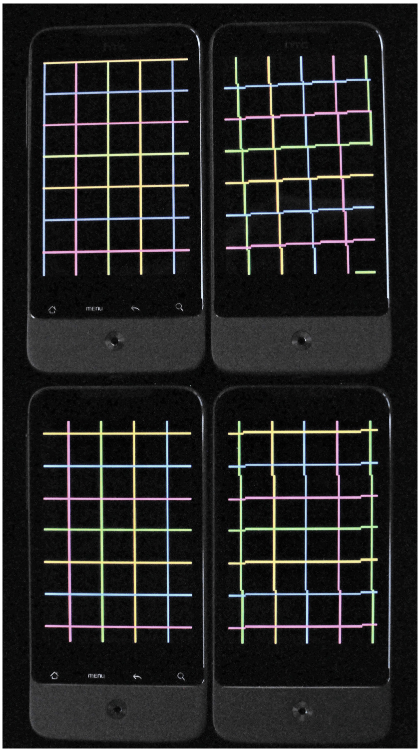

Fig. 14. Test pattern for measuring the offsets of the multi-display setups

the fine grained phase. The test image consisting of coloured concentric circles is best suited for the coarse grained calibration phase.

With the test image consisting of equidistant coloured lines all participants found the calibration of a multi-display to be easy or very easy (Fig. 17). In addition, they rated the perceived quality of the display setup to be accurate or very accurate (Fig. 16). With the monochrome concentric circles as test image the participants found calibration to be hard or very hard. They rated the quality of the display setup to be only 2–3. These results reflect the need of using adequate test images for an easy to use and accurate interactive setup of a multi-display.

As the results of the second task show this does not only apply for the fine grained calibration step but also for the coarse grained calibration step. With the test image consisting of coloured concentric circles the participants found the task of identifying the permutation and rotation of the devices to be very easy. But with the test image consisting of coloured grids the participants only identified and fixed half of the offsets (Fig. 15). In this case they found the task of identifying the offsets to be hard or very hard. Moreover the participants only identified and fixed half of the offsets in the multi-display setup compared to the test image consisting of coloured concentric circles.

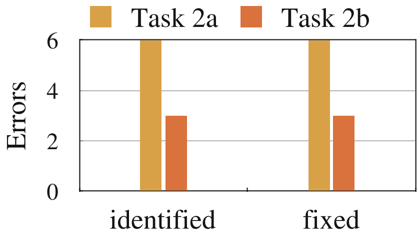

Fig. 15. Identified and fixed errors, second task (Color figure online)

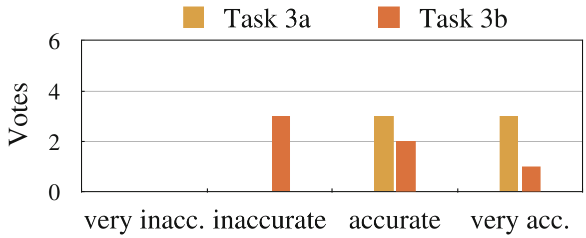

Fig. 16. Perceived accuracy, third task (Color figure online)

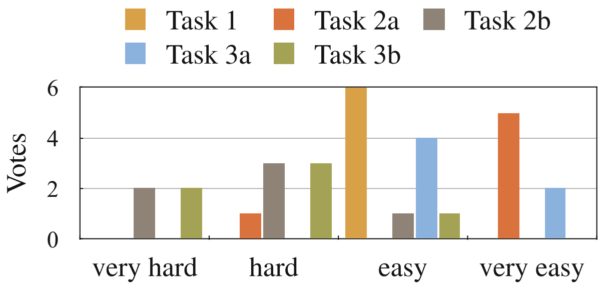

Fig. 17. Complexity of solving the five tasks (Color figure online)

In addition to evaluating perceived accuracy we also photographed the generated test pattern and measured accuracy of the multi-display model. The measured offsets between model and physical setup for each pair of individual devices were mostly between 1 and 2 mm. There have been few exceptions of up to 8 mm, depending on participant and task (Fig. 18). With the proposed test images the offsets turned out to be significantly smaller compared to the multi-displays which were set up using one of the alternative patterns.

The average processing time of the three tasks was 116 s as in Table 1. The individual processing time not only depends on the participant but also on the task as well as the test image used for setting up the multi-display. With the alternative test image the processing time was 15–50 % higher in comparison to the proposed test images.

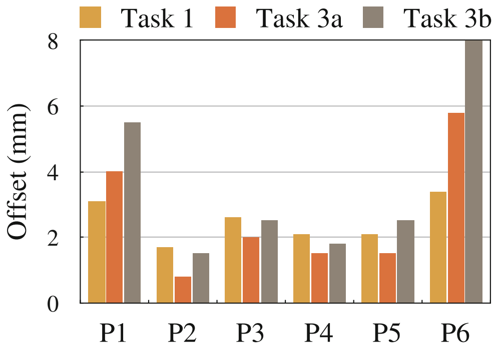

Fig. 18. Measured accuracy per participant (Color figure online)

Table 1. Average processing times in seconds

Task	1	2a	2b	3a	3b
min	55	43	63	68	54
Ø	102	107	164	98	113
max	142	300	256	181	241

Note, that the user may unintentionally move the device physically while applying the gestures. This may happen if the friction between device and surface is very small, which is not the case for typical office tables (medium-density fibreboard), or the users applies too much force. We have experienced these issues only during the coarse grained phase, if the devices are too far apart. If that happens, the users adjust the physical layout to the intended layout by moving the devices physically. After that the fine-grained gestures are applied, which is possible without unintentionally moving the devices physically.

6 Conclusion

The visual system of human users paired with their cognitive capabilities can be used to quickly build a multi-display consisting of heterogeneous mobile devices including smartphones and tablets. The resulting model of the actual physical layout on a flat surface can be adjusted to comply with enough perceived accuracy for most common visualisation tasks.

The main contribution is a two phased interactive method to first build a coarse grained model followed by a fine grained correction phase. Users do not need to leave the interaction mode and can continuously use any of the three offered gestures. This allows them to quickly set up a multi-display consisting of many devices and correct them accurately with dedicated test images and immediate feedback on their actions. The setup is limited to multi-displays that are at least in the initial phase almost row-based and left-aligned. In the opinion of the authors this covers most multi-display scenarios.

In the future, we plan to use additional sensors for setting up the multi-display. For example, magnetic sensors provide a compass which already gives at least coarse grained rotation information. In addition to touch points, we may use light sensors or approximation sensors instead of touch points for computing the initial layout. This might not be as accurate as the touch points, but it is contactless and thus prevents modifying the physical layout during the initial phase. In addition, we plan to enhance the framework to allow touch interactions on the set up multi-display. Thus, we use the multi-display not only for visualisation but for interaction as well.

References

1. Barth, P., Groß, H., Pras, M.: Blaubot middleware, March 2015. https://github.com/Blaubot
2. Chen, H., Sukthankar, R., Wallace, G., Li, K.: Scalable alignment of large-format multi-projector displays using camera homography trees. In: IEEE Visualization, pp. 339–346 (2002)
3. Chen, Y., Clark, D. W., Finkelstein, A., Housel, T.C., Li, K.: Automatic alignment of high-resolution multi-projector displays using an uncalibrated camera. In: IEEE Visualization, pp. 125–130, October 2000
4. Cristian, F.: Probabilistic clock synchronization. Distrib. Comput. **3**(3), 146–158 (1989)
5. Li, M., Kobbelt, L.: Dynamic tiling display: building an interactive display surface using multiple mobile devices. In: 11th International Conference on Mobile and Ubiquitous Multimedia, MUM 2012, pp. 24:1–24:4. ACM, New York (2012)
6. Rädle, R., Jetter, H.C., Marquardt, N., Reiterer, H., Rogers, Y.: 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 2014, pp. 45–54. ACM, New York (2014). http://doi.acm.org/10.1145/2669485.2669500
7. Raskar, R., Brown, M. S., Yang, R., Chen, W. C., Welch, G., Towles, H., Seales, W., Fuchs, H.: Multi-projector displays using camera-based registration. In: IEEE Visualization, pp. 161–168 (1999)
8. Schmitz, A., Li, M., Schönefeld, V., Kobbelt, L.: Ad-hoc multi-displays for mobile interactive applications. In: Eurographics, vol. 29, p. 8 (2010)
9. Shen, G., Li, Y., Zhang, Y.: Mobius: enable together-viewing video experience across two mobile devices. In: Knightly, E., Borriello, G., Cceres, R. (eds.) MobiSys, pp. 30–42. ACM (2007)