Projector-Leap Motion calibration for gestural interfaces

Pavel Najman   Jiri Zahradka   Pavel Zemcik

Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2, Brno
Czech Republic
{inajman, izahradka, zemcik}@fit.vutbr.cz

ABSTRACT

Calibration of a projector and a tracking device is an essential step for interaction with projected content using gestures. We propose a novel technique for calibration of a data projector and a Leap Motion sensor. Using the proposed approach, users can calibrate the system by touching few points on the screen plane and in the space above it. No printed patterns, reflective markers, or additional tools are needed. The calibration process involves two steps. In the first step, we collect finger positions which we then use in the second step to find the calibration matrix and projector position. We compared the accuracy and precision of the proposed method to the accuracy and precision of a capacitive touchscreen in a touch based interaction task. During the evaluation we measured the Euclidean distance between the displayed and touched points. The best average distance for our method was 1.23 mm which is comparable to 0.79 mm for touch screen. The experiments demonstrate that the technique is suitable for an interaction with user interface elements designed in the usual way.

Keywords

projector, Leap Motion, hand tracking, calibration, dynamic mapping, HCI

1 INTRODUCTION

Human computer interfaces research in the last years becomes more and more focused on 3D gestural interfaces. 3D gestural interaction is natural and powerful method of communication between humans and computers [10]. The popularity of the gestural interfaces is supported by increasing number of different body and hand tracking devices, such as Microsoft Kinect, Asus Xtion or Leap Motion Controller. These sensors are relatively cheap and therefore available for wide range of standard computer users. In comparison to traditional keyboard and mouse interfaces, the gestural interfaces bring novel interaction techniques, that might be beneficial for many applications but especially for games or applications on tabletop displays.

The tabletop display interfaces are often equipped with capacitive touchscreens, which are quite expensive due to their large size [15, 7]. An alternative to touchscreen is a combination of a data projector and a motion tracking sensor. This approach has several advantages over the touchscreen solution. The image might be projected on various surfaces with different shapes and reflection properties. Unlike touchscreens, these surfaces are usually more scratch and fingerprint resistant and can be cleaned more easily. We also have additional information about the types and poses of fingers. This information comes from a sensor that detects and tracks user's hands and fingers during interaction. The detected poses are transformed into the projector image space, which allows the projected image modification to provide a feedback to the user. This approach brings the necessity of calibration, in order to obtain the spatial transformation between the projector's image coordinates and the sensor's space coordinates.

In this paper we propose a calibration technique for a data projector and the Leap Motion Controller, which might be applied to the tabletop interfaces. A unique contribution of our approach is that the technique doesn't require printed patterns, reflective markers, or other additional tools. The effective range of the Leap Motion extends from approximately 25 to 600 millimetres and its field of view is about 150 degrees. The sensor utilizes two stereoscopic IR cameras and three IR light emitting diodes for fast and accurate hand tracking [11]. The high accuracy (up to 0.01 mm) and the low latency (less than 10 ms) are the main reasons why we chose to use Leap Motion sensor instead of other body tracking sensors that are available.

The further parts of this paper are organized as follows. The following Related Work briefly reviews the calibration techniques of a data projector with a standard camera, an infrared camera and depth sensors. The Proposed Method section defines a novel technique for the calibration of a data projector and the Leap Motion Controller. The section contains also a description of our experimental environment. The accuracy of the proposed method was evaluated with two experimental setups and compared to a capacitive touchscreen accuracy. All experiments and results are presented in Evaluation section. Finally, Conclusion section summarizes the paper contribution.

2 RELATED WORK

Projector-based user interfaces often utilize a standard camera as a simplest way to enable user interaction with the projected content. In these interfaces, the camera observes the projected image. The calibration of such systems is usually done semi-automatically by projection of structured light pattern over a printed pattern of a known size. The overall scene is captured by a camera [4, 3]. Although it is easy to find the calibration points, it is difficult to track hands or fingers because conventional skin color detection algorithms fail in the recognition of the hand overlapped by the projected image. Thus, the combination of a data projector and the standard camera is more suitable for structured-light scanners [17].

The problem with the image overlapping of the hands and fingers disappear, when the standard camera is replaced with an infrared (IR) camera. The visible light projected by the data projector is invisible for the IR camera and therefore the calibration point's measurement process must be different. The projected points might be marked by IR markers or user has to touch the points with the finger. Both approaches were presented in [19, 20], where the calibration was calculated just as a 2D homography between the camera and the projector image planes. The spatial relationship between a world and a projector coordinates was not required in these setups.

The above described difficulties might be overcome when the projector-based interface is equipped with a RGB-D devices such as the Asus Xtion[2] or the Microsoft Kinect. These devices are based on PrimeSense technology[8, 12], which combines an RGB camera with a depth sensor made using an IR camera and IR projector. A drawback of these devices is in their high latency and low depth accuracy, which has relative error between few millimeters and 40 mm [9]. The PrimeSense technology is used in combination with a pico-projector in the wearable interface for multi-touch applications on everyday surface called the OmniTouch[6]. The calibration of a data projector and Kinect is utilized in a tabletop multi-touch display system presented at [18]. This system is able not only to detect if the screen is being touched, but can also tell which finger was used to touch the surface.

Unlike the described previous works, our technique does not require printed patterns, reflective markers, or other additional tools. Also the hand tracking speed is far better, since we are using the Leap Motion controller. This controller has hand tracking latency less than 10 ms. For comparison, the Microsoft Kinect has latency around 60 ms [14].

3 PROPOSED METHOD

In this section, the proposed calibration method is presented. The calibration process involves two steps. The inputs to the first step are coordinates of points that will be projected onto the screen plane. The goal of the first step is to collect points in Leap Motion coordinate space that correspond to the projected points in screen coordinate space. The outputs of the first step are sets of points. Each set contains a point in screen coordinate space and several corresponding points in the Leap Motion coordinate space. These sets are then processed in the second step which computes the transformation matrix and projector position. The whole process is shown in Figure 1.

The first step begins with the projection of the calibration pattern, that contains at least three non-collinear points, onto the screen plane. The first pattern point is highlighted and the user positions his hand so that the tracked finger is touching the currently highlighted point on the screen surface. The position of the finger is captured and the same point is highlighted with different color. User then raises his hand from the surface and moves it towards the projector along the projection ray. During this motion the user keeps the projected point on his finger. When the hand movement stops, the spatial position of the finger is captured again. The height, in which the user stops above the screen plane, is not important and it can be different for each point. At this point, we can either highlight the same point with different color and continue by collecting more finger positions along the projection ray or highlight next pattern point and repeat the collection process.

We use the acceleration of finger motion to identify the appropriate moment for capturing the finger position. During the collection process the tracked finger moves either fast or slowly. The fast motion occurs when the user moves between two point positions and the slow motion usually means that the user stops to touch the highlighted point. To distinguish between the fast and slow motion, we utilize two thresholds. When the finger speed drops below the lower threshold we capture the current finger position. Afterwards we wait for the speed to raise above the upper threshold before we capture another point. The thresholds values reflect the registered speed of motion when the hand is still and when the hand moves between two points. Therefore, the lower threshold value depends to a large extent on

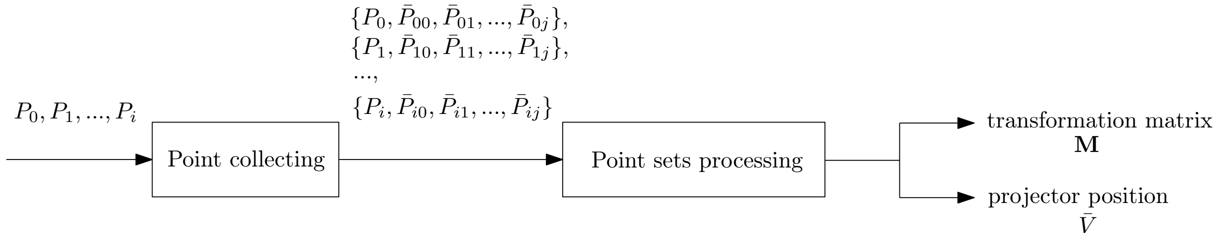

$P_0, P_1, ..., P_i$
Point collecting
$\{P_0, \bar{P}_{00}, \bar{P}_{01}, ..., \bar{P}_{0j}\},$
$\{P_1, \bar{P}_{10}, \bar{P}_{11}, ..., \bar{P}_{1j}\},$
$...,$
$\{P_i, \bar{P}_{i0}, \bar{P}_{i1}, ..., \bar{P}_{ij}\}$
Point sets processing
transformation matrix $\mathbf{M}$
projector position $\bar{V}$

Figure 1: Calibration process involves two steps. The inputs to the first step are at least three non-collinear points that are projected. In the first step, points in Leap Motion coordinate space, that correspond to the projected points, are collected. The collected points, together with the projected points, are processed in the second step. The outputs of the calibration process are the transformation matrix and the projector position.

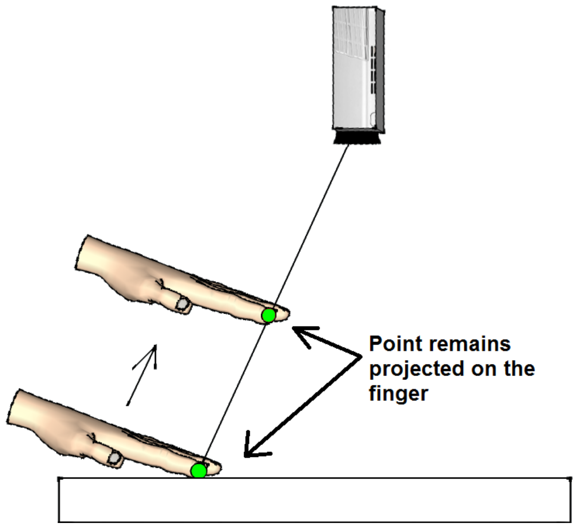


Figure 2: To capture the calibration points above the screen plane, the user moves his hand along the projection ray while keeping the projected point on his finger.

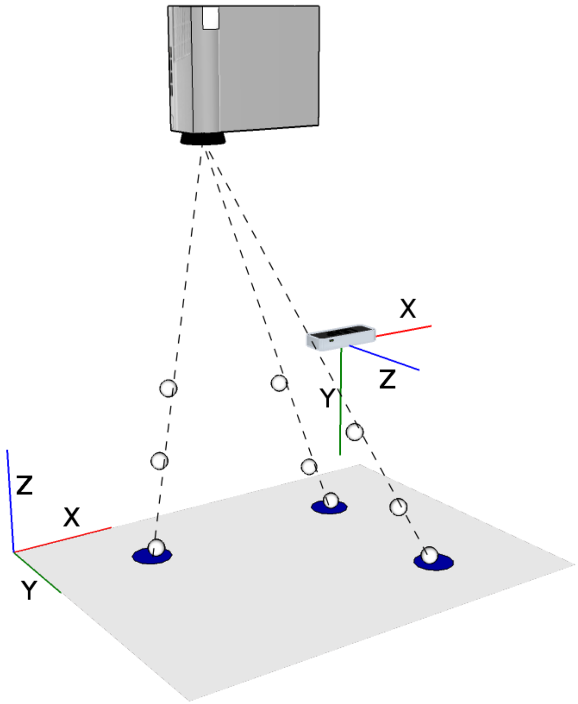


Figure 3: Input data measurements for proposed calibration technique. The white spheres represent collected finger positions in the Leap Motion coordinate space. The corresponding pattern points in the screen coordinate space are highlighted by blue circles.

the tracker noise and the natural hand tremor and the value of the upper threshold depends mainly on the distance between two pattern points. We use 1.5 mm/s and 75 mm/s as the values for lower and upper threshold respectively, which we found suitable for our environment. Alternatively, pressing of a button or timer based approach can be used for point capturing.

For each pattern point, we collect three finger positions, one on the screen surface and two above. The collected finger positions in Leap Motion coordinate space and corresponding point coordinates in projector screen space are processed in the next step. The situation is depicted in Figure 3.

In the second step, the projector position relative to Leap Motion and the transformation matrix is computed. The relationship between point $\bar{P} = [\bar{p}_x, \bar{p}_y, \bar{p}_z, 1]$ in the Leap Motion coordinate space and the corresponding point $P = [p_x, p_y, p_z, 1]$ in screen coordinate space is given by

$$P = \mathbf{M}\bar{P} \tag{1}$$

where $\mathbf{M}$ is the transformation matrix that can be further decomposed to

$$\mathbf{M} = \mathbf{SRT} \tag{2}$$

scale, rotation and translation matrices. There are eight parameters that need to be found in order to construct these matrices. These parameters are two scales s_x, s_y , three angles of rotation α, β, γ and three elements of a translation vector $\mathbf{t}$. We find their values by first computing their initial estimates which we then refine using Levenberg-Marquardt method [16].

To compute the initial estimates we need three non-collinear points P_0, P_1 and P_2 in screen coordinates and their counterparts $\bar{P}_{00}$, $\bar{P}_{10}$ and $\bar{P}_{20}$ in Leap Motion coordinate space captured on the screen plane. From these points two vectors in each coordinate space can be computed.

$$\mathbf{u} = P_1 - P_0 \qquad \bar{\mathbf{u}} = \bar{P}_{10} - \bar{P}_{00} \tag{3}$$

$$\mathbf{v} = P_2 - P_0 \qquad \bar{\mathbf{v}} = \bar{P}_{20} - \bar{P}_{00} \tag{4}$$

To construct the scale matrix $\mathbf{S}$ we need two scales s_x and s_y which represent the number of pixels per mm in x and y direction. Only one estimate

$$s = \frac{||\mathbf{u}||}{||\bar{\mathbf{u}}||} \tag{5}$$

can be used for both values because the difference between these two values is usually very small and will be compensated for in the following refinement. For the third scale s_z, we use fixed value of 1. By doing this the z coordinate of the point P will represent the height above the screen in mm.

For the rotation matrix $\mathbf{R_{xyz}}$, three angles are needed

$$\alpha = -\arctan 2\left(\frac{\bar{\mathbf{y}}\cdot\mathbf{z}}{\cos\beta}, \frac{\bar{\mathbf{z}}\cdot\mathbf{z}}{\cos\beta}\right) \tag{6}$$

$$\beta = -\arcsin\left(\bar{\mathbf{x}}\cdot\mathbf{z}\right) \tag{7}$$

$$\gamma = -\arctan 2\left(\frac{\bar{\mathbf{x}}\cdot\mathbf{y}}{\cos\beta}, \frac{\bar{\mathbf{x}}\cdot\mathbf{x}}{\cos\beta}\right) \tag{8}$$

which express the rotation around x, y and z axis respectively. Vectors $\mathbf{x} = [1,0,0,0]$, $\mathbf{y} = [0,1,0,0]$ and $\mathbf{z} = [0,0,1,0]$ are unit vectors that represent the orientations and the directions of the screen coordinate system. Vectors $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$ and $\bar{\mathbf{z}}$ are unit vectors that represent the same orientations and directions but in the Leap Motion coordinate space. Vector $\bar{\mathbf{z}}$ is the normal to the plane specified by $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$. Vector $\bar{\mathbf{x}}$ can be found by first finding the angle between $\mathbf{u}$ and $\mathbf{x}$ and then rotating the $\bar{\mathbf{u}}$ vector by the same angle around the $\bar{\mathbf{z}}$ axis. Vector $\bar{\mathbf{y}}$ can be found analogously.

The initial estimate for vector of translation is given by

$$\mathbf{t} = -\left(\bar{P}_{00} - \frac{p_{00x}}{s}\bar{\mathbf{x}} - \frac{p_{00y}}{s}\bar{\mathbf{y}}\right) \tag{9}$$

To refine the initial estimates, we minimize

$$E_1 = \sum_{n=0}^{i} ||P_n - \mathbf{M}\bar{P}_{n0}||^2 \tag{10}$$

using Levenberg-Marquardt method, as implemented in MPFIT [13] which contains a translation of MINPACK [16] algorithms to C. We use all the points $\bar{P}_i$ in the Leap Motion space that were captured on the screen plane together with their counterparts P_i in the screen space for this refinement.

To find the projector position, we first need to reconstruct the projection rays from the collected points. This can be done by fitting a straight lines through the points that were captured for the same pattern point in different heights above the screen plane. To fit the straight line $\bar{\mathbf{l}}$ through the points that were captured for the ith pattern point, we minimize

$$E_2 = \sum_{n=0}^{j} dist(\bar{P}_{in}, \bar{\mathbf{l}})^2 \tag{11}$$

the sum of square distances from these points to the line.

In an ideal situation, the projector would be positioned in the intersection of these lines. Because the lines are almost always skew with no intersections, we rather find the point of closest approach $\bar{V}$ for all lines. This point can be found by minimizing

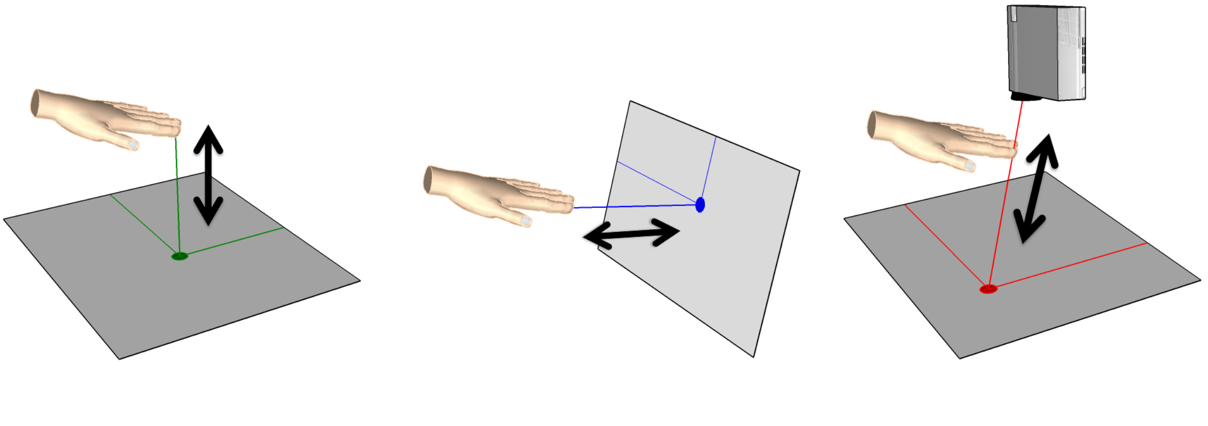
$$E_3 = \sum_{n=0}^{i} dist(\bar{V}, \bar{\mathbf{l_n}})^2 \tag{12}$$

the sum of square distances from the point to all the lines.

Several ways exist in which we can use the results of the calibration. We present three of them that could be the most common ones. They are touch based interaction, direct pointing interaction, and dynamic finger/hand projection mapping. The first two are interaction styles in which we need to find the point on the screen with which we are trying to interact. The last one uses the results of calibration to project an image on user's hand or fingers that can contain additional information.

For touch based interaction, the point of interest is found by simply multiplying the finger position with the calibration matrix. The result is a point with x and y coordinates that represent the position on the screen in px. The z coordinate represents the height of the finger above the screen in mm which can be used to find out if the finger is touching the screen or not. See Figure 4a.

For direct pointing interaction we need to transform not only the finger position but also the direction in which the finger is pointing. These two pieces of information specify a line that is then intersected with a plane which is defined by a point $A = [0,0,0]$ and normal $\mathbf{n} = [0,0,1]$. By intersecting the transformed line with that plane, we obtain the position on the screen at which we are pointing. See Figure 4b.



(a) Touch interaction (b) Direct pointing interaction (c) Dynamic mapping

Figure 4: The results of the calibration can be used in several ways, for example - touch based interaction, direct pointing interaction, or dynamic hand/finger projection mapping.

To project the image on user's hand, we need to find the point on the screen plane that lies on the line connecting the projector position and user's hand. This line can be defined by a point on the line and direction vector. As a point on the line, we can use the projector position and the direction vector is given by subtracting the projector and hand positions. This line is then intersected with the same plane as above. See Figure 4c.

4 EVALUATION

In this section, the accuracy of the proposed calibration method is compared to the accuracy of the capacitive touch display in the touch based interaction task.

The task involved touching the displayed points as close to their center as possible. The points that had to be touched were arranged in a regular five by five testing pattern. The testing pattern is shown in Figure 6. Each point was touched five times for one calibration attempt. Five calibration attempts for each of the three calibration cases, were performed. These calibration cases differed in the number of points (3, 9 and 16) in their calibration patterns. This process was repeated twice, once on a small screen and once on a screen with larger dimensions. The small screen was the screen of the capacitive touch display and the large screen was projected screen. The diagonal of the touch display was 256 mm and the screen resolution was 1368x768 px. The diagonal of the projected screen was 443 mm with resolution of 800x600 px. On the capacitive touchscreen, we performed the same task, but since there was no way to recalibrate the touchscreen we touched each point five times in five attempts without recalibration. No measurements for the capacitive touchscreen with large size were recorded since we were unable to find the screen with larger diagonal than 256 mm.

In our experiments we used the data projector Acer K11, laptop Asus Transformer Book T100 with touchscreen and the Leap Motion sensor. The environment for the experiments on the small screen was composed of laptop Asus Transformer and the Leap Motion Controller. The laptop's touchscreen was laid down horizontally on a table, while the Leap Motion was mounted over the screen looking down to the table. The environment is shown in Figure5 left. A setup for the experiments on the large screen used the Acer K11 projector instead of the laptop. The data projector was mounted approximately 800 mm over the table, in distance of circa 500 mm from the Leap Motion. By positioning the projector perpendicularly to the projection screen, we compensated for the keystoning. Other possible projector distortions were not compensated. Both devices were looking straight down to the table as shown in Figure 5 right and Figure 6.

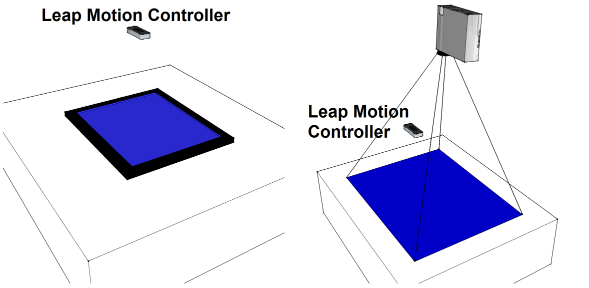


Figure 5: The experimental environment with the Leap Motion, the touchscreen (left) and the data projector (right).

The Leap motion was positioned 280 mm above the screen surface for both screen sizes. At that height, the tracking of the hand and fingers was most reliable in every point of the screens. During the experiments we noticed an issue with a reflection of the emitted IR light in the Leap Motion camera images. The reflected light made a bright spot on the table surface as shown

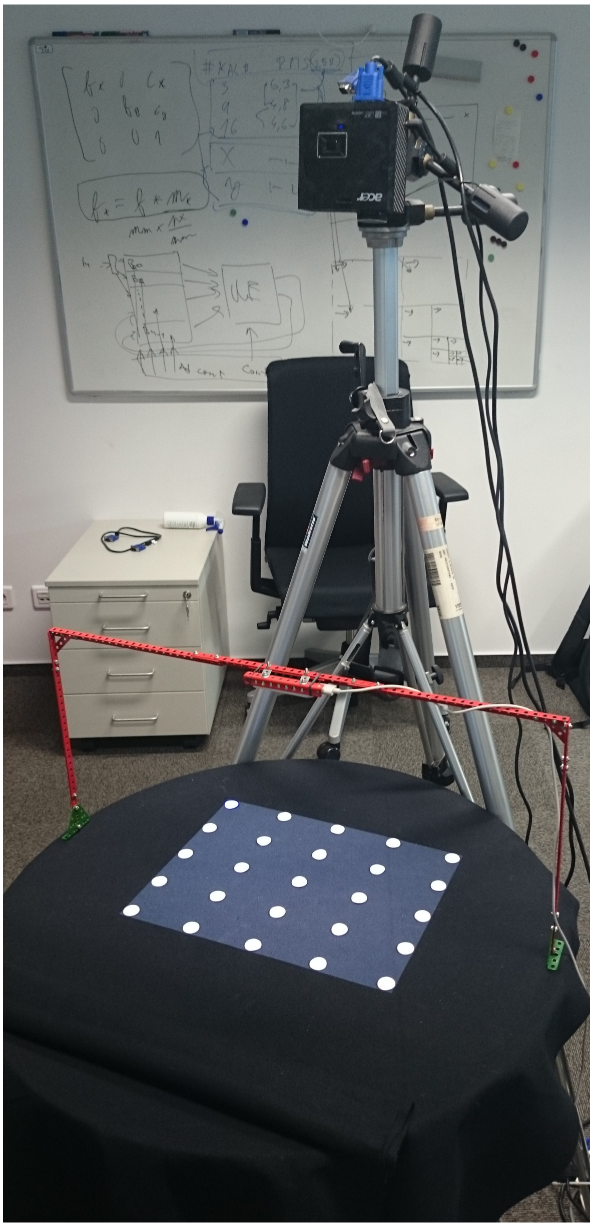


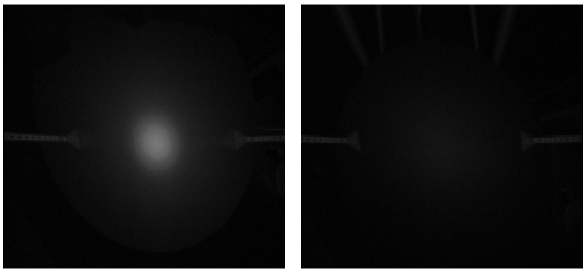
Figure 6: Our experimental tabletop setup with the data projector and the Leap Motion. Five by five pattern that was used for evaluation of the accuracy of our technique is being projected.

in Figure 7 left. Inside this spot the tracking was not reliable and often failed. Thus, it was necessary to cover the table desk with a black cotton fabric which absorbs IR light, but also reflects the visible light that forms the projected image. See Figure 7 right.

We used the Euclidean distance between the center of the displayed point and the point that was touched on the screen as a measure of accuracy. This distance was measured in px and then converted to mm using scale computed as

$$scale = \frac{\sqrt{H^2+V^2}}{D} \tag{13}$$

where H and V are horizontal and vertical resolutions of the screen in pixels and D is diagonal size in mm. This conversion makes the results for the large and small screens comparable. We evaluated the touch accuracy for seven configurations, one small touchscreen (TS), three calibration cases on the small screen (3S, 9S, 16S) and three calibration cases on the larger screen (3L, 9L, 16L). For each of these configurations, we performed 625 measurements. The collected data are visualized in Figure 8. From these measurements we calculated the mean and the standard deviation of distances between the touched point and the displayed point. We also calculated the hit percentage of a target with a radius of 7.75 mm which is recommended by Apple as a suitable size for interactive elements [1]. All results are displayed in Table 1.



(a) Table desk (b) Black cotton

Figure 7: The Leap Motion's view of the clear wooden table desk and the same desk covered by black cotton.

Config.		Mean (mm)	Stdev (mm)	Hit (%)
T	S	0.7897	0.4689	100
3	S	3.5420	2.2226	94.56
	L	3.8179	2.2223	96
9	S	1.7528	0.9585	100
	L	2.5187	1.2272	100
16	S	**1.2338**	**0.7468**	100
	L	2.1846	1.1162	100

Table 1: The mean and standard deviation are obtained from the distances between the displayed and measured point for each configuration together with hit percentages of the target with 7.75 mm radius. The values are measured for touchscreen (T) and proposed calibration method with 3, 9 and 16 calibration points on small (S) and large (L) screen sizes.

It is not surprising that the accuracy and precision of the calibration method rises with the number of points

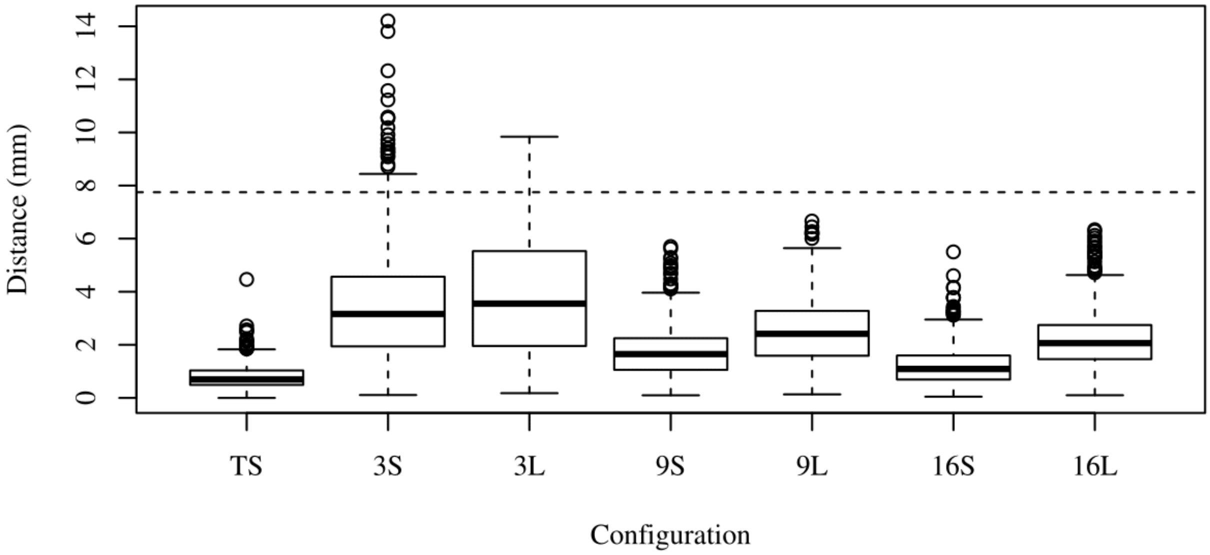


Figure 8: Boxplots of measured Euclidean distances in mm between the displayed and touched points for each configuration. The bottom and top of the box are the first (Q_1) and the third quartiles (Q_3). The band inside the box is the second quartile (Q_2) - the median. The whiskers extend from the lowest distance within $Q_1 - 1.5 * IQR$ to the highest distance within $Q_3 + 1.5 * IQR$. The circles are distances outside the whiskers. The dashed horizontal line represents the recommended target size.

in the used calibration pattern. Also, the accuracy and precision is better on the smaller screen. This can be caused by the projector distortions or by lower precision of hand tracking on the boundaries of the sensory space [5]. The best mean and standard deviation values for our method were obtained with 16 calibration points on the small screen. However, the results are approximately 60% worse than the touchscreen results, the mean value is slightly over one millimetre and the standard deviation is even less than one millimetre. The chance to hit a target of a recommended size are for all configurations 100%, except for calibrations that used 3 points in the calibration pattern. For these cases the hit chance is about 95%.

5 CONCLUSION

We have proposed a novel technique for calibration of a data projector and a Leap Motion Controller. Using the proposed approach, users can perform the calibration only by touching several points on and above the screen plane. No printed patterns, reflective markers, or additional tools are needed.

We have evaluated the proposed technique in a scenario that focuses on the touch based interaction. In this scenario we measured the distance between the center of the displayed point and the point that was touched on the screen. We compared the accuracy and precision of the proposed method with a capacitive touchscreen. The average accuracy of our method is about 1.2 mm on the small screens and 2.2 mm on the larger screens with 16 points calibration pattern. Although our calibration method does not outperform the accuracy and the precision of the touchscreen, it is comparable to it and can easily be used for effective touch based interaction as long as the target sizes follow the previously established recommendations for the touch screen interfaces. The results of the calibration can be used not only for touch based interaction but also for direct pointing and dynamic projection mapping.

Future research will focus on evaluation of the accuracy from the user's point of view. A user study will be used to evaluate whether the accuracy of our method affects user interaction experience. Another possibility is to focus on user interaction techniques suitable for the tabletop displays or to improve the calibration process by compensating for projector distortions.

ACKNOWLEDGEMENT

This work has been supported by the TACR centre of competence project V3C (no. TE01020415).

6 REFERENCES

[1] Apple Inc. iOS Human Interface Guidelines. Retrieved March 10, 2015, from `https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/LayoutandAppearance.html`.

[2] ASUSTeK Computer Inc. Multimedia - Xtion - ASUS. Retrieved March 10, 2015, from `http://www.asus.com/Multimedia/Xtion/`.

[3] A. Ben-Hamadou, C. Soussen, C. Daul, W. Blondel, and D. Wolf. Flexible calibration of structured-light systems projecting point patterns. *Comput. Vis. Image Underst.*, 117(10): 1468–1481, Oct. 2013. ISSN 1077-3142. URL `http://dx.doi.org/10.1016/j.cviu.2013.06.002`.

[4] G. Falcao, N. Hurtos, J. Massich, and D. Fofi. Projector-Camera Calibration Toolbox Tech. Rep. , 2009, available at `http://code.google.com/p/procamcalib`.

[5] J. Guna, G. Jakus, M. Pogacnik, S. Tomazic, and J. Sodnik. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors*, 14 (2):3702–3720, 2014. ISSN 1424-8220. URL `http://www.mdpi.com/1424-8220/14/2/3702`.

[6] C. Harrison, H. Benko, and A. D. Wilson. Omnitouch: Wearable multitouch interaction everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 441–450. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-0716-1. URL `http://doi.acm.org/10.1145/2047196.2047255`.

[7] Ideum. Touch Tables and Multitouch Coffee Tables. Retrieved March 15, 2015, from `http://ideum.com/touch-tables/`.

[8] iPiSoft. Depth Sensors Comparison. Retrieved March 10, 2015, from `http://wiki.ipisoft.com/Depth_Sensors_Comparison`.

[9] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. ISSN 1424-8220. URL `http://www.mdpi.com/1424-8220/12/2/1437`.

[10] J. J. LaViola, Jr. An introduction to 3d gestural interfaces. In *ACM SIGGRAPH 2014 Courses*, SIGGRAPH '14, pages 25:1–25:42. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2962-0. URL `http://doi.acm.org/10.1145/2614028.2615424`.

[11] Leap Motion Inc. Leap Motion. Retrieved March 10, 2015, from `https://www.leapmotion.com/`.

[12] K. Litomisky. Consumer rgb-d cameras and their applications. Tech. rep. University of California, Tech. Rep., 2012.

[13] C. B. Markwardt. Non-linear Least-squares Fitting in IDL with MPFIT. In D. A. Bohlender, D. Durand, and P. Dowler, editors, *Astronomical Data Analysis Software and Systems XVIII*, volume 411 of *Astronomical Society of the Pacific Conference Series*, page 251, Sept. 2009.

[14] Microsoft. Kinect for Windows. Retrieved March 10, 2015, from `https://www.microsoft.com/en-us/kinectforwindows/`.

[15] Microsoft. Samsung SUR40. Retrieved March 10, 2015, from `http://www.microsoft.com/en-us/pixelsense/whatsnew.aspx`.

[16] J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. Watson, editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer Berlin Heidelberg, 1978. ISBN 978-3-540-08538-6. URL `http://dx.doi.org/10.1007/BFb0067700`.

[17] D. Moreno and G. Taubin. Simple, accurate, and robust projector-camera calibration. In *Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, 3DIMPVT '12, pages 464–471. IEEE Computer Society, Washington, DC, USA, 2012. ISBN 978-0-7695-4873-9. URL `http://dx.doi.org/10.1109/3DIMPVT.2012.77`.

[18] S. Murugappan, Vinayak, N. Elmqvist, and K. Ramani. Extended multitouch: Recovering touch posture and differentiating users using a depth camera. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 487–496. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1580-7. URL `http://doi.acm.org/10.1145/2380116.2380177`.

[19] A. D. Wilson. Playanywhere: A compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 83–92. ACM, New York, NY, USA, 2005. ISBN 1-59593-271-2. URL `http://doi.acm.org/10.1145/1095034.1095047`.

[20] A. D. Wilson. Touchlight: An imaging touch screen and display for gesture-based interaction. In *ACM SIGGRAPH 2005 Emerging Technologies*, SIGGRAPH '05. ACM, New York, NY, USA, 2005. URL `http://doi.acm.org/10.1145/1187297.1187323`.