Exploring Machine Learning Object Classification for Interactive Proximity Surfaces

Andreas Braun[1,2(✉)], Michael Alekseew[1], and Arjan Kuijper[1,2]

[1] Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
{andreas.braun,michael.alekseew,arjan.kuijper}@igd.fraunhofer.de
[2] Technische Universität Darmstadt, Darmstadt, Germany

Abstract. Capacitive proximity sensors are a variety of the sensing technology that drives most finger-controlled touch screens today. However, they work over a larger distance. As they are not disturbed by non-conductive materials, they can be used to track hands above arbitrary surfaces, creating flexible interactive surfaces. Since the resolution is lower compared to many other sensing technologies, it is necessary to use sophisticated data processing methods for object recognition and tracking. In this work we explore machine learning methods for the detection and tracking of hands above an interactive surface created with capacitive proximity sensors. We discuss suitable methods and present our implementation based on Random Decision Forests. The system has been evaluated on a prototype interactive surface - the CapTap. Using a Kinect-based hand tracking system, we collect training data and compare the results of the learning algorithm to actual data.

Keywords: Capacitive proximity sensing · Interactive surfaces · Machine learning

1 Introduction

Capacitive sensing drives the most common interaction device of the recent years - the finger-controlled touch screen. A generated electric field is disturbed by the presence of grounded objects, such as fingers. This disturbance can be measured and is used to detect the location of an object on the surface [1]. Capacitive proximity sensors are a variety of this technology that is able to detect the presence of a human body over a distance. They can be used to create interaction devices that are hidden below non-conductive surfaces [2, 3]. However, for these sensors, object recognition is a major challenge, due to a comparatively low resolution and ambiguity of the detected objects [4].

In the last years researchers have investigated a number of different processing methods, including algorithms that try to distinguish multiple hands. Machine learning methods have become more popular in object recognition for 3D interaction devices. The Microsoft Kinect is a popular example that extensively uses machine learning for posture classification [5]. Le Goc et al. used the Random Decision Forest (RDF) method to improve the object recognition of a commercially available, small area capacitive proximity interaction device [6]. They used a stereo camera system to track the position

N. Streitz and P. Markopoulos (Eds.): DAPI 2016, LNCS 9749, pp. 157–167, 2016.
DOI: 10.1007/978-3-319-39862-4_15

of a fingertip with high precision as ground truth and trained a RDF, resulting in a finger tracking precision that exceeded the manufacturer's methods. In this paper we want to evaluate, how machine learning methods, such as RDF can be used to realize object recognition and tracking on large-area interaction systems. The CapTap is an interactive table that uses capacitive proximity sensors to provide a 3D interaction paradigm [7]. It is comprised of 24 sensors hidden under a wooden or plastic cover. It starts detecting objects at a distance of approximately 40 cm. The sensors use the OpenCapSense rapid prototyping toolkit for capacitive sensing [8].

Using a second system that supports precise hand tracking, we can acquire ground truth data. The idea is to perform a classification of volume elements that the center of the hand resides in. For this it is necessary to collect training samples from within each volume element. The hand tracker uses a background subtraction method from a depth image and looks for the innermost point in a recognized hand shape. In order to determine the 3D position the extrinsic camera parameters have to be determined in a calibration routine. This idea was implemented in a prototype that uses the Kinect v2 as a system to acquire the hand position. Additionally, we propose several modifications to improve the localization accuracy.

A study was performed with the proposed system. We collected several hundred training samples to create and test the classifier. This paper proposes the following scientific contributions:

- Propose a method to track multiple hands over a capacitive proximity interaction device using RDF classification
- Create a system to accurately measure the relative position of hands from a surface using a fast hand detection algorithm for the Microsoft Kinect v2
- Evaluation of the system using 1500 training samples and proposing several methods to improve localization based on RDF classification results.

2 Related Works

In the past there have been several interaction devices for mid-air interaction using capacitive proximity sensors. Zimmerman et al. propose various applications for this technology in HCI [3]. One of the prototypes is a smart table that tracks the hand position above the surface. For object detection they model the electric field of dipole pairs and use inverse methods to determine the position of one or more hands. Simpler methods that use a weighted average to interpolate between evenly spaced sensors and approximations for estimating the elevation above the surface have been proposed [2]. Grosse-Puppendahl et al. present a probabilistic method that estimates for each sensor areas, in which no object can be present [9]. This results in a "swiss-cheese-like" probability distribution, whereas the object has to be part of the remaining high-probability areas. The method that inspired our work was presented by Le Goc et al. [6]. They use RDF classification to improve the accuracy of finger tracking for a capacitive interaction device by Microchip [10]. This system uses a layout of a single sender and multiple receiver electrodes placed around the sender. They achieve a good accuracy in a constrained interaction area. The accuracy of the supplied object detection algorithm

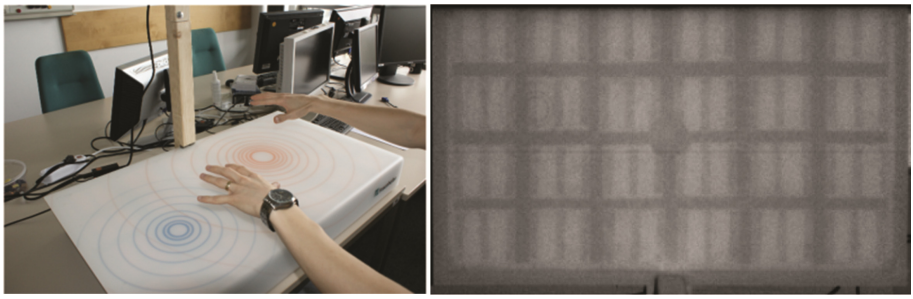

Fig. 1. CapTap capacitive proximity interaction table (left) and an infrared images that shows electrodes and microphone below the surface (right).

could be significantly improved from an average error of >20 mm to less than 5 mm, by using a RDF classifier that was trained with 32000 samples interpolation of the tracked object using Kalman filtering.

Our proposed method is designed for large area interactive tables that support interaction above the surface. One of those systems is the MirageTable by Benko et al. [11]. This curved interactive surface combines hand tracking in three dimensions using the Microsoft Kinect and Augmented Reality with a stereo projector attached above the table. The users have to wear shutter glasses for the stereo projection. An early system that uses capacitive sensors for touch recognition is DiamondTouch by Dietz et al. [12]. It uses a multilayer setup of electrodes and is thus able to track several touch points from different users. The system we used in this work is CapTap, an interactive table that combines capacitive proximity sensors for object detection above the surface and an acoustic sensor for detecting and distinguishing different touches (Fig. 1) [7]. It creates capacitive images and applies various image processing methods to detect the location and orientation of hands and arms. The acoustic sensor uses analysis in the frequency domain and a trained classifier to recognize different touch events, such as taps and knocks [13].

The system requires training data as ground truth. Therefore, we need methods that track the position of a hand in 3D. The Microsoft Kinect is an affordable sensor that provides reasonably precise collection of depth data in a sufficiently large area. It is primarily intended for pose tracking [5]. Some methods have been proposed for the first version that track the position of multiple hands [14]. A novel method for the Kinect v2 enables the tracking of fingers with high precision [15]. However, for our proposed system the focus was on fast tracking of the palm position above a surface. Therefore, a custom method was implemented that will be outlined in the following section.

3 Hand Tracking Using Random Decision Forests

RDF is a classification method in machine learning [16]. Based on classical decision trees it tries to overcome their tendency to overfitting by creating a multitude of trees during training time and ending up with the class that is the statistical mode of the classes or mean prediction of the individual trees.

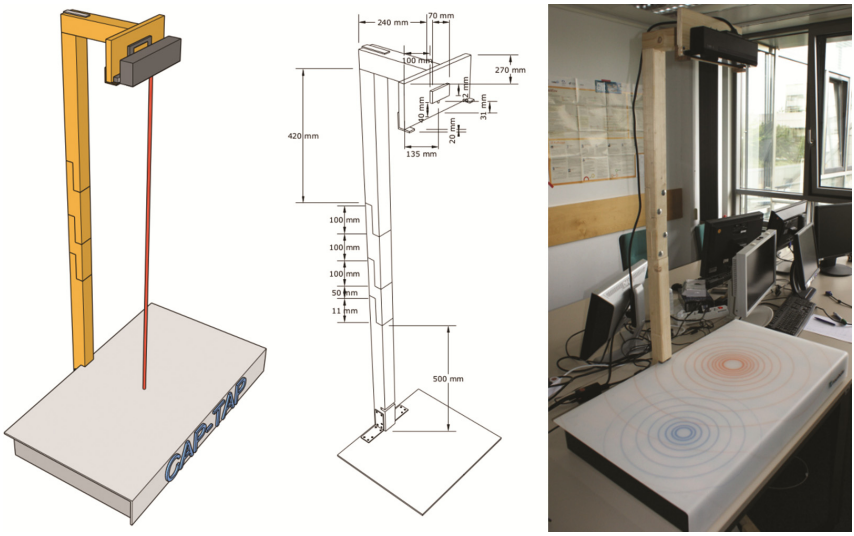

Fig. 2. Rendering and dimensions of Kinect v2 stand and its position above CapTap

The method presented by Le Goc et al. that we presented previously is less suitable for large interaction devices, as the supervised area is considerably larger and as we want to track the position of multiple hands above a surface. This requires a system that is applied over the interactive surface that observes the full interaction space. Another requirement we had is flexibility of the approach. The users should not be required to wear any markers, even during the training phase. Thus we decided on the following setup, which is shown in Fig. 2:

1. A stand was built that holds a video system at an elevation above the surface of the interaction device.
2. The Kinect v2 is attached to the stand and observes the scene with both depth and RGB camera.
3. A computer vision algorithm detects the center of the palms of the hands in the interaction area in three dimensions.
4. A software suite collects the data of the capacitive sensors and the detected hand positions as training and test data.

3.1 Hand Tracking Algorithm for Kinect v2

Before we can start to accurately track the position of hands using the Kinect depth image, it is necessary to calibrate the position of the Kinect with respect to the interactive surface. Even if the stand is mechanically solid, small movements of the interactive surface or the Kinect can lead to differences in the resulting position.

We are using an edge detector on the depth image to find the boundaries of CapTap in the scene. As the depth images are fairly noisy, an average of 150 frames is used. The four corner points of the boundary allow us to create a transformation matrix that corrects

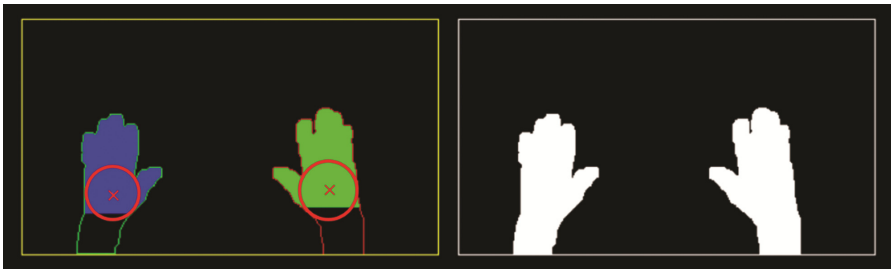

Fig. 3. Left - detected hand palm centers and their distance to the borders of the hand. Right - background subtraction of the hands in the air.

for position in x, y, and z. This transformation matrix is applied to points of the Kinect coordinate systems that are translated to points in the CapTap coordinate system.

The next step is to use computer vision methods to find the center points of the palms of multiple hands, which are later used as input for the classification. The method is inspired by previous works that detect hand position from depth images [17]. There are four basic steps:

1. Segmentation of hands using background subtraction, as shown in Fig. 3 on the right. Creates a binary black and white image.
2. Use a combination of the morphological operations erode and dilate to reduce noise in the remaining image.
3. Apply a region growing algorithm to find pixels that belong to the interior or exterior of a hand.
4. Find the interior point of a hand that is furthest from any edge, as shown in Fig. 3 on the left. Transform the coordinates to the CapTap coordinate system.

This approach is fast enough for execution in real-time (>30 frames per second). We applied some optimizations that reduce the number of candidate pixels for the palm center, such as discarding candidates very close to the edges or candidates that are part of a finger.

3.2 Acquiring Training Data

During the training process a certain amount of samples has to be collected for each potential hand location. The location in this context is a cubic volume element (voxel) with a specific edge length. This length is the achievable resolution of the system. For example, if we want to have a resolution of 3 mm and need 5 samples from each location, the naïve approach for the CapTap with its interaction area of 800 mm × 400 mm × 200 mm leads to a required number of samples of.

$$num_{samples} = 5 * \frac{800\,\text{mm} * 400\,\text{mm} * 200\,\text{mm}}{(3\,\text{mm})^3} \approx 11851852$$

As we are limited to collecting 30 samples per second the whole process would take approximately 110 h if the hands move perfectly. The complexity increases if we want to recognize multiple hands.

This is not feasible in the scope of this work, thus several simplifications have been applied. The resolution of the system was restricted to voxels with 20 mm edge length, which cuts down the training time significantly. Additionally, the classification of multiple hands assumes a minimal distance between those, which reduces the number of training samples required.

The training data is an array of 24 sensor values that are acquired by the CapTap and the hand position as acquired from the Microsoft Kinect. A software suite was created that supports the collection of trainings samples, as well as the configuration of all relevant parameters in the data collection and training steps. A screenshot of this system is shown in Fig. 4. On the left we can see a heatmap of the interaction area in two dimensions that indicates how many voxels have all training samples collected. Since this is a three dimensional problem, an additional progress bar on the top right shows how many samples have been collected in the current voxel. Additionally, the software tells how far the training has progressed by a counter in the bottom center. The collected training data can be directly used to create a RDF, whereas the number of branches and tress can be configured using several input fields. The results are stored in files that can be loaded during the classification view that is introduced in the next section.

Fig. 4. Software suite that collects training data. Left - view of hand tracker and classification results, right - view of collected training samples within the interaction area.

4 Random Decision Forests Location Classification

In this work, the result of the classification with the RDF is a probability for trained region to be the most fitting to current sensor values. Figure 5 shows an example of the four most likely voxels in a classification process. There are multiple ways to use these results in the final localization routine. In this work we investigated several methods:

1. Assigning the center of the dominant voxel as the location of the palm of the hand, subsequently called *naïve-center*.
2. Linear interpolation between the centers of the vxoels with the highest probability, subsequently called *linear-interpolation*.
3. Weighted interpolation between the centers of the most probable voxels according to their probability, subsequently called *weighted-interpolation*.

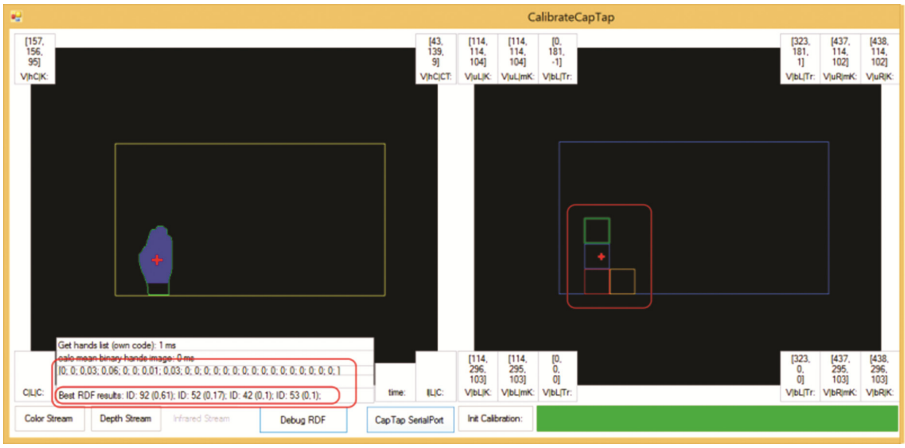

Fig. 5. Left - position as determined by Microsoft Kinect. Right - result of a RDF classification showing four most likely voxels in two dimensions.

Naïve-center is the basic approach that does not require any additional knowledge about the geometric distribution of the voxels in the interaction area. Disadvantages include a highly-quantized localization, whose resolution is limited to the voxel edge length. If there are multiple voxels that have a similar probability, it is likely that the true position is somewhere between the detected voxels.

Linear-interpolation solves the problem of similar probabilities and can successfully localize hands that are between voxels. However, the simple linear interpolation can lead to overrepresentation of unlikely voxels. If the most likely voxel has a probability of 0.9 and a voxel that is far away has 0.05, the linear-interpolation would lead to a position somewhere between the voxels that is likely wrong. A first improvement is discarding voxels that have a low probability.

Weighted-interpolation tries to overcome this disadvantage by attaching a weight to the voxels according to their probability. This leads to a location that has a tendency towards the center of the most probable voxel. A remaining disadvantage of this approach is the lack of correlation between classification probability and hand position. If the hand is at the border between two voxels there is no guarantee that the classification assigns both the same probability. However, in practical applications we found that this often leads to the best results.

5 Evaluation and Discussion

On the following lines we are discussing some findings occurring during the system design and implementation stages, as well as the results of a study performed on a fully trained system.

In some instances the classification results can lead to inconclusive results, an example of which is shown in Fig. 6. The dominant voxels are near the elbow with only one voxel having low probability near the hand. This effect occurs if the arm is at fairly high elevation above the CapTap. The measurements are strongly influenced by noise, which in turn affects the classification. Apart from improving any applied noise reduction, there are some other options to improve the classification, such as a minimal probability that is required for classification, or a non-uniform handling of the elevation information acquired by capacitive sensors [4].

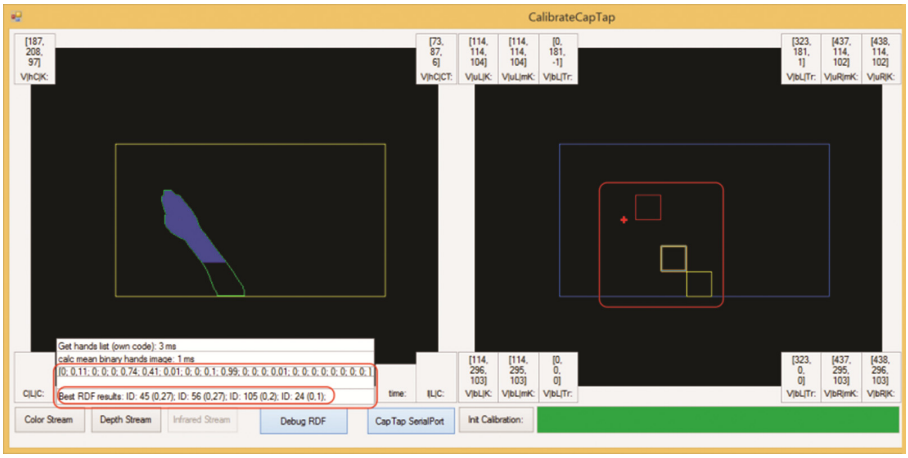

Fig. 6. Inconclusive classification result with low probabilities for most voxels

A second example is shown in Fig. 7. The classification is slightly more supportive of the bottom voxel, whereas the hand is in fact closer to the blue voxel in the middle. This is an instance where weighted-interpolation will lead to acceptable results of the overall localization process. The upper red and yellow voxels have a smaller probability assigned and would lead to a minimum trend of the detected hand palm position towards this direction.

We evaluated two of the algorithm varieties on two RDFs that were trained using a different set of parameters. Table 1 presents the results of this evaluation. The tracking methods were naïve-center and weighted-interpolation. Using a set of 827 collected samples, two varieties of RDFs were trained. The first uses 100 trees N and 75 % of the samples for training (denoted r in the table). The second uses 50 trees and 66 % of the samples for training. The ratio of correct voxels denotes the ratio of overall classified locations that coincide with the true position voxels within the collected training data.

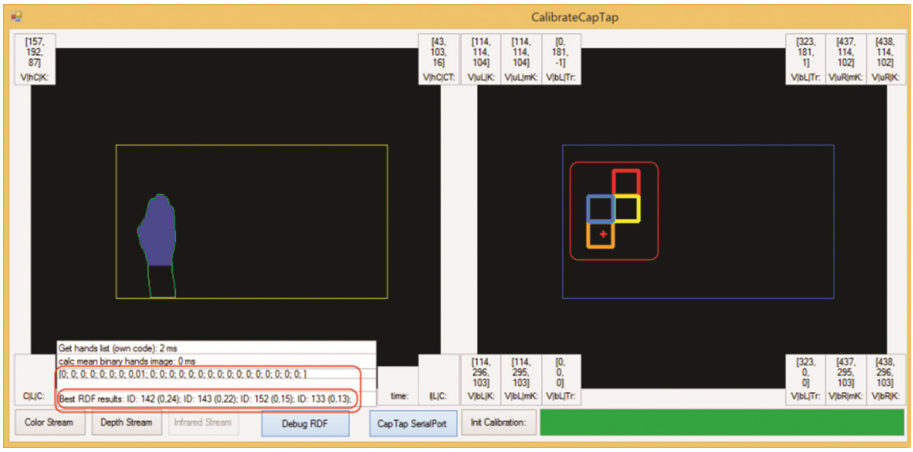

Fig. 7. Classification result suitable for weighted-interpolation with nearby voxels being assigned suitable probabilities.

Table 1. Evaluation results of several algorithm varieties and RDF settings using 827 training and test samples.

Algorithm	RDF parameters	Ratio correct voxels	Average distance error
Naïve-center	N = 100, r = 0.75	27.58 %	4.4 cm
Naïve-center	N = 50, r = 0.66	37.48 %	3.9 cm
Weighted-interpolation	N = 100, r = 0.75	31.13 %	4.2 cm
Weighted-interpolation	N = 50, r = 0.66	47.50 %	2.3 cm

The ratio of correctly classified voxels was comparatively low for both varieties, ranging from 28 % to 48 %. The average distance error was better, whereas the classified hand position was between 2.3 cm and 4.4 cm from the ground truth. In this case the distance is measured from the center of the voxel as collected by the Kinect.

The RDF with a lower number of trees generally performed better than the RDF with a higher number of trees. During the evaluation we could observe that there are is the expected strong correlation of RDF classification probability and voxels close to the training data, even if they often do not fit exactly. Even with a very coarse voxel resolution this creates the opportunity to get a good average distance error of the hand palm from the true position. This finding can be used to optimize the training process in the future, by using suitable interpolation methods.

6 Conclusion and Future Work

In this work we explored the use of RDF classification for the tracking of hands in three dimensions above a large-area interaction devices using capacitive proximity sensors. There are several challenges in using this methods. The number of training samples

increases exponentially with the intended resolution, as the interaction area is a three-dimensional space. In addition sensor noise and calibration of the initial sensor values become a challenge in unconstrained conditions.

We have proposed two interpolation methods for improving the localization of objects, based on classification results. An efficient algorithm was developed to calculate the center of the hand palm based on simple computer vision operations on the depth image of a Kinect v2. We evaluated the system and achieve a good localization error, even if the classifier did not find the correct voxel in the majority of the cases.

We would like to use the symmetry of the sensor setup to create more efficient training methods that are performed over a smaller area, but whose results can be used for other areas of the interaction device. We will collect more training data to get measurements at smaller resolutions, using this method. Based on that we can evaluate if resolutions comparable to vision-based systems are achievable.

Acknowledgments. We would like to thank all volunteers that participated in our studies and provided valuable feedback for future iterations. This work was supported by the European Commission under the 7th Framework Programme (Grant Agreement No. 611421).

References

1. Barrett, G., Omote, R.: Projected capacitive touch technology. Inf. Disp. **28**, 16–21 (2010)
2. Braun, A., Hamisu, P.: Designing a multi-purpose capacitive proximity sensing input device. Proc. PETRA (2011). Article No. 15
3. Zimmerman, T.G., Smith, J.R., Paradiso, J.A., Allport, D., Gershenfeld, N.: Applying electric field sensing to human-computer interfaces. In: Proceedings of the CHI, pp. 280–287 (1995)
4. Braun, A., Wichert, R., Kuijper, A., Fellner, D.W.: Capacitive proximity sensing in smart environments. J. Ambient Intell. Smart Environ. **7**, 1–28 (2015)
5. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. Commun. ACM **56**, 116–124 (2013)
6. Le Goc, M., Taylor, S., Izadi, S., Keskin, C., et al.: A low-cost transparent electric field sensor for 3d interaction on mobile devices. In: Proceedings of the CHI, pp. 3167–3170 (2014)
7. Braun, A., Zander-Walz, S., Krepp, S., Rus, S., Wichert, R., Kuijper, A.: CapTap - combining capacitive gesture recognition and knock detection. Working Paper (2016)
8. Grosse-Puppendahl, T., Berghoefer, Y., Braun, A., Wimmer, R., Kuijper, A.: OpenCapSense: a rapid prototyping toolkit for pervasive interaction using capacitive sensing. In: 2013 IEEE International Conference on Pervasive Computing and Communications, PerCom 2013, pp. 152–159 (2013)
9. Grosse-Puppendahl, T., Braun, A., Kamieth, F., Kuijper, A.: Swiss-cheese extended: an object recognition method for ubiquitous interfaces based on capacitive proximity sensing. In: Proceedings of the CHI, pp. 1401–1410 (2013)
10. Microchip Technology Inc.: GestIC ® Design Guide: Electrodes and System Design MGC3130 (2013)
11. Benko, H., Jota, R., Wilson, A.: Miragetable: freehand interaction on a projected augmented reality tabletop. In: Proceedings of the CHI, pp. 199–208 (2012)
12. Dietz, P., Leigh, D.: DiamondTouch: a multi-user touch technology. In: Proceedings of the UIST, pp. 219–226 (2001)

13. Harrison, C., Schwarz, J., Hudson, S.E.: TapSense: enhancing finger interaction on touch surfaces. In: Proceedings of the UIST, pp. 627–636 (2011)
14. Ren, Z., Meng, J., Yuan, J.: Depth camera based hand gesture recognition and its applications in Human-Computer-Interaction. In: 2011 8th International Conference on Information, Communications and Signal Processing, pp. 1–5 (2011)
15. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., Izadi, S.: Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3633–3642. ACM, New York (2015)
16. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
17. Cerezo, F.T.: 3D hand and finger recognition using Kinect. Project report, University of Granada (2011)