Multimed Tools Appl
DOI 10.1007/s11042-016-3695-5

Touch detection for planar interactive displays based on lateral depth views

Antonios Ntelidakis[1] · Xenophon Zabulis[1] · Dimitris Grammenos[1] · Panagiotis Koutlemanis[1]

Received: 14 December 2015 / Revised: 26 April 2016 / Accepted: 14 June 2016

Abstract This work regards fingertip contact detection and localization upon planar surfaces, for the purpose of providing interactivity in augmented, interactive displays that are implemented upon these surfaces. The proposed approach differs from the widely employed approach where user hands are observed from above, in that user hands are imaged laterally. An algorithmic approach for the treatment of the corresponding visual input is proposed. The proposed approach is extensively evaluated and compared to the top view approach. Advantages of the proposed approach include increased sensitivity, localization accuracy, scalability, as well as, practicality and cost efficiency of installation.

Keywords Human computer interaction · Spatial augmented reality · Interactive surface · Touch detection · Depth camera

Electronic supplementary material The online version of this article (doi:10.1007/s11042-016-3695-5) contains supplementary material, which is available to authorized users.

✉ Antonios Ntelidakis
ntelidak@ics.forth.gr

Xenophon Zabulis
zabulis@ics.forth.gr

Dimitris Grammenos
gramenos@ics.forth.gr

Panagiotis Koutlemanis
koutle@ics.forth.gr

[1] Foundation for Research and Technology — Hellas (FORTH), Institute of Computer Science, N. Plastira 100, Vassilika Vouton, Heraklion, Crete, 700 13, Greece

Published online: 28 June 2016

1 Introduction

A significant component of smart environments is the direct interaction with non-instrumented physical surfaces. Corresponding systems, augment such surfaces using a projector to create a "display" upon them [36]. Simultaneously, they utilize sensing to detect and localize the contact of fingertips upon the surface and generate touch events. In the majority of cases, the interactive surface is planar (i.e. a wall or a table), because planar surfaces facilitate projection and touch detection [6, 18, 39]. This work focuses on the detection of fingertip contact, or touch, upon planar, non-instrumented surfaces for use in an augmented, interactive display.

Requirements for natural interaction and avoidance of surface instrumentation call for the use of unobtrusive, visual sensors and corresponding detection approaches. In visual approaches, touch is conventionally detected based on 3D information about the interaction surface and the fingertips. Early such approaches employed stereo cameras [43] for this purpose. The recent proliferation and advantages of consumer depth cameras has expanded interaction capabilities and dominated state of the art. Full articulation, hand tracking systems (i.e. [28]) solve a more complex problem and exhibit high computational cost. They also prioritize finding the overall hand articulation over achieving fine accuracy in fingertip localization. Thus, approaches that are more focused to the specific problem of touch detection have been proposed in the literature.

Common ground in these approaches is the placement of the sensor above the interaction surface. In this top-view configuration, it is mainly the top face of a finger that is imaged. Contact hypotheses are based on the estimated distance of the imaged finger to the surface. Touch is, implicitly, detected when this distance falls below some threshold. This threshold is determined by an assumption of finger thickness, as well as, the precision and accuracy of the sensor.

In this work, the lateral placement of the sensor is proposed, along with a method for touch detection and localization. In this configuration, the interaction surface is imaged as a line, or a horizon. Touch is detected in the depth image as the contact of the finger contour with this line. The proposed approach exhibits greater sensitivity than the top-view touch detection. This is due to that it directly images the contact event. Contact detection is based on xy pixel locations of finger detection, rather than inferred from depth values. This increased sensitivity is important in terms of usability, because it means that system response better matches haptic feedback from actual fingertip contact with the surface.

The proposed approach is scalable to the utilization of multiple sensors, in order to cover greater areas. Compared to the conventional top-view approach, it exhibits wider area coverage per sensor and reduced computational cost. In Fig. 1, both approaches are illustrated. Both approaches are based on the depth image of the sensor. However, as more intuitive, the RGB images are instead shown in several figures of this paper.

The remainder of this paper is organized as follows. In Section 2 related work is reviewed. In Section 3 the proposed method is formulated. Details are provided for its calibration, the utilization of multiple sensors, and a discussion of its limitations. The method is evaluated in Section 4, from multiple perspectives. These include localization error, touch sensitivity, sensor placement as well as, usability and applicability in interactive displays. Where relevant the method is, furthermore, evaluated comparatively to the top view approach. Conclusions and directions for future work are provided in Section 5.

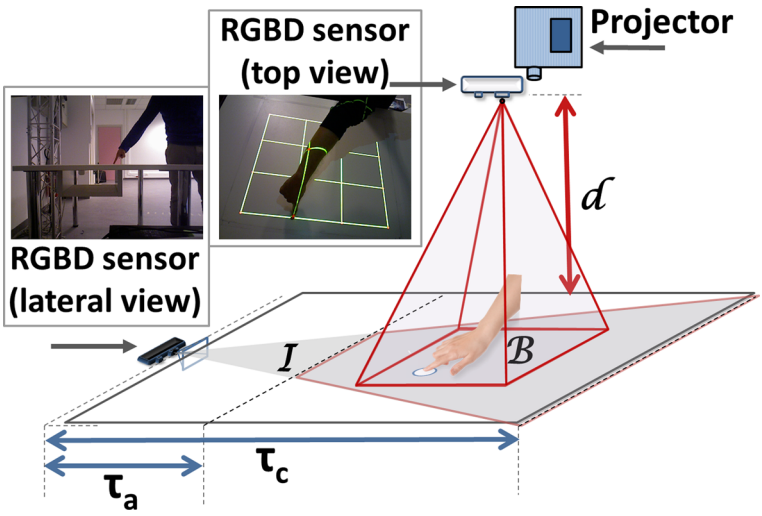

Fig. 1 Illustration of top and lateral RGB-D sensor placement and acquired images. For lateral placement, $[\tau_a, \tau_c]$ is the utilized depth range and $\mathcal{I}$ the area that fingertips can be detected. For top placement, d is the distance of the sensor to the surface and $\mathcal{B}$ the area that fingertips can be detected. (see Section 3 for definition of notation)

2 Related work

Spatial Augmented Reality (SAR) [4], is relevant to smart environments. It envisages the potential utilization of any physical surface as an interactive display. A prominent category of surfaces are planar surfaces, due to ease of projection, availability in ordinary environments, and practicality. This review regards approaches to the detection of touch events upon a surface in order to cast it interactive. Focus is given upon unobtrusive methods.

Non-visual approaches to contact detection on surfaces require special hardware and arduous deployment. At the same time, instrumentation can be obtrusive to the users. Such interactive displays utilize resistance, capacitive, and surface wave sensors to detect touch (see [3, 33] for reviews). SmartSkin [31] and DiamondTouch [6] utilize mesh-shaped antennas that cover the interactive surface. These antennas along with a deployment of a front projection unit form a tangible interactive surface. Recently, using a pertinent approach, touch detection and localization was provided upon worn garments [32].

Early efforts to visual touch detection upon planar surfaces, utilized semi-transparent, back-projected systems. In such systems IR illumination is cast to facilitate contact detection and localization, through a visual sensor placed behind the projection screen along with the projector [22, 24, 25]. Similarly, systems based on the Frustrated Total Internal Reflection principle [9, 17], utilizes LEDs that emit light within the surface. The purpose is to illuminate fingertips in contact and facilitate their detection. These systems require surface instrumentation and cannot be augmented in arbitrary surfaces, such as a wall or a desk. Also, due to the projection, they require a large installation volume and, typically, a fragile semi-transparent screen.

This work focuses on touch detection approaches, that can be applied to virtually any surface without surface instrumentation. To detect fingertips upon the surface, early vision

approaches utilized distortions upon the projected image pattern. In particular, these distortions were due to the presence of fingers, [1, 10, 35], or hand shadows [12]. More recent approaches use RGB-D sensors to estimate fingertip 3D locations. These locations are compared against an, a priori obtained, 3D model of the surface. Besides directly providing 3D information, such sensors are invariant to illumination shadows and other artifacts. In the seminal work in [40], the camera is above a planar interaction surface. During setup time, depth data are used to approximate the plane that models the physical surface. At run time, 3D points close to this plane are considered. Only the top face of the finger is imaged, rather than the bottom which comes in contact with the surface. Thus, touch events are implicitly detected. Detection utilizes an upper threshold to isolate candidate pixels close to the surface. A lower one, τ_X, is used to select pixels imaging fingertips, from pixels imaging the surface. As sensor depth accuracy is limited this threshold is usually in the order of $\tau_X = 1\,cm$. In this order, pixels imaging fingertips are robustly discriminated from sensor noise. Due to limitations of sensor precision and noise, fingertips approximately or even closer than τ_X to the surface, but still not in contact with it, trigger spurious touch events. This reduces interaction intuitiveness as the temporal disparity between actual fingertip contact and touch event generation is perceived by users. The proposed work increases sensitivity to touch by reducing this distance, where touch events can be spuriously triggered.

In dSensingNI [19, 20], the same principle as in [40] is employed. In that work the shape of the blob contour is analyzed in order to more accurately localize fingertips. In Microsoft's LightSpace concept [41], the approach in [40] is extended for multiple planar surfaces. In [21, 44] the approach in [40] is extended for a rotatable planar surface. Instead of a priori modeling the planar surface, it continuously estimates its orientation at run-time, excluding user hands with a robust plane fitting method.

In WorldKit [42] the same principle as in [40] is employed, but the method detects palms in contact with the interaction surface. Spatial consistency is better exploited this way, as only large blobs can trigger a touch event. Thereby, lower threshold τ_X that determines sensitivity can be set to an even lower value ($\approx 3\,mm$). However, since touch is determined for a palm instead of a fingertip, the lower threshold comes at the cost of reducing the spatial granularity by which touch is sensed.

The assumption regarding the planarity of the interaction surface has been relaxed in [11] and [14], which allow touch detection on arbitrary surfaces. The approaches employ the depth camera to model the interaction surface as background. The method in [11] uses the same principle as in [40] to detect touch, while [14] employs a stylus to create touch events.

Microsoft's Holodesk, MirageTable and Roomalive concepts [2, 13, 15], further extend the affinity of interaction. Collision detection is utilized to detect touch events. As in the aforementioned approaches, a top-view sensor placement is utilized and a 3D representation of the stationary scene is captured off-line. These works mainly focus on the interaction of user hands with virtual objects. A physics engine is employed to detect contacts (collisions) of body parts or hand-held objects with the virtual objects.

Lateral view approaches to touch detection have been proposed. These approaches either require multiple sensors and illumination components or utilize specialized high-cost laser sensors. In [38], multiple conventional (i.e. 4) cameras and stripe-shaped luminous patterns occluded by user fingertips were used to detect touch. In [30], a high-cost Time-of-Flight (ToF) laser scanner is employed to support multitouch interaction. A similar but more cost-efficient approach was utilized in [37]. There multiple laser planes were utilized to estimate the distance of a fingertip to the interactive surface. In [16] multiple views, including both

top and lateral views, are employed to reconstruct fingertip location in 3D. To the best of our knowledge, the proposed work is the first that utilizes a single RGB-D sensor laterally imaging the interactive surface.

An early presentation of the proposed approach is made in [27]. This work extends it, elaborates upon the process of system calibration and investigates system limitations. The evaluation of the proposed approach is also extended for a wide range of configurations and use cases. Calibration of the projector camera system is challenging in the proposed approach. This stems from that the camera and projector do not share a common field of view / projection. In the extended evaluation, system localization and accuracy is comparatively evaluated in a wider range of experimental conditions. In this way, the proposed approach is better characterized, its limitations understood, and its advantages to the top view approach more clearly discussed. In addition, more thorough usability evaluation through pilot applications is provided.

3 Implementation

In the proposed approach, the camera is imaging the scene laterally. Ideally, the camera would be placed so that the interaction plane is perpendicular to the image plane. Thus, it is projected solely as a 2D line $\mathcal{L}$, or a "horizon", in the middle row of the acquired image. In practice, this is technically difficult to achieve. The sensor may be placed somewhat higher above the interaction plane, than the ideal configuration (see Fig. 2). In this case the interaction plane is imaged quite obliquely. The horizon occurs above the middle image row, and $\mathcal{L}$ is placed below the middle image row to cover the volume of interest. In this configuration, the depth image region corresponding to the interaction surface contains typically null depth values. This is due to the sensor's limitation in capturing very oblique surfaces. Rare, transient, and noisy reconstructions of small segments of the interaction plane in the depth image are filtered (in Sections 3.2.2 and 3.4).

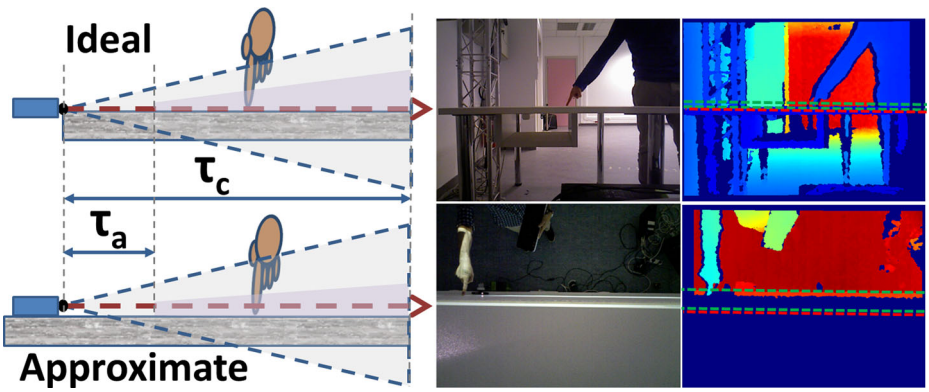

Fig. 2 *Left:* Side-views of ideal (*top*) and approximate (*bottom*) lateral sensor placement. *Dashed red arrows* plot sensor principal axes. The *light gray area* represents the volume within which 3D data is collected. *Dark gray rectangles* represent the interaction plane. *Right:* Images acquired for the ideal (*top*) and approximate (*bottom*) sensor placement. Superimposed, *dashed green rectangles* show $\mathcal{Z}$, whose lower edge occurs upon the horizon $\mathcal{L}$ (*red dashed line*)

By thresholding depth values, the search for detection of touch can be constrained within the $[\tau_a, \tau_c]$ range of depths. In that range the sensor provides fairly reliable depth measurements. Given the sensor's Field of View (FOV) this range defines a quadrilateral. If a rectangular display is pursued, its (approximately) parallel faces can be aligned with the edges of the display, in multiple configurations (see Section 3.1).

To collect pixel support for contact detection a zone, aligned to $\mathcal{L}$, is considered. This zone, $\mathcal{Z}$, is a region of interest, oriented parallel to $\mathcal{L}$, that has a rectangular shape. Zone $\mathcal{Z}$ has height τ_h pixels and its frustrum is shown in Fig. 2. This volume is not constant across the interaction surface. However, it was observed that a few image rows (i.e. $\tau_h = 5$) are sufficient for the collection of a reliable support for fingertip detection and localization.

The top-view, depth-based methods in Section 2, detects touch not only when an object is actually in contact with the surface, but also when in proximity to it. We define s, as the maximum distance from the surface that contact is detected. Sensitivity is then to be thought as inversely analogous to s. Intuitively, sensitivity is the minimum distance at which the system can reliably discriminate when the fingertip is in contact to the surface, or not. Thereby, the smaller s is, the more sensitive the contact detection method is.

3.1 Calibration

The purpose of the calibration process is to estimate, at setup time, geometrical entities and quantities, utilized by the system at runtime. A difficulty in the calibration of the proposed approach is that the camera does not properly image the projection area, due to its large obliqueness. Typically, null values are returned from such surfaces in depth cameras (i.e. see Fig. 2, bottom-right). To associate coordinates on the interaction area with projector coordinates, the proposed touch detection method in Section 3.2 is utilized. The following quantities and geometrical entities are estimated during calibration.

The region of interest $\mathcal{Z}$ in the depth image. This involves a user task, where $\mathcal{L}$ is determined by clicking upon two points upon the horizon of the interaction plane. Then, $\mathcal{Z}$ is a 2D rectangle above the horizon with its lower edge occurring in $\mathcal{L}$ and with height τ_h pixels that extends across the image (see Fig. 2). Intuitively, it is a "zone" of τ_h pixels above the horizon. In the case of "approximate" sensor placement, τ_h is larger to support touch detection at close ranges (see Fig. 2).

An estimate $\mathcal{P}$ of the interaction plane in the camera reference frame, so that fingertips in contact with the surface are localized in 2D display coordinates. This is achieved implicitly as the interaction plane is not imaged by the depth camera. The operator traces a fingertip across the interaction plane, while the system collects pixels from $\mathcal{Z}$ and interprets them as 3D points. A RANSAC-based robust plane fit estimates a plane from these points [7]. Coordinate transformation $\{R, \mathbf{t}\}$, that maps $\mathcal{P}$ to plane xy ($z = 0$), is computed through SVD decomposition.

A homography H that maps 2D coordinates from the xy plane to coordinates of the display buffer $\mathcal{F}$ of the projector. During the calibration procedure, the display projects minute luminous circles at designated display locations on $\mathcal{F}$ and the operator touches them. The system detects fingertips in $\mathcal{Z}$ and computes their corresponding 3D points $\mathbf{c}_j$. Points $\mathbf{c}_j$, are converted to 2D coordinates first by bringing them in $\mathcal{P}$'s reference frame, as $[x_j\ y_j\ z_j]^T = R\,\mathbf{c}_j + \mathbf{t}$. Truncation of the z dimension, projects these points on $z = 0$, converting them to 2D points $[x_j\ y_j\ 1]^T$. H is estimated from the correspondences between $\mathcal{P}$ and $\mathcal{F}$, by application of the Levenberg-Marquardt algorithm [26]. Note that H facilitates

projector placement as well, as it compensates for the potential perspective distortion, i.e. due to oblique placement.

A 2D polygon $\mathcal{Q}$ upon the P that outlines the spatial extent of interactive display, whether rectangular or arbitrary. $\mathcal{Q}$ restricts the search space for detection of fingertip contacts. This is achieved using H and the cooperation of the operator, who traces a finger at the limits of the interaction area.

In the case of multiple sensors, each one is independently calibrated as above. In this case, sensors are enumerated by k and the aforementioned calibration results are denoted as $\mathcal{Z}_k$, P_k, H_k, and $\mathcal{Q}_k$. Let $\mathcal{O}_n$ be the n intersections of polygons $\mathcal{Q}_k$. These intersections are utilized in Section 3.3 where touch events occurring in $\mathcal{O}_n$ are treated specially.

The area where touch detection is achieved, is determined by the exact camera posture, the sensor's FOV, and range $[\tau_a, \tau_c]$. In all experiments the Asus Xtion pro sensor, FOV $(58°, 45°)$ was employed within $\tau_a = 0.5\,m$, $\tau_c = 2.0\,m$. In an ideal sensor placement, this range forms an isosceles trapezoid $\mathcal{I}$ of $2.07\,m^2$ area (see Fig. 3). In practice, this area may become smaller according to how much the sensor placement deviates from the ideal. For rectangular displays, multiple configurations can be considered, depending on geometrical constraints of the surfaces and the projection. In Fig. 3, two characteristic configurations are illustrated: one with a portrait and one with a landscape orientation of the interactive surface. The landscape configuration is typical for displays and interactive surfaces, as it better matches the visual field and the reach of the user. In this case, it allows for a greater area to be covered $(1.1m^2)$ compared to the portrait configuration $(0.83\,m^2)$. However, in the portrait configuration, τ_a is set closer to the sensor, which exploits better the higher accuracy of the sensor in shorter ranges.

We assume an ideal placement for the top-view approach [40], when we compare it against our method in Section 4. The sensor is placed perpendicularly to the surface. The area covered is the base of the FOV's frustrum (see Fig. 1), at height d. The sensor is placed at a distance up to $d = 1.3\,m$ from the surface. For distances greater than $1.3\,m$, depth measurement becomes unreliable for finger touch detection [34] using a top-view

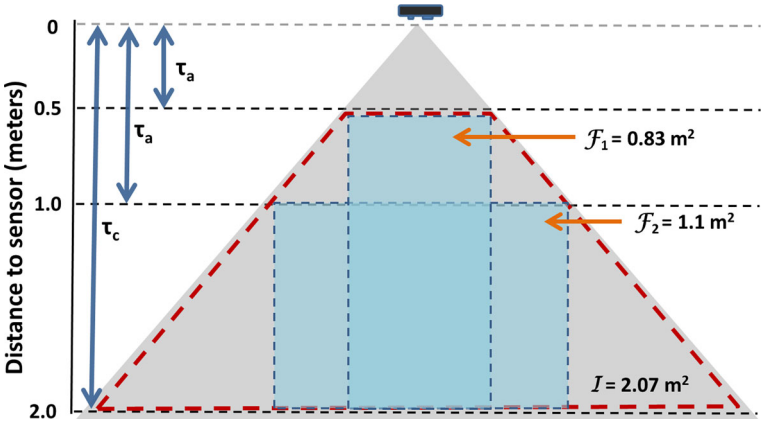

Fig. 3 Geometry of lateral sensor placement. Sensor FOV is shown in *light gray triangle* and effective finger touch detection zone $\mathcal{I}$ with *dashed red lines* has an area of $2.07\,m^2$. Rectangular subregions in $\mathcal{I}$ (i.e. $\mathcal{F}_1$, $\mathcal{F}_2$) can be defined to avail "portrait" or "landscape" interaction areas

placement. Thus for $d = 1.3\,m$, $\mathcal{B} = 1.55\,m^2$ compared to the $2.07\,m$ of the proposed approach.

3.2 Contact detection and localization

3.2.1 Image preprocessing

The proposed method employs only a small subset of the available pixels in the depth image. These pixels are looked up within $\mathcal{Z}$. During run-time, candidate pixels are further reduced based on the 3D-point interpretation of their depth values. In particular, only pixels within $\mathcal{Q}$ are considered ensuring that only surfaces within the defined workspace can trigger a touch event. In this way, arbitrary workspace shapes (besides rectangular) can be instantiated. The result is a set of 3D valid points, $\mathbf{p}_i$, henceforth enumerated by index i.

3.2.2 3D vision

Detecting distinct fingertips in contact is central to the proposed approach. Points $\mathbf{p}_i$ occur clustered in 3D space, around locations that correspond to user fingertips. The goal of this process is to find these clusters, group points $\mathbf{p}_i$ accordingly, and then estimate fingertip locations from the points of each such group.

A conventional foreground/background approach would cluster "foreground" pixels using Connected Component Labeling (CCL) in 2D. In our case, a 3D clustering technique is employed upon points $\mathbf{p}_i$ to find the fingertips in contact with the interaction surface. The CCL algorithm has been extended to $3D$ to apply the same label to two points which are closer than $3D$ distance τ_d. The following example provides intuition for this decision. Let the case where two fingers are imaged adjacently and connected, but are actually located far from each other on the surface (see Fig. 4). Blob detection in $2D$ would spuriously form a single group of labels, whose centroid occurs in-between the two fingers. The benefit of the $3D$ clustering approach over conventional, 2D blob detection on "foreground" pixels is that it resolves cases where two fingers at different depths are imaged as joined.

Detection candidates are filtered based on their horizontal spatial extent. This is defined as the horizontal distance of the two 3D points that correspond to the outer pixels of a point cluster, in the direction of $\mathcal{L}$. Clusters that exhibit a horizontal spatial extent outside the range $[\tau_s,\ \tau_b]$ of fingertip width (typically 10–30 mm) are rejected. Rejecting clusters smaller than τ_s filters spurious detections, due to noise that gives rise to spurious, minute clusters. Potential residuals of this filtering do not cause spurious detections, as they are further filtered, in Section 3.4, by the tracker module. Threshold τ_b safeguards for large surfaces in contact with the interaction surfaces, such as a palm or other irrelevant objects. The result is a set of valid 3D clusters, henceforth enumerated by j with centroids $\mathbf{c}_j$.

In the final step, the contact points of the detected fingertips are estimated and mapped to display coordinates as follows. Coordinate system transformation $\{R,\ \mathbf{t}\}$ is applied to $\mathbf{c}_j$ as $[x_j\ y_j\ z_j]^T = R\,\mathbf{c}_j + \mathbf{t}$. In this system, the z component of the transformed points is truncated. This effectively projects $\mathbf{c}_j$ on P. The resultant 2D points are finally transformed by H, to convert to display coordinates $\mathbf{q}_j = (u_j, v_j)$, where $[u_j\ v_j\ 1]^T = H\,[x_j\ y_j\ 1]^T$.

3.3 Multiple sensors

Multiple sensors can be utilized to cover a larger interaction area. We propose combining the sensors to expand across one of the interaction area's dimensions, either horizontally or

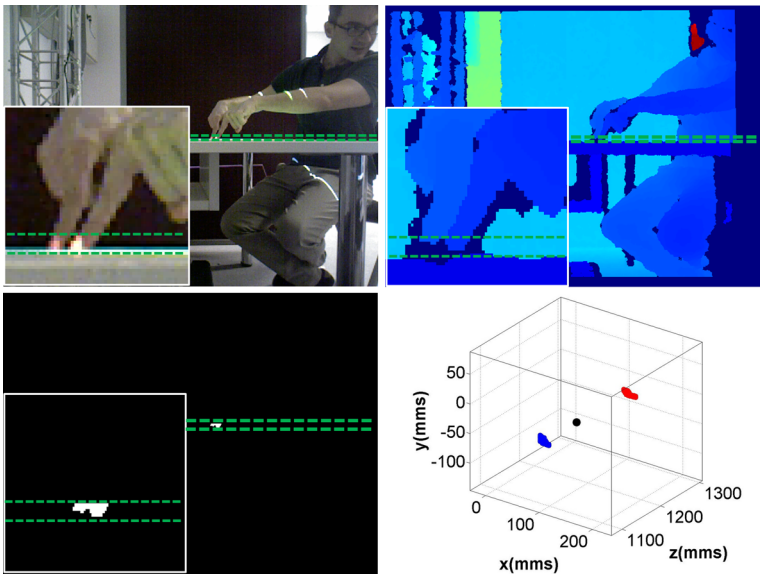

Fig. 4 Fingertip detection. *Top:* A case where two fingers are located far from each other but are imaged as attached. Superimposed, *dashed green rectangles* visualize $\mathcal{Z}$ on the RGB (*left*) and depth (*right*) images. *Bottom:* A 2D CCL detection approach (*left*) spuriously detects a single blob and an erroneous fingertip location, plot as a black sphere (*right*). In contrast, the proposed 3D approach (*right*) correctly forms two clusters of 3D points (*blue, red*). Thumbnails show in magnification the regions of interest of the original images

vertically. The proposed horizontal placement is shown in Fig. 5, resulting in a display with "landscape" aspect ratio. The alternative case of stacking multiple sensors vertically (not shown), in a "portrait" configuration is less practical, as the workspace extends beyond user reach.

In the proposed configuration problems related to sensor interference are not encountered. The reason is that their FOVs, and correspondingly their active illumination, do not overlap except for the "seam" regions, $\mathcal{O}_n$. In these regions, interference is also absent as the active illumination of the RGB-D sensor shines IR light on different faces of the finger.

Despite the lack of interference special treatment is required to cope with the multiple reconstructions of the same finger. Touch estimates in overlapping detection areas O_n, are imaged by two sensors and would, otherwise, give rise to multiple detections of the same fingertip. For this reason, the contact points detected from each sensor are gathered in a common data structure. Contact points occurring in regions $\mathcal{O}_n$, $n \in [1, k-1]$, are found, through point-in-polygon testing. For these points, the ones which are more proximate than distance threshold τ_o (e.g. 10 pixels) are associated to the same finger. These points are finally merged by computing their centroid.

The calibration scheme provided in Section 3.1 allows a mapping between one or more sensors and the display buffer F of the projector. This mapping is distinct for each sensor but, at the same time, allows multiple sensors to refer to the same reference system; the projector image coordinate system. Thereby the extrinsic calibration of sensors, typically required when multiple cameras are employed, is not necessary in this case. The proposed approach is compatible to the use of multiple projectors, as long as they refer to a common display buffer F.

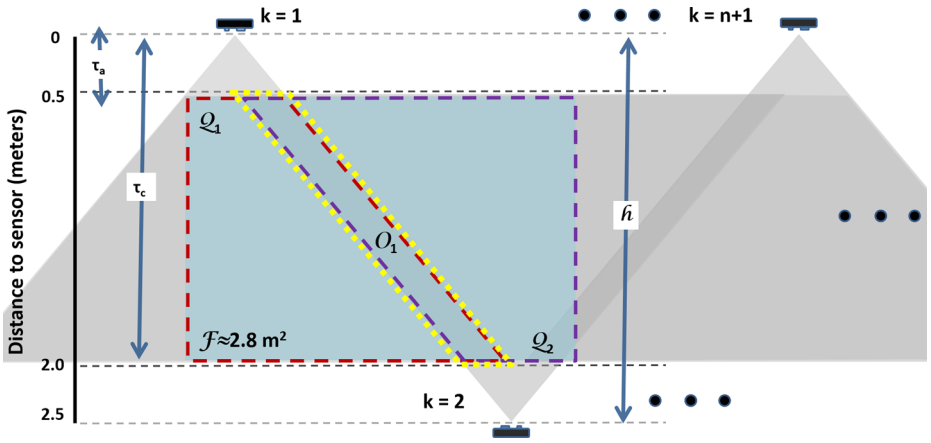

Fig. 5 Illustration of horizontal placement of k sensors. Sensors are placed equidistantly and against each other with a small overlap (O_1), to minimize depth interference between the sensors in $\mathcal{Q}_{1,2}$ polygons. The distance between sensors is $h = 2.5\,m$ while $[\tau_a, \tau_c] = [0.5\,m, 2.0\,m]$. This allows for finger touch detection, in a rectangular ($2.8m^2$, for $k = 2$) interaction area on the surface (marked in *blue*). *Black dots* indicate geometry at which more sensors can be added

3.4 Fingertip tracking and touch event creation

Contact detections along with their localization estimates are received by a tracking module. The 2D tracker assigns unique ids to $\mathbf{q}_j$ the first time detected and tracks their trajectories in consecutive frames. Touch points of the current frame are corresponded to touch points of the previous frame, based on proximity. These, temporal, correspondences are established with the closest touch location at the previous frame, given that its distance is below threshold τ_t; otherwise a new id is generated.

A 2D Kalman filter [5] is employed to better estimate location, and obtain smooth trajectories of detected fingers. As the filter provides a prediction of the next location of the tracked fingertip, it is also used to compensate for transient detection failures, due to noise or brief occlusions. Upon disappearance of a 2D touch point, the estimates of its next predicted state for a few frames are retained. Conversely, to compensate for spurious detections due to noise, a dwell time is required for detection. Thus, contact points are tracked, again, for a few frames before deemed as valid i.e. 4 frames or $\approx 33msec$ for image acquisition at $30\,Hz$.

Tracking of touch points is required to distinguish between single and multitouch touch events as well as avoid confusion in time-lasting events (e.g. "Press and hold"). The implemented touch events are *touchstart*, *touchmove* and *touchend*, of the *Windows 8 Touch Injection API*. The *API* transparently provides the emulated events to applications run by the operating system. The *touchstart* event is triggered when a touch point is detected on the touch surface and is deemed valid. The *touchmove* event when a previously detected valid touch point is still detected along the touch surface. The *touchend* event when a valid touch point is no longer tracked on the touch surface.

The benefit of producing native UI events compatible with the operating system is that no additional integration effort is required to utilize the system in virtually any application. Moreover, it enables the detection of higher level gestures based on these events, which are recognized by the operating system or 3^{rd} party software (i.e. [8]).

3.5 Limitations of the proposed approach

3.5.1 Occlusions

A limitation of the proposed approach is due to occlusions, in the sense that it cannot detect touch for fingers that are not visible to the sensor. As irrelevant objects are not present upon the interactive surface these occlusions are mainly self-occlusions of a hand. Occlusions may also occur when multiple hands interact with the surface.

The way and extent to which occlusions that affect system performance is relevant to the placement of the sensor and the posture of user hands. There are two prominent options to lateral sensor placement regarding the viewpoint from which hand interaction with the surface will be imaged. These options are determined by the orientation of the display, or otherwise, the preferred orientation of the user hand against the display. In this, preferred, orientation fingers are, approximately, parallel to the columns of the display. The two options, namely "frontal view" and "side view", are illustrated in Fig. 6. Clearly, the side view is more prone to self-occlusions, as in the preferred orientation fingers are likely to occlude each other. The comparison of these two options is the topic of the experiment in Section 4.3. Similarly, occlusions may occur between multiple hands when both are observed by the same sensor.

3.5.2 Sensor range, finger size, and localization accuracy

Finger size, in combination with the observation distance and sensor resolution, determines the apparent size, or the number of pixels by which a fingertip is imaged. It is thus, relevant to the reliability and accuracy of detection. At great distances, even within the sensor range, fingers are imaged in only a few pixels and sensor noise is more influential. Sensor signal is then unreliable and, as a consequence, reduced localization accuracy may be encountered.

Given sensor capabilities and plausible finger sizes ($.5\,cm$ to $1.5\,cm$) , we consider an operating range of distances $[t_a, r_c]$. In that range accuracy is acceptable for the use cases considered. This is the reason for truncating the upper distance range of sensor operation with threshold τ_c (at $\approx 2\,m$) instead of the actual sensor range (at $\approx 6\,m$). In this range of distances, fingers of a wide range of users, including adults and children, were sufficiently sensed by the system, as studied in experiments in Section 4.

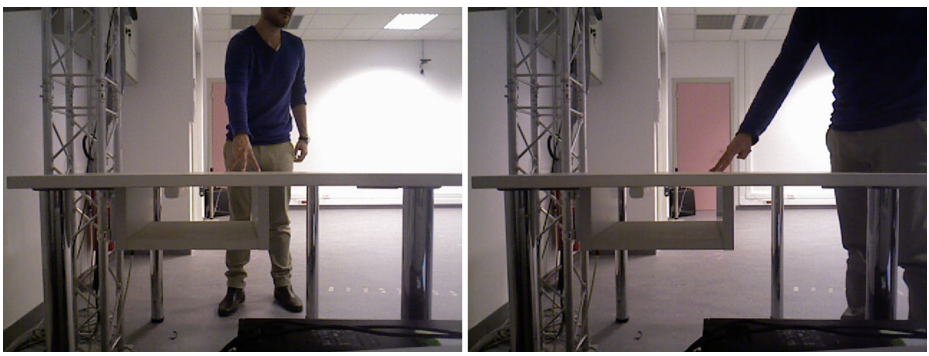

Fig. 6 Original RGB images for the two options of lateral camera placement: frontal (*left*) and side view (*right*)

4 Experiments

The primary axis of the experiments in this section is to assess the accuracy of the proposed method. Its advantages are also shown when employed in fingertip touch sensing for an interactive augmented display. In this context, its localization accuracy and practicality are comparatively evaluated against the widely employed, top-view approach in Section 4.1. Behavior of the method in terms of fingertip motions in the interactive area is also presented in that section. A secondary axis is explored in Section 4.2, which evaluates how accurately touch is detected, with respect to the actual distance of the fingertip to the interactive surface. It is there shown that the proposed method provides increased sensitivity to touch detection, compared to the top-view approach. Evaluation of multi-finger interaction is presented in Section 4.3. Experiments pertinent to the setup and performance of multiple sensor configurations are reported in Section 4.4. In that subsection, it is also shown that in practice the method is sufficiently reliable for touch detection regardless of the identity of fingers. Usability assessment of the proposed method in realistic applications is presented in Section 4.5.

The context of use of the proposed approach is that of an interaction surface, at the scale of arms reach, wide enough to support one or multiple users. The most prominent use cases refer to the interaction surface being a wall or a desk, as generic touch surfaces. In this context, interaction regards conventional uses (such as "click" or "tap"), arbitrary touch trajectories upon the surface (such as drawing a contour or "drag"), and simple gestures (such as zoom-in by dragging two fingers, upon the surface, away from each other). The aforememtioned use cases have been evaluated for multiple touches, whether these originate from multiple fingers of the same user or from multiple users. The farther extent of the operating range, τ_c, is sufficient as it matches with the extent of arms reach. Thus the main impact of limitations (see Section 3.5) is due to occlusions. If user fingers and hands occlude each other relative to the same sensor of observation, system performance deteriorates failing to detect the occluded contacts. The consequence is that cases of multi-touch interaction, where a single or multiple users intermingle their hands, may result in erroneous system behavior. In practice, this is a quite unlikely case, as on one hand, there are very few applications that support/require this type of input, while on the other hand, each user usually interacts with the part of the surface corresponding to his/her physical location. Beyond typical finger-touch interaction, two use cases explored the application of the proposed method as a means for digital games' control. In this exploration less conventional wall-based interaction types employing projectiles and hand-held props was evaluated, in addition to conventional interaction.

In all experiments 3 users, age $28-33$, were requested to perform tasks. An exception was made for the experiment in Section 4.1.2, where data were collected by 5 users, age $28-50$. All users were naive to the experimental hypotheses. In Section 4.5 usage assessment was conducted in a public installation; more than 100 adult and child users (60 % children, 40 % adults), interacted with the pilot applications. Sensor placement was approximate for experiments in Sections 4.1.1, 4.4 and 4.5. Sensor placement was ideal for experiments in Sections 4.1.2, 4.1.3 and 4.3. Finally, sensor was placed both in an approximate and an ideal position for the experiment in Section 4.2. When in the approximate configuration, the distance of the optical center of the sensors from the interactive surface was $\approx 5\,mm$.

All the experiments were performed using the *Asus Xtion Pro* RGB-D sensors on a conventional PC with a Intel i7 CPU at 2.93 GHz, 8 Gigabyte RAM and an Nvidia GTX 650 1 Gigabyte RAM. Depth cameras operated at 480×640 pixel resolution at 30 fps. The workspace was constrained by the operating range $[\tau_a,\ \tau_c]$ in $[0.5m,\ 2\,m]$ where sensory

input is reliable. System performance matched the input frame rate. In particular, our CPU implementation of the proposed method executes in $4.6\,msec$. This is faster compared to a contemporary GPU implementation of the top-view method which executes at $7.3\,msec$ for input frames of the same resolution [23]. Performance effectively remains the same for the multi-sensor case as the computation of touch points q_j is parallelized: a thread is committed to the computation q_j, for each sensor.

4.1 Localization accuracy

To assess the localization accuracy of the proposed approach, three experiments were conducted. In the first experiment, accuracy was evaluated in a wide area, that tested the limits of the operating range $[\tau_a, \tau_c]$. In the second experiment, accuracy was compared against that of the top-view approach, in a common workspace. In the third, accuracy was evaluated for arbitrary finger motions.

In Experiments 1 and 2, the projector displayed touch targets (minute dots) on the interaction surface and users were instructed to touch their centers with their fingers. Accuracy error was measured as the distance of the estimated touch location to the center of the dot in F, in pixels. The reported errors are the effective errors as they contain also the camera-projector calibration error (from the process in Section 3.1). In Experiment 3, the projector displayed patterns and users were requested to trace them with their fingers. Touch estimates were recorded, and superimposed upon the patterns in F.

4.1.1 Experiment 1

The accuracy of the proposed approach was tested on a wide spatial extent. An interactive surface was set on a wall, with a sensor placed close to the ceiling. The area of the interaction surface was $\approx 1.2\,m^2$. Range $[\tau_a, \tau_c]$ was $[0.6\,m, 1.9\,m]$. Covering an equivalent spatial extent with a top-view approach is impractical. A top-view approach on a wall would require placing the sensor opposite to the display where the user should be able to freely move.

The projector displayed a grid of 160 dots, radius 1 cm, and users were instructed to touch their centers. The mean fingertip localization error (and its standard deviation) was 3.02 (2.19), in pixels. Figure 7 plots the projected dots along with contact localization estimates for the 3 users and provides an image from the experiment. Figure 7 also illustrates the mean localization error and standard deviation for each dot over its distance to the sensor. To visualize the increasing trend, a curve is fit on the mean error. A second degree polyonym was utilized as the pixels imaging a fingertip are proportional to the reciprocal of its squared distance from the sensor. The error increases as the user interacts further from the sensor. Localization inaccuracy indicated was less than 2 pixels in short range and, on average, no more than 6 pixels of at $\approx 2\,m$ distance.

The system detected finger contact for all users touching the centers of all of the 160 illuminated dots without a failure. This shows that touch detection is reliable within the specified range $[\tau_a, \tau_c]$ for single finger interaction.

4.1.2 Experiment 2

In this experiment, two configurations, a lateral and a top-view, were implemented to image the same planar surface simultaneously, as in Fig. 1. The common workspace covered by both sensors was approximately $0.7\,m^2$ ($0.9 \times 0.8\,m^2$). Range $[\tau_a, \tau_c]$ was set to

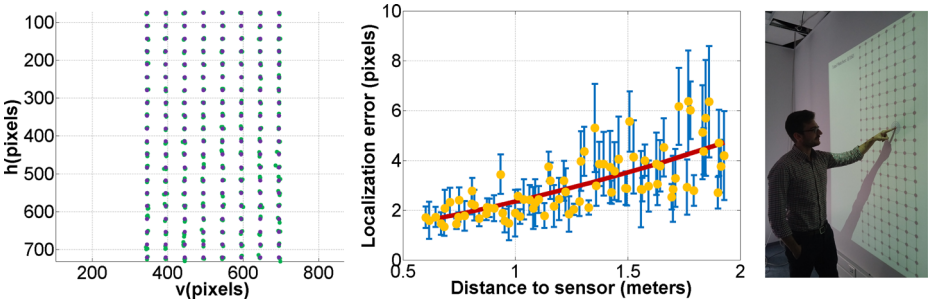

Fig. 7 Touch localization accuracy. *Left:* The plot illustrates the projected 160 target dots (*purple*) and the estimated contact locations (*green*) for 3 different users. *Top* of the figure corresponds to the ceiling, where sensor is placed, while *bottom* to the floor. *Middle:* Mean localization error (*yellow*) with standard deviation (*blue*) over distance for each dot. A second degree polyonym (*red*) fits the error over distance. *Right:* A user touching the target grid in a wall configuration where a depth sensor is placed close to the ceiling

$[1.0\,m, 1.8\,m]$ for the proposed method. In the experiment, the projector displayed a grid of 16 dots, with a radius of 1 cm on the interaction surface.

Users were instructed to touch the dots at their centers. At the same time, activation of the two sensors alternated, so the active illumination systems of the two sensors would not interfere. In this way, RGB-D frames were acquired for fingertip interaction from the two investigated viewpoints (lateral and top), almost simultaneously. The mean fingertip localization error (and its standard deviation) were 5.02 (2.62) pixels for the lateral and 5.36 (3.23) pixels for the top-view.

4.1.3 Experiment 3

The purpose of the experiment was to assess the method in arbitrary fingertip motions on an interactive surface. The setup was as in Section 4.1.2. As no common visible area with the top-view approach was required, the interaction surface was $1.08\,m^2$ ($1.2 \times 0.9\,m^2$), fully covered by the projector. Range $[\tau_a, \tau_c]$ was $[0.7\,m, 1.9\,m]$. The projector displayed two types of patterns, a spiral and a snowflake pattern. Patterns dimensions in the real world were $0.8 \times 0.8\,m^2$ and covered most of the interaction area. Users were instructed to trace them using their fingers. Touch estimates were recorded and superimposed on the projected display F (see Fig. 8). It is observed that interactive fingertip motions are consistent for all patterns. Thereby, the approach is considered sufficiently accurate for fingertip motion interaction.

4.1.4 Discussion

The higher accuracy observed in the first experiment, is attributed to the closer distance ($\tau_a = 0.6\,m$) of fingers compared to the second experiment ($\tau_a = 1.0\,m$). In the first experiment, the mean error of the proposed approach, encountered at $\approx 2\,m$ distance from the sensor, is ≈ 4 pixels. This is comparable to the mean accuracy of top-view approach in the second experiment. That is, even in the most distant case of contact, the proposed approach is more or equally accurate than the top-view approach.

Touch detection is quite reliable and neither detection failures nor detection false-positives were observed in the experiments. This is, in general, the case for unoccluded,

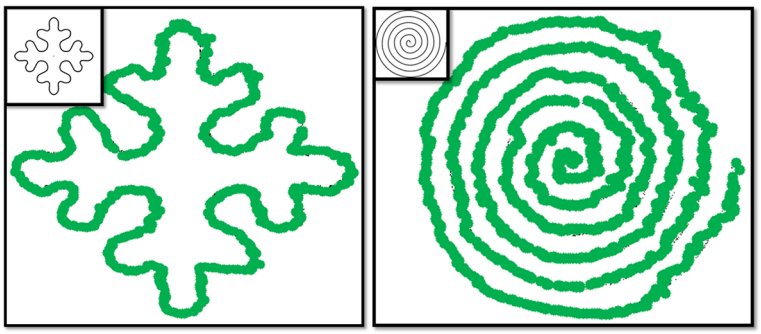

Fig. 8 *Bottom left, right:* Recorded touch estimates (*green dots*) for a user moving his finger aligned with a snowflake and spiral pattern respectively. Thumbnails illustrate the original patterns. Projection of the patterns covered most of the interaction area. Patterns dimensions in the real world were $0.8 \times 0.8\, m^2$ and $600 \times 600\ pixels$ on F. *Left* of the figure corresponds to the side where sensor is placed

single finger interaction within range $[\tau_a, \tau_c]$. Robustness to sensor noise, which in rare cases may transiently obstruct finger detection, is compensated by tracking.

Finally, it ought to be noted that fingertip motions in the interaction area were consistent and sufficiently accurate. The patterns required that users interacted with their fingers in a variety of postures, so to be able to trace the curved and complex trajectories.

4.2 Sensitivity

The purpose of this experiment was to compare sensitivity for the lateral and top-view approaches. The experiment uses the lowest possible value of threshold τ_h and τ_X for the lateral and top-view approach respectively, while preserving robust fingertip detection. In addition, for the lateral approach, experiments were conducted both for the ideal as well as the approximate position. The experimental setup was as in Section 4.1.2.

As in Section 4.1.1, a grid of 16 dots was projected, one dot highlighted at a time. The grid dimensions on the interaction surface were $[780 \times 780\, mm^2]$ and the distance between 2 non-diagonally neighboring points was $195\, mms$. On each dot we stacked 4 *Lego* blocks comprising 4 different heights configurations (3.16, 6.33, 9.5 and 19 mm). For each height, users were instructed to place the index finger at the top of the stack. Depth data were captured, from the two sensors, for each fingertip contact. The purpose of the Lego blocks was to ensure that the fingertip was at the same, known distance from the surface for both methods. Image acquisition was as in Section 4.1.2, to avoid sensor interference. At each location 4 contact detections were attempted, one for each height. Then, s was measured as the maximum height that a touch detection occurred out of these 4 attempts.

For the proposed method, acquired data were post-processed to remove depth pixels imaging the blocks. In particular, the blocks were of a particular color (green). Using the RGB image they were segmented and the corresponding pixels in the depth map cast as invalid. In this way, the case where the fingertip occurs at a known distance from the surface was simulated. Users were first requested to perform the tasks for the ideal position of the sensor. Another round of experiments was conducted for an approximate position of the sensor. For the top-view case no pre-processing was required, as the blocks were occluded by the finger.

In Fig. 9, the values of s for each of the grid's dots are presented. As the mean values of s for the lateral approach are smaller than for the top-view, it is concluded that the former yields greater sensitivity to touch. The mean values of s and its standard deviation were 4.55 (1.99) in mm for the lateral in an ideal sensor placement and 6.72 (2.27) in an approximate sensor placement. The error for the top-view was 11.08 (4.01). The results indicate greater sensitivity for the proposed approach, for both the conditions of the ideal and the approximate sensor placement. Moreover, this increased sensitivity is consistent across the interaction area for the ideal placement. This, slightly changes for the case of approximate placement, when interacting at more distant to the sensor areas. In these areas, distance to the lateral sensor is $\approx 2\,m$ but for the top view sensors is $\approx 1\,m$. Deviations for the ideal configuration mildly affect sensitivity for the most distant regions of the workspace. In these regions, where sensitivity for the proposed method is the worst, it is still better than the sensitivity of the top-view approach in all regions of the workspace. We conclude that comparatively to the conventional, top view, approach the proposed approach exhibits greater sensitivity.

4.3 Multiple finger interaction

The purpose of the experiment was to assess the suitability and limitations of the proposed method for multiple finger interaction. Though unobstructed single-finger interaction is quite reliable, multi-finger interaction is subject to distance limitations and self-occlusions. Regarding distance, fingers that are not in contact with each other may appear merged in the depth image. The effect is more pronounced as distance to the sensor increases. The setup was as in Section 4.1.2. As no common visible area with the top-view approach was required, the interaction surface was $1.08\,m^2$ ($1.2 \times 0.9\,m^2$), fully covered by the projector. Range $[\tau_a, \tau_c]$ was $[0.7\,m, 1.9\,m]$.

To study limitations of our approach with respect to distance, self-occlusions, and ergonomy, the frontal and side view configurations of lateral sensor placement were tested (see Section 3.5). The projector presented 3 parallel purple lines on the interaction surface, having distance $5\,cm$ from each other, in 3 regions, in a vertical and in a parallel orientation relative to the principal axis of the sensor (see Fig. 10). Users were instructed to trace the lines with their fingers. Touch estimates appeared as green dots in F.

In the frontal view configuration, self-occlusions do not occur and the system successfully detects the 3 fingers at close and middle ranges. When fingers interact close to $2m$,

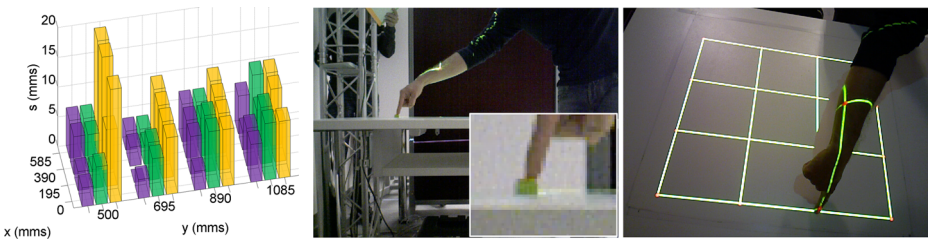

Fig. 9 Touch sensitivity. *Left: s* values for the lateral (*purple, green*) and top-view approach (*yellow*) on each of the 16 positions of the $[780 \times 780\,mm^2]$ grid. For the lateral approach, sensor was placed such that its principal axis was parallel with the y-axis of the grid and at $\approx 290\,mm$. The y-axis starts from $\tau_a = 500\,mm$. The lateral approach is more sensitive in touch detection compared to the top-view, both for the ideal (*purple*) as well as the approximate (*green*) placement. *Middle-Right:* Two simultaneous input images from the experiment, one from each sensor

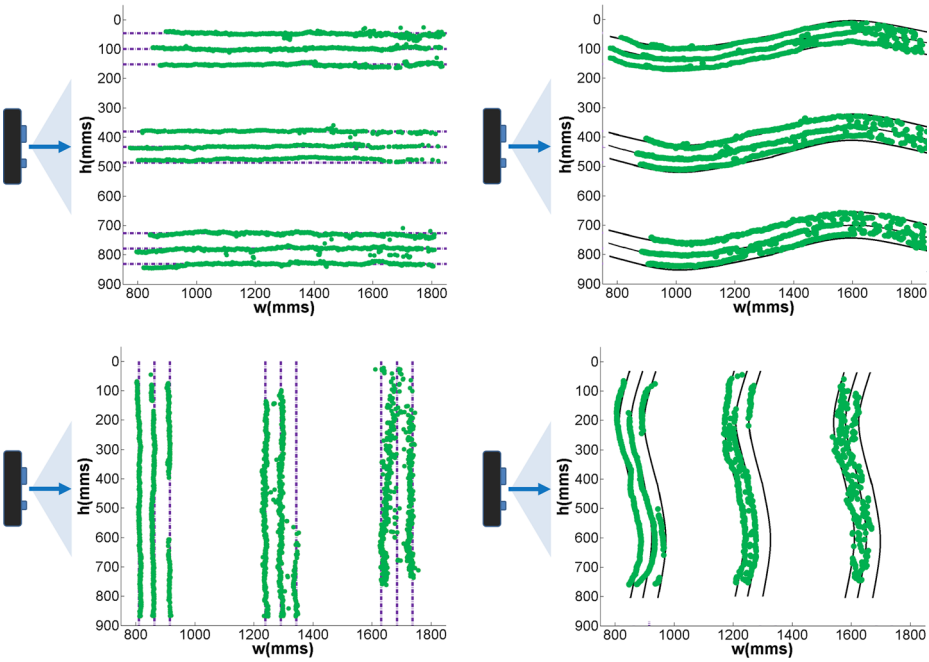

Fig. 10 Multitouch detection results, with the sensor placed in the frontal view (*top*) and on the side view (*bottom*) lateral placement configurations. *Green dots* plot the estimated coordinates in $\mathcal{F}$, while *purple dashed dots* the projected pivot line patterns. Approximate sensor placement is shown on the *left* of plots. Results are also presented for wave patterns. Failures produced from lack of resolution and depth uncertainty (*top right*) and self occlusions (*bottom right*) are more evident

there are cases where they are not well discriminated, as they appear merged in the image. In this case and at this distance, due to lack of resolution and depth uncertainty at greater distances (close to $2m$), the clustering technique (Section 3.2.2) may not discriminate between fingers in proximity. The result is a single cluster, thus, a single touch estimate, which is the reason for the green dots between lines in Fig. 10. An auxiliary experiment was also conducted to characterize the behavior of our method for multiple finger interaction for arbitrary motions. In particular, 3 wave patterns were presented in the interactive surface, in 3 regions. Limitations due to lack of resolution and depth uncertainty are more evident (see Fig. 10).

In the side view configuration, failures due to self-occlusions, i.e. a finger occluding another finger(s), are more often encountered. The first experiment was repeated, with the set of 3D lines appearing in an orientation vertical to the principal axis of the sensor. Due to the hand posture, self-occlusions occur even at close range (see Fig. 10). The effect is further pronounced as distance to the sensor increases, due to the aforementioned reasons. Similarly to the frontal view, an auxiliary experiment using wave patterns was conducted. Failures produced from self occlusions are more evident (see Fig. 10).

The results characterize distance and self-occlusion effects on the performance of the proposed approach. Moreover, the experiment further evaluates two lateral sensor placement configurations, that image the scene from frontal or side lateral views of the interactive surface. The former of the two is recommended as, in that case, self-occlusions are less often encountered.

4.4 Multiple sensor integration

The purpose of the following experiments was to test the behavior of the proposed method for the multiple sensor case. The experiments test in terms of localization accuracy as well as performance of interactive fingertip motions. They also characterize the behavior of the method in areas were sensors' FOVs overlap. In these areas, more than one touch estimate could be associated with a finger. Two sensors were employed to cover a wall surface. Range $[\tau_a, \tau_c]$ of our algorithm was set between $[0.7\,m, 1.7\,m]$, as we were limited by construction in the wall where our system was employed. The workspace area formed by $\mathcal{Q}_k$ union was $\approx 2\,m^2$.

4.4.1 Localization for two sensors

We assessed the localization accuracy for the multiple sensor case by conducting an experiment similar to the one described in Section 4.1.1. The projector displayed a grid of 176 dots, radius $1\,cm$. The area covered by the grid was approximately $1.6\,m^2$. Users were instructed to touch their centers. Users were also asked to randomly try all their fingers. The mean fingertip localization error (and its standard deviation) was 2.26 (1.58), in pixels. Fig. 11 plots the projected dots and localization estimates, as well as the trend of the error over distance, similarly to the single sensor experiment.

The system detected finger contact, for all users touching the centers of all of the 176 illuminated dots without a failure. Localization accuracy and detection results show that the approach is reliable for the multiple sensor case, both in $\mathcal{Q}_k$ as well as $\mathcal{O}_n$. $\mathcal{O}_n$ are the areas where sensors FOVs overlap. Finally, results indicate that the method exhibits equivalent performance for all fingers.

4.4.2 Touch trajectories for two sensors

The purpose of this experiment was to assess the performance of the proposed method in terms of interactive fingertip motions. It also characterizes the behavior of the method in areas were sensors' FOVs overlap. The projector displayed 3 type of patterns and $\mathcal{Q}_k$ and $\mathcal{O}_n$ areas were highlighted. The projector displayed a spiral pattern, a wave pattern and a checkerboard pattern. Users were instructed to trace them with their fingers within

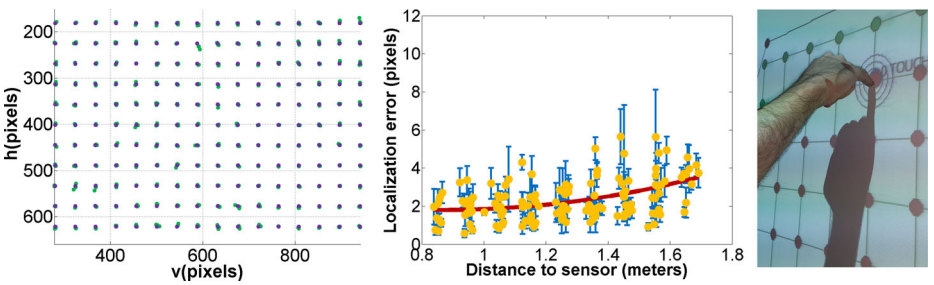

Fig. 11 Touch localization accuracy for a two sensor installation in a wall configuration. *Left:* The plot illustrates the projected 176 target dots (*purple*) and the estimated contact locations (*green*) for 3 different users. *Middle:* Mean localization error (*yellow*) with standard deviation (*blue*) over distance for each dot. A second degree polyonym (*red*) fits the error over distance. *Right:* A user touching the target grid using his pinkie finger

the highlighted areas. Touch estimates were recorded and superimposed on the projected display F (see Fig. 12).

It is observed that interactive fingertip motions are consistent for all patterns, arbitrary or straight lines. It is also observed that touch estimates during transitions from $\mathcal{Q}_k$ to $\mathcal{O}_n$ are smooth. Thereby, the combination of the two sensors is considered sufficiently accurate for the implementation of large interactive surfaces. The execution time of the algorithm essentially remains the same. The implementation is CPU-multi-threaded, where a CPU thread is dedicated to q_j computation for each sensor. Computational overhead of treating points in areas $\mathcal{O}_n$ is negligible mainly due to the small number of points (< 10) typically occurring in them.

4.5 Pilot applications

The proposed method was employed in three pilot applications. These were user-tested in a realistic setting in a public installation, during the TEDx Heraklion 2015 event. The same projector camera configuration was used in all three applications. In particular, two Xtion sensors are mounted on a wall surface, covering the projection of an interaction area $\mathcal{F} = 2.8m^2$ (see Fig. 5). The applications were used to evaluate user experience regarding 3 complementary attributes of touch interaction: (a) sensitivity, (b) speed and (c) robustness; using 3 different types of interaction triggers: (i) fingers, (ii) projectiles (e.g., plastic balls) and (iii) hand-held objects (e.g., sponge hammer). Figure 13 illustrates users interacting with the applications. Video demonstrations of application usage can be found at the supplementary material of this paper (Online Resource 1, 2 and 3). An overview of the applications characteristics and interaction requirements are illustrated in Table 1.

The first application, entitled Infocloud, is a multimedia public information system presenting a flowing stream of keywords, images and video thumbnails. Users can touch to select any of these items in order to retrieve related information. More specifically, when

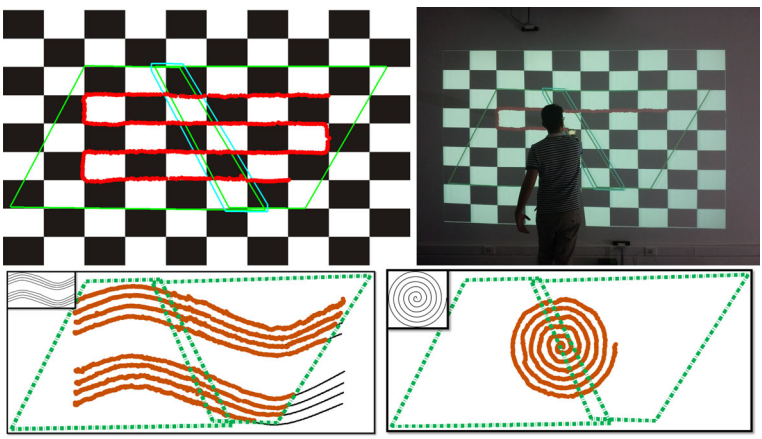

Fig. 12 Using two sensors to create a larger interactive display. *Top Left:* Touch estimates of a finger moving on a surface covered by two sensors. The *green polygons* plot $\mathcal{Q}_1$ and $\mathcal{Q}_2$ and the cyan $\mathcal{O}_1$. *Red dots* indicate the recorded touch estimates. *Top Right:* A user moving his finger aligned with the checkerboard lines within $\mathcal{Q}_1$, $\mathcal{Q}_2$. *Bottom left, right:* Recorded touch estimates (*red dots*) of a finger tracing across a wave and spiral pattern respectively. The *green polygons* plot $\mathcal{Q}_1$ and $\mathcal{Q}_2$ and the cyan $\mathcal{O}_1$. Thumbnails illustrate the original patterns

Fig. 13 *Left:* Application that showcases conventional touch functionality. Multimedia information (images, video) is presented using touch interaction. *Middle:* A game showcasing alternative use. The player swaps projected insects using a toy hummer. *Right:* An aiming game. The players throw toy balls on projected bricks. When the ball hits the projected digital brick, it breaks

an image is touched, it is magnified and a caption is appended to it. If a video thumbnail is selected, it is magnified and a play button is added allowing starting and stopping it. If a keyword is selected, a pop-up window appears showing a related illustrated piece of text. Selected items can be freely dragged and tossed around using a single finger. Furthermore, they can be closed through a button on their top right corner, marked with an 'X'. Since multi-touch is supported, multiple users can concurrently interact with multiple items. As the interactive items are constantly moving, speed and sensitivity are of paramount importance for this application. Speed, because the users should be able to instantly select the item residing underneath their fingertips before it moves away. Sensitivity, to avoid accidental selections that may occur when users approach their finger close to the wall in anticipation of an incoming item to be selected.

The other two applications belong to the domain of multiplayer digital games and showcase possible uses, other than touch detection, of the proposed methodology. One of the games, Debugger, is played using hand-held objects, like a sponge hammer (or even the players' fists) which are employed in order to hit small moving targets (a swarm of flying bugs). The bugs' behavior is influenced by user actions (e.g., whenever the wall is hit, they get "scared" and move faster) and game events (e.g., when a pie appears the get attracted around it). Occasionally a butterfly also appears. Players try to eliminate the bugs without hitting the butterfly. Detection speed is obviously important for this application. To detect objects greater than fingertips $[\tau_s, \tau_b]$ are appropriately increased. Also, as sensor framerate may not be fast enough to image the actual contact event, τ_h is increased, to ensure that the projectiles location is imaged right before or after the actual contact.

Table 1 Pilot applications characteristics and interaction requirements

Name	Touch targets	Input trigger	Interaction priority
Infocloud	Large, moving slowly	Fingertips	Sensitivity, speed
Debugger	Small, moving fast	Handheld objects (e.g. sponge hammer)	Speed
Breakout wall	Varying sizes, static	Projectiles (e.g. plastic balls)	Robustness in brief contact

The other game, called Breakout Wall, employs projectiles, such as toy balls of different size and material. During gameplay, various types of walls are displayed comprising brick of diverse sizes and "materials". For example, some bricks require several hits to be broken. Others hide bonus items, which if hit provide players with some additional capability (e.g., canon, bomb) or reward (e.g., extra life, more time). Players must aim and hit the bricks and bonus items using their projectiles. For this game, the most important detection aspect is robustness in cases of brief contact. The reason is that most users throw the balls using the top of their strength, resulting in rapid bouncing of the object against the wall. To be able to support such rapid bounces, the values of $[\tau_s, \tau_b, \tau_h]$ are set appropriately.

In both cases, the system behaved very well, supporting smooth and robust gameplay. All applications received positive feedback, as interaction was intuitive and robust, indicating the suitability of the method for creating interactive wall-sized displays. When asked, users commented that the overall feeling was natural and pleasant. The fact that in the 2 games people were able to hit the "display" using various means, created a lot of excitement to players of all ages, since this is an action that is typically prohibited for any other common type of interactive surface. The only problem encountered occurred when children (or shorter people) wanted to touch or hit a target located beyond their reach, at an area covered by the sensor that was placed at the bottom of the projection. In order to reach that point, they might place their arm or body against the wall, thus temporarily blocking the bottom sensor's view.

5 Conclusion

An approach is presented for touch detection and localization upon planar surfaces using a depth camera. In contrast to conventional approaches, where the sensor is placed above the interaction surface, in the proposed approach the sensor is placed laterally to this surface. Correspondingly, an algorithmic approach for touch detection and localization is proposed, for the visual input acquired in this configuration.

Several conclusions are drawn from the experiments in Section 4 that evaluate the proposed approach comparatively to the conventional, top view approach. A direct conclusion from the experiments in Section 4.1 is that the proposed method exhibits greater practicality in terms of sensor placement. It also, covers a greater interaction area than the top-view approach. Moreover, it is shown that the proposed method has equivalent or better accuracy than the top-view approach. The reason that top-view approaches do not achieve better localization, is traced back to the fact that they do not directly image the contact event. In top view approaches, the fingertip has to be segmented in the depth map, a process that contains significant uncertainty. This uncertainty stems from that, when in contact, the depth values of the fingertip are very similar to those of the surface.

For the sensitivity experiments, in Section 4.2, it is concluded that lateral viewing provides better sensitivity to fingertip contact. This promotes user experience as system response better matches actual fingertip contact with the interaction surface.

Multitouch interaction is supported as long as user(s) do not engage unlikely complex and intermingled configurations with their hands and fingers. In this respect, the options of the lateral placement are investigated in Section 4.3. The conclusion is that lateral placement of the sensors "from above" is preferable, as self-occlusions are even less likely to occur

in that configuration. In Section 3.3, it is shown that multiple sensors can be combined in order to create larger interactive displays without performance reduction, or problems at the seams between adjacent sensors.

The pilot applications, in Section 4.5, demonstrate the suitability of the method for contact based interaction upon the augmented display, either this is carried out with fingers or other means. In particular, it allows for physical objects, projectiles or hand-held, without any assumption on the type of objects, while it also provides a brisk, sensitive, and reliable response. In contrast, a top-view approach would require modeling of these objects, in order to infer contact with the interaction surface.

Future work, regards imaging the interaction surface from multiple views in order to better cope with occlusions. Another future goal is the combination of this work with a hand-tracker, i.e. [29], in order to better assess hand posture and provide richer means of interaction.

Acknowledgments This work has been supported by the FORTH-ICS internal RTD Programme "Ambient Intelligence and Smart Environments".

References

1. Agarwal A, Izadi S, Chandraker M, Blake A (2007) High precision multi-touch sensing on surfaces using overhead cameras. In: IEEE international workshop on horizontal interactive human-computer systems, pp 197–200
2. Benko H, Jota R, Wilson A (2012) Miragetable: freehand interaction on a projected augmented reality tabletop. In: SIGCHI conference on human factors in computing systems, pp 199–208
3. Bhalla M, Bhalla A (2010) Article: comparative study of various touchscreen technologies. Int J Comput Appl 6(8):12–18
4. Bimber O, Raskar R (2005) Spatial augmented reality: merging real and virtual worlds. A. K. Peters, Ltd., Natick
5. Bishop CM (2006) Pattern recognition and machine learning. Springer
6. Dietz P, Leigh D (2001) Diamondtouch: A multi-user touch technology. In: ACM symposium on user interface software and technology, pp 219–226
7. Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
8. Gesture Works http://gestureworks.com/
9. Han J (2005) Low-cost multi-touch sensing through frustrated total internal reflection. In: ACM symposium on user interface software and technology, pp 115–118
10. von Hardenberg C, Berard F (2001) Bare-hand human-computer interaction. In: Workshop on perceptive user interfaces. ACM, New York, NY, USA, pp 1–8
11. Harrison C, Benko H, Wilson A (2011) Omnitouch: wearable multitouch interaction everywhere. In: ACM symposium on user interface software and technology, pp 441–450
12. Hartmann G, Wunsche B (2012) A virtual touchscreen with depth recognition. In: Australasian user interface conference, pp 39–48
13. Hilliges O, Kim D, Izadi S, Weiss M, Wilson A (2012) Holodesk: direct 3D interactions with a situated see-through display. In: Human factors in computing systems, pp 2421–2430
14. Jones B, Sodhi R, Campbell R, Garnett G, Bailey B (2010) Build your world and play in it: interacting with surface particles on complex objects. In: IEEE international symposium on mixed and augmented reality, pp 165–174
15. Jones B, Sodhi R, Murdock M, Mehra R, Benko H, Wilson A, Ofek E, MacIntyre B, Raghuvanshi N, Shapira L (2014) RooMalive: magical experiences enabled by scalable, adaptive projector-camera units. In: ACM symposium on user interface software and technology, pp 637–644
16. Katz I, Gabayan K, Aghajan H (2007) A multi-touch surface using multiple cameras. Springer, pp 97–108

17. Kim J, Park J, Kim H, Lee C (2007) HCI (human computer interaction) using multi-touch tabletop display. In: IEEE pacific rim conference on communications, computers and signal processing, pp 391–394
18. Kjeldsen R, Pinhanez C, Pingali G, Hartman J, Levas T, Podlaseck M (2002) Interacting with steerable projected displays. In: Automatic face and gesture recognition, pp 402–410
19. Klompmaker F, Fischer H, Jung H (2012) Authenticated tangible interaction using RFID and depth-sensing cameras. In: International conference on advances in computer-human interactions, pp 141–144
20. Klompmaker F, Nebe K, Fast A (2012) dSensingNI: a framework for advanced tangible interaction using a depth camera. In: International conference on tangible, embedded and embodied interaction, pp 217–224
21. Koutlemanis P, Ntelidakis A, Zabulis X, Grammenos D, Adami I (2013) A steerable multitouch display for surface computing and its evaluation. Int J Artif Intell Tools 22(06):13600,161
22. Leibe B, Starner T, Ribarsky W, Wartell Z, Krum D, Weeks J, Singletary B, Hodges L (2000) Toward spontaneous interaction with the perceptive workbench. IEEE Comput Graph Appl 20(6):54–65
23. Margetis G, Zabulis X, Ntoa S, Koutlemanis P, Papadaki E, Antona M, Stephanidis C (2014) Enhancing education through natural interaction with physical paper. Univ Access Inf Soc:1–21
24. Matsushita N, Rekimoto J (1997) Holowall: designing a finger, hand, body, and object sensitive wall. In: ACM symposium on user interface software and technology, pp 209–210
25. Michel D, Argyros AA, Grammenos D, Zabulis X, Sarmis T (2009) Building a multi-touch display based on computer vision techniques. In: IAPR conference on machine vision applications, pp 74–77
26. Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer, New York
27. Ntelidakis A, Zabulis X, Grammenos D, Koutlemanis P (2015) Lateral touch detection and localization for interactive, augmented planar surfaces. In: International symposium on visual computing
28. Oikonomidis I, Kyriazis N, Argyros A (2011) Efficient model-based 3d tracking of hand articulations using Kinect. In: British machine vision conference, pp 101.1–101.11
29. Oikonomidis I, Kyriazis N, Argyros A (2011) Efficient model-based 3d tracking of hand articulations using kinect. In: British machine vision conference (BMVC 2011), vol 1. BMVA, Dundee, UK, pp 1–11
30. Rakkolainen I, Palovuori K (2005) Laser scanning for the interactive walk-through fogScreen. In: ACM symposium on virtual reality software and technology, pp 224–226
31. Rekimoto J (2002) Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In: SIGCHI conference on human factors in computing systems, pp 113–120
32. Saponas S, Harrison C, Benko H (2011) Pockettouch: Through-fabric capacitive touch input. ACM, New York, NY, USA
33. Schoning J, Brandl P, Daiber F, Echtler F, Hilliges O, Hook J, Lochtefeld M, Motamedi N, Muller L, Olivier P, Roth T, von Zadow U (2008) Multi-touch surfaces: a technical guide. Tech rep
34. Smisek J, Jancosek M, Pajdla T (2011) 3D with kinect. In: IEEE international conference on computer vision workshops, pp 1154–1160
35. Song P, Winkler S, Gilani S, Zhou Z (2007) Vision-based projected tabletop interface for finger interactions. In: ICCV, lecture notes in computer science, vol 4796. Springer, pp 49–58
36. Streitz N, Tandler P, Müller-Tomfelde C, Konomi S (2001) Roomware: towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds. Human-computer interaction in the New Millenium, Addison Wesley, pp 551–576
37. Takeoka Y, Miyaki T, Rekimoto J (2010) Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. In: ACM international conference on interactive tabletops and surfaces. ACM, New York, NY, USA, pp 91–94
38. Walker G (2011) Camera-based optical touch technology. Information Display 3:30–34
39. Wilson A (2005) Playanywhere: a compact interactive tabletop projection-vision system. In: ACM symposium on user interface software and technology, New York, NY, USA, pp 83–92
40. Wilson A (2010) Using a depth camera as a touch sensor. In: ACM international conference on interactive tabletops and surfaces, New York, NY, USA, pp 69–72
41. Wilson A, Benko H (2010) Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In: ACM symposium on user interface software and technology, pp 273–282
42. Xiao R, Harrison C, Hudson S (2013) Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In: Human factors in computing systems, pp 879–888
43. Zabulis X, Baltzakis H, Argyros A (2010) Vision-based hand gesture recognition for human-computer interaction. In: Stephanidis C (ed) The universal access handbook, chap 34. Lawrence Erlbaum Associates, Inc, pp 34.1–34.30
44. Zabulis X, Koutlemanis P, Grammenos D (2012) Augmented multitouch interaction upon a 2-DOF rotating disk. In: International symposium on visual computing, pp 642–653

Antonios Ntelidakis has been working as a Research and Development engineer at the Institute of Computer Science - Foundation for Research and Technology, Hellas (FORTH) as of 2010. He received his M.Sc. in the domain of Artificial Intelligence from the University of Edinburgh, Scotland, U.K. in 2010. He received his B.Sc. degree in Computer Science from the University of Crete, Greece in 2009. His interests include but are not limited to Computer Vision, Machine Learning, Robotics, Augmented Reality and Human Computer Interaction.

Xenophon Zabulis is a principal researcher at the Institute of Computer Science - FORTH. He received his Ph.D. in Computer Science from the University of Crete, Greece, in 2001. From 2001 until 2003 he was a Postdoctoral Fellow at the GRASP and at the IRCS laboratories, at the University of Pennsylvania, USA. During 2004 to 2007, he was a Research Fellow at the Institute of Informatics and Telematics - CERTH, Greece. His research interests include 3D reconstruction, pose estimation, medical image analysis and visual estimation of human motion.

Dimitris Grammenos is a Principal Researcher at the Institute of Computer Science (ICS) of the Foundation for Research and Technology - Hellas (FORTH). He is the lead interaction designer of the Human-Computer Interaction (HCI) Laboratory, specializing in the domains of Ambient Intelligence Environments, Public Information Systems & Interactive Installations, User Experience Design and Universal Access.

Panagiotis Koutlemanis received his B.Sc. degree in Music Technology and Acoustics from the Technological Educational Institute of Crete, with a major in virtual reality using binaural audio, in 2008. He has worked as a developer for the Technological Educational Institute of Crete from 2005 to 2008. Since 2009, he has been working as a developer at the Institute of Computer Science - Foundation for Research and Technology, Hellas (FORTH), participating in research programs in the field of Ambient Intelligence.