Augmented Multitouch Interaction upon a 2-DOF Rotating Disk

Xenophon Zabulis, Panagiotis Koutlemanis, and Dimitris Grammenos

Institute of Computer Science, Foundation for Research and Technology - Hellas, Herakleion, Crete, Greece

Abstract. A visual user interface providing augmented, multitouch interaction upon a non-instrumented disk that can dynamically rotate in two axes is proposed. While the user manipulates the disk, the system uses a projector to visualize a display upon it. A depth camera is used to estimate the pose of the surface and multiple simultaneous fingertip contacts upon it. The estimates are transformed into meaningful user input, availing both fingertip contact and disk pose information. Calibration and real-time implementation issues are studied and evaluated through extensive experimentation. We show that the outcome meets accuracy and usability requirements for employing the approach in human computer interaction.

1 Introduction

The emerging trend of smart environments entails the need for direct interaction with non-instrumented physical surfaces. In this context, often referred as "surface computing" [1], systems combine the projection of a user interface on a surface (e.g., tabletop, wall), with visual sensing of finger contacts with the surface to provide multi-touch interaction. The use of non-instrumented surfaces simplifies the application and maintenance of such systems. Recent availability of consumer depth cameras has reinforced the interaction capabilities upon virtually any surface, as the shape of interaction surfaces and the location of user hands can be accurately found in 3D.

We explore the issues arising when enabling augmentation and interaction upon a non-instrumented dynamically moving surfaces and, in particular, upon a disk surface mounted on two concentric gimbals, providing two degrees of freedom (DOF). Besides the provision of augmented multitouch interaction, such an achievement can serve two distinct functions. On one hand, the rotation mechanism can be used as a means for easily and intuitively browsing and interacting with alternative, dynamically changing, projection views. On the other, the high flexibility and extensive range of projection poses supported by the system can be used in order to dynamically personalize the physical properties of an interactive projection surface to the ergonomic preferences and needs of users.

The system employs a circular planar surface, or a disk, mounted on two gimbals (see Fig. 1). The outer gimbal rotates about the vertical axis at an angle θ (yaw). The inner gimbal is horizontal, dependent on the outer gimbal,

G. Bebis et al. (Eds.): ISVC 2012, Part I, LNCS 7431, pp. 642–653, 2012.

and rotates along with it; the disk is mounted at joints $\boldsymbol{q}_{1,2}$, and has an elevation (pitch) angle of ϕ. These axes intersect at the center $\boldsymbol{c}$ of the disk. A projector above the disk fully covers it with its projection. A depth camera overlooks this scene acquiring its depth map D. The two gimbals can be freely rotated by the user. Due to the finite resolution of the camera and the projector, the operation of the system is limited at very oblique angles as, then, the disk corresponds to very few pixels in the camera and the projector.

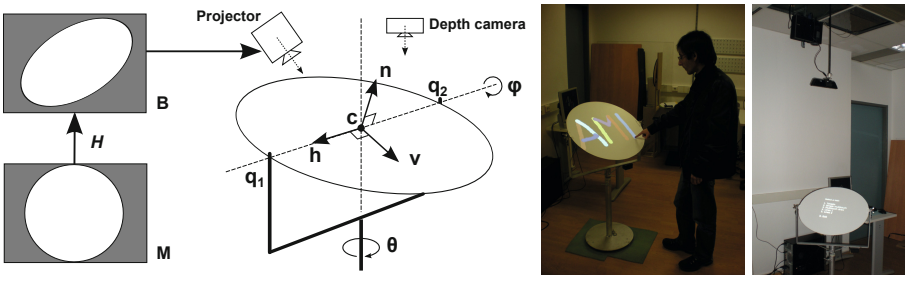

Fig. 1. System overview. *Left, middle:* illustration of system geometry and dataflow. The virtual display M is mapped upon the disk devoid of projective distortions by transforming M, according to the pose of the disk, prior copying it to the projector buffer B. Fingertip contacts are mapped back in M as multitouch events. *Right:* two photographs of a user testing a "finger-paint" pilot application, where drawings remain on the disk regardless of its posture, and an image of the system setup.

The system creates the following user interface metaphor. Let M be a virtual display buffer that renders a circular display.[1] The projector renders M upon the disk devoid of projective distortions, as if M was a multitouch display upon the disk's surface. This is achieved by continuously estimating the disk's pose and updating the projection appropriately. The pitch and yaw estimates can be used as additional user interfaces, by associating their values to (two) application variables.

Thus, a central system component is the real-time estimation of the disk's pose from depth map D. At each camera frame, depth map D is updated yielding a new pose estimate. To support brisk and accurate interactivity upon the surface, it is essential that this operation is performed in real-time and robustly.

By distorting M according to this pose, prior its copy it into the projector's pixel buffer B, M appears undistorted on the disk. Intuitively, rotating the disk while M is displaying a static image would create the illusion of the image being "painted" upon the disk. In the projection, the row axis of M is aligned with the disk's intrinsic horizontal rotation axis $\boldsymbol{h}$. Correspondingly, the column axis $\boldsymbol{v}$ lies on the disk surface and is perpendicular to $\boldsymbol{h}$. When user hands are in contact with the surface, touch events are generated and attributed with the 3D

[1] As displays are rectangular, the display region outside the circle is inactive.

contact coordinates of this contact. These coordinates are transformed into M's, 2D, reference frame, implementing a multitouch display upon the disk.

The remainder of this paper is organized as follows. In Sec. 2 related work is reviewed. Disk pose estimation is described in Sec. 3. A way to spatially calibrate of the above setup is proposed in Sec. 4. The display and contact estimation modules are described in Sec. 5. In Sec. 6, experiments which evaluate the accuracy, performance and usability of the approach are presented. In Sec. 7, conclusions and directions for future work are provided.

2 Related Work

To date, several touch-based interactive surfaces exist both in the form of research prototypes [2–7], but also as commercially available products [8–10]. Such systems use a static planar surface as an interaction surface.

In early approaches towards augmented interactive surfaces [11, 12], the dynamic component concerned steering a projector to display upon the surface of choice. The display was adapted using prior modeling of the surfaces and limited hand interaction was based on the input of a color camera. More recent approaches have used a more detailed 3D model of the whole scene, availing the ability to project at virtually any geometry of surfaces [13], but used a stylus instrumented with an infra-red beacon to enable single user interaction. The recent growth of depth cameras enabled the dynamic modeling, augmentation and touch interaction upon arbitrary surfaces [14]. Still, a training time is required in order for the system to model the interaction surfaces which, additionally, should remain static during interaction. The proposed approach constantly estimates the interaction surface and, thus, does not require an adaptation time.

Another aspect of dynamically moving interaction surfaces is that the location or pose of the surface itself can avail valuable information to the user interface. In [15], a coarse estimate of the inclination of a handheld surface (a piece of cardboard) provides input to an interactive game. In [16, 17], a similar surface is used to explore a maps. As the above approaches use a conventional camera, they offer limited (or none) touch interaction. To estimate surface pose, they rely on visual markers, thus being sensitive to marker occlusions by user hands and illumination artifacts. The proposed approach overcomes such limitations using depth information and, furthermore, is able to support multitouch interaction.

3 Disk Pose Estimation

Disk pose estimation, is used both in the real-time operation as well as the calibration of the proposed system. It is comprised of the two processes described below. The first estimates the plane that the disk lies upon, despite outliers arising from user hands and noisy pixels in D. To meet real-time requirements, the method is parallelized in the GPU. The second disambiguates disk yaw when the disk is approximately parallel to the ground plane.

3.1 Parallel and Robust Plane Estimation

To estimate disk pose the plane $\mathcal{E}$ is robustly fit to 3D points originating from depth map D. Only points originating from an elliptical region of interest (ROI) within depth map D, are considered. This ROI is large enough to image the entire disk and is predicted from camera calibration and disk geometry (see Sec. 4.1). Plane $\mathcal{E}$ has equation $\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{c}) = 0$, where $\boldsymbol{n}$ is the normal of the plane. The spherical coordinates of $\boldsymbol{n}$, ϕ and θ, indicate disk pose. A rotation of π about the horizontal axis is considered to bring the disk to the same posture and, thus, $\phi \in [0, \pi/2)$ and $\theta \in [0, 2\pi)$, where $\phi = 0$ corresponds to a posture parallel to the ground plane.

Significant amounts of outlier points are included in the data. Some occur due to sensor noise. Others occur as within the ROI more surfaces besides the disk are imaged, such as the user's hands or body. A robust plane estimation is obtained using RANSAC [18]; a threshold of 1 *cm* distance to the plane is set to characterize a point as an inlier.

Conventionally, RANSAC iterates by selecting a random triplet of points and evaluating the number of inliers for the plane they define, until it finds a good fit or a maximum number of iterations is reached. The method is parallelized in the GPU by performing all trials in parallel threads and selecting the triplet with the most inliers. Finally, using least squares, a plane is fit to all the inlier points to this plane which constitutes the final result. By convention, the normal of this plane is set to be in the direction of gravity.

A singularity, known also as the "gimbal lock" problem, is met when the $\phi = 0$. Then, any value of θ produces the same $\boldsymbol{n} = [0\,0\,1]^T$ and, in this case, the value of θ is determined using the method in Sec. 3.2.

3.2 Horizontal Axis Estimation

To disambiguate its pose when ϕ is approximately 0, the horizontal axis of the disk $\boldsymbol{h}$ is found by estimating the line through the 3D locations of the inner gimbal joints, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. These joints are detected in D and their 3D locations extracted by the corresponding 3D depth values.

Candidate points for this detection are sought in the periphery of the ellipse (or circle) at which the disk appears in D. As above this ellipse is predicted for the particular pose from camera calibration and disk geometry (see Sec. 4.1). In addition, the 3D coordinates of candidate points are required to approximately validate the current plane equation $\mathcal{E}$. In D, candidate points are grouped into blobs by a Connected Component Labeling process. Each blob is represented by its centroid, which comprises a candidate point.

Spurious candidates can arise from user hands occurring at the periphery of the disk. The pair of centroids selected as $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ is the one at which the two candidates are diametrically opposed across the disk center $\boldsymbol{c}$. By considering the 3D coordinates of the candidate pair of centroids, the pair that defines the line segment with the least distance from $\boldsymbol{c}$ is selected.

As this line does not specify the direction of the horizontal axis, this is determined to be the same with that of the previous frame, assuming that the

framerate of the system is frequent enough for the user to perform a rotation of π about the vertical axis during a single frame. The assumption is reasonable as the framerate of the system operation is $60\,Hz$ (see Sec. 6.

4 Calibration

System calibration includes the estimation of center $\boldsymbol{c}$ and the spatial modeling of the projection, both in the depth camera's coordinate frame. An additional color camera is used during the second part of this calibration (see Sec. 4.2) providing image I; we conveniently use the *Kinect* depth camera that already incorporates this additional sensor. A calibration of the depth and color cameras is assumed, based on [19].

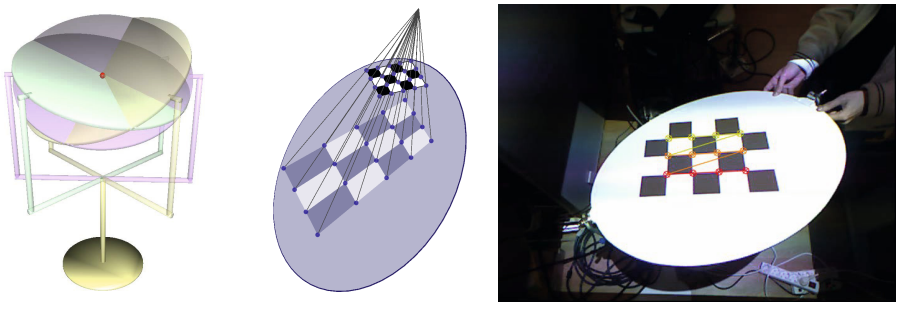

Fig. 2. Calibration geometry. *Left:* The intersection of all plane estimates during calibration yields an estimate of center $\boldsymbol{c}$. *Middle:* By projecting a calibration pattern and finding the 3D coordinates of its reference points, the projector is calibrated. *Right:* The projected pattern detected in image I.

4.1 Disk Center

Center $\boldsymbol{c}$ is estimated from the depth maps acquired, while the disk is freely rotated about both its axes.

At each frame, the method in Sec. 3.1 estimates the plane approximating the disk. Angles ϕ and θ are discretized, at a step of $1°$, yielding $k = 360 \times 90$ potential planes. A $k \times 4$ matrix, is employed as a lookup table (LUT) and each time a plane is estimated its 4 parameters are copied into the corresponding LUT entry. Thus, very similar planes are considered once.

The result is obtained as the intersection of all, let m, estimated planes and computed as the point that minimizes the sum of its squared distances from all estimated planes. As $\boldsymbol{c}$ should ideally validate the equations of all planes we seek $\boldsymbol{x}$ so that $||A\boldsymbol{x} - B|| = 0$ is minimized, where A is a $m \times 3$ matrix containing in each row the first 3 parameters α_i, β_i, γ_i of the estimated planes, B is a $m \times 1$ matrix containing values $-\delta_i$ in each row, and $i \in [1, m]$. A least-squares solution is then found through the SVD decomposition of A. Typically, less than a minute is required to fill a sufficient proportion of the LUT ($\approx 20\%$) for an accurate estimate of $\boldsymbol{c}$.

4.2 Projector Calibration

The projector operation is modeled by a 3×4 perspective projection matrix P that predicts the pixel coordinates in B that will illuminate a given homogeneous point in 3D space. Given the high quality optics of projectors and the usage of only its central portion of the display, the lens distortion of the projector is assumed to be negligible. During the calibration process both color I and depth D images of the sensor are used. Using their calibration, a registration of these two images is obtained and, thus, 3D coordinates of the surfaces imaged in the color camera can be obtained.

Matrix P is estimated as follows. The image of a calibration target, a checkerboard, is constantly projected upon the disk. This image appears distorted upon the disk, according its posture. As in Sec. 4.1, the disk is freely rotated in both angles, while the system is acquiring frames. For each frame, the projected checkerboard upon the disk surface is detected in I and the image coordinates of its corners identified using a checkerboard detector [20]. The 3D coordinates of these points are then found at the same pixel coordinates of, registered, image D.

Using then these 2D-3D correspondences the method in [21, p.181] estimates matrix P. Inclusion of lens distortion in this optimization is left for future work.

5 Real-Time Application

The real-time system is comprised of three modules that operate at the camera's framerate. Using the module in Sec. 3.1 a pose estimate of the disk is availed for each camera frame. Using this estimate, the transformation that undistorts display M upon the disk is computed. At the same time, a second module estimates finger contacts with the disk and produces multitouch user interface events in M's, 2D, coordinate frame.

5.1 Display

This module computes the distortion that M must undergo before projection so that (i) it appears undistorted upon the disk and (ii) each of its pixels illuminates the same physical region upon the disk, regardless of its posture. The distortion that an image undergoes when projected upon the disk is modeled through a 3×3 homography matrix, let H.

The homography can be defined if we determine its required physical limits on the disk surface and predict their coordinates in B (see Fig. 3). These limits are predicted as the corners of a hypothetical square that lies upon $\mathcal{E}$, encloses the disk, and is aligned with its intrinsic axes, $\boldsymbol{v}$, $\boldsymbol{h}$ which are computed from the estimates of ϕ and θ.

Once the 3D coordinates of these hypothetical points are determined, their coordinates in B are predicted through P. By associating these points with the points in B that define the limits of the displayed content H is calculated. Image M is warped using H and the result is copied into B.

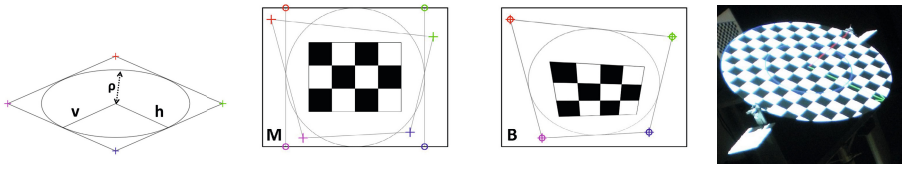

Fig. 3. Display geometry (left to right). (a) The disk enclosed in a hypothetical square. (b) Establishing point correspondences for homography computation: M is shown annotated with (a) the 4 points that correspond to the corners of the hypothetical square and (b) the same corners projected upon the projector buffer. (c) Warping of M according to H makes M appear undistorted upon the disk and aligned with its intrinsic axes. (d) M is set to contain a checkerboard image and visualize axes $\boldsymbol{h}$ (red) and $\boldsymbol{v}$ (green); when warped and projected on the disk checkers appear undistorted and axes accurately aligned with the disk intrinsic axes.

5.2 Touch Detection

This process detects which physical points of the disk that are in contact with the fingertips of the user, as well as, the coordinates of these points in M.

The points of fingertip contact upon the disk are found as follows. Given the current plane equation $\mathcal{E}$, $\boldsymbol{c}$, and ρ, the ellipse upon which the disk projects in D is predicted. The pixels of D within this ellipse are transformed into 3D points. For each of these points, its distance to $\mathcal{E}$ is computed. As in [22], we also use two thresholds to detect contact. The first (d_{max}) indicates if a pixel is closer to the camera than the disk surface. However, only this constraint will include points belonging to the user's arm as well. The second threshold (d_{min}) eliminates points that are overly far from the surface to be considered part of object in contact.

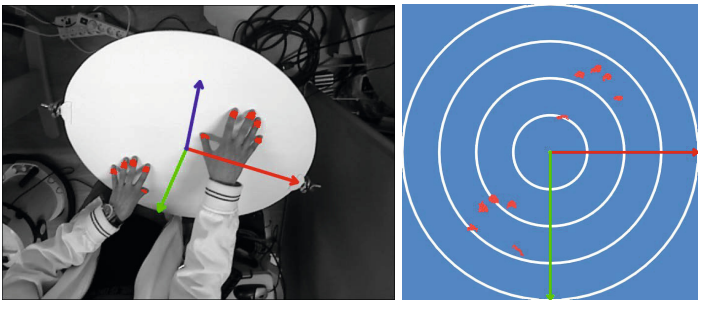

Fig. 4. Touch detection. *Left:* a user touches the surface with all his fingers and the projections of the detected contact points on I are superimposed with red. For reference, estimates of $\boldsymbol{h}$, $\boldsymbol{c}$, and $\boldsymbol{n}$ are also superimposed in red, green, and blue respectively. *Right:* contact blobs in the M's reference frame; column (red) and row (green) axes correspond to $\boldsymbol{h}$ and $\boldsymbol{c}$.

The 3D points found in contact with the surface are then mapped using H^{-1} into an image, let T, that has the same size as M (see Fig. 4). In T, a pixel is 1 if it considered to signify contact at the corresponding point of the surface and 0 otherwise. Typically, a blob of pixels corresponds to the contact of each finger. A blob tracker [23] is employed upon images T to track individual fingers with ids that are consistent across frames.

6 Experiments

The experiments focused in validating the suitability of the proposed system as a multitouch interactive surface. In the first set of experiments, the accuracy of disk pose estimation and projection upon the disk were measured. In the second, the accuracy of contact detection for single and multiple fingertips was measured. Finally, the effectiveness of the system in supporting interactive applications was qualitatively assessed by implementing a pilot application and performing a formative usability evaluation.

The experiments were performed on a conventional PC equipped with *nVidia GeForce GTX 260* 1.2 GHz GPU. The system is adequately fast to keep up with the image acquisition and image projection rates, which occur at 60 Hz. In fact, in offline experiments the system's operation rate is ≈ 120. In our setup, the disk has a radius of $\rho = 30\,cm$ and is placed 150 cm from the ground. A Microsoft Kinect sensor (640 × 480 pixel resolution) and a projector (1280 × 800 pixels) overlook the disk from a height of 149 cm and 102 cm, respectively.

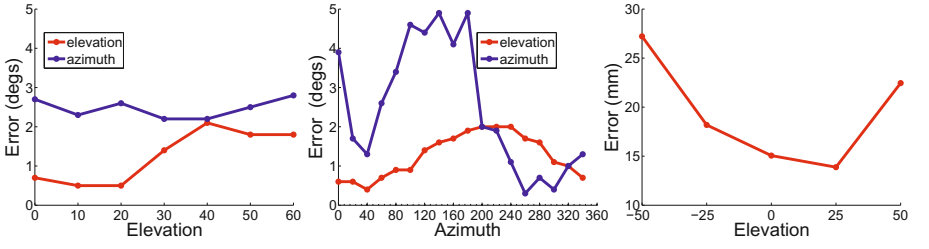

Fig. 5. Accuracy experiments. *Left, middle:* Elevation and azimuth error in degrees per elevation (left) and per azimuth (right). *Right:* Distance error (mm) per elevation.

6.1 Ground Truth Experiments

In this experiment, the accuracy of disk pose estimation was evaluated. The disk was placed at known poses and the error was measured as the difference between the ground truth values and the estimation of the elevation and azimuth angles ϕ and θ, respectively. The disk's elevation range from 0° to 60° was discretized in 7 steps of 10° each. The full azimuth range (360°) was discretized in 18 steps of 20°. Ground truth was measured using a digital inclinometer, temporarily mounted on the disk. No occlusions of the disk occurred in the experiment. The results are presented in Fig. 5(left, middle).

The accuracy of projection, based on the disk's pose estimation was assessed as follows. A checkerboard pattern was displayed in M and correspondingly projected upon the disk. The aspect ratio of checkers was confirmed to be 1 upon the disk, for the above range of elevations as above, meaning that M's appearance on the disk is devoid of projective distortions for that range of elevations. We observed that though the aspect ratio was preserved up to angles as steep as 70°. However, images projected in more oblique angles than $\approx 50°$ were poor in quality and, thus, limited operation of the system up to that obliqueness.

6.2 Interaction Experiments

To assess the system's usability as a touch display two interaction experiments were performed. In the first, the accuracy of touch detection was measured and, in the second, tracking of multiple fingers in simultaneous contact was evaluated. In essence, the evaluation concerned the accuracy of registration between the projected display and contact localization estimates. The two experiments were performed by 5 users each that were naive to the experimental hypotheses.

In the first experiment, users had to touch a sequence of dots, of .75 cm radius, that were appearing upon the disk (see Fig. 6, left). The users were instructed to touch the dots at their centers. By comparing the distance of the touch event in M with the projected location of the dot, the error was measured. The sequence consisted of 70 dots, laid out on 7 concentric circles, centered at the disk's center. The entire surface of the disk was used. The dots were presented to the users one at a time and in random order. The experiment was repeated for 5 elevation angles, in the range of $[-50°, 50°]$, in steps of 25°. Fig. 5(right) presents the average error in millimeters, for each elevation angle.

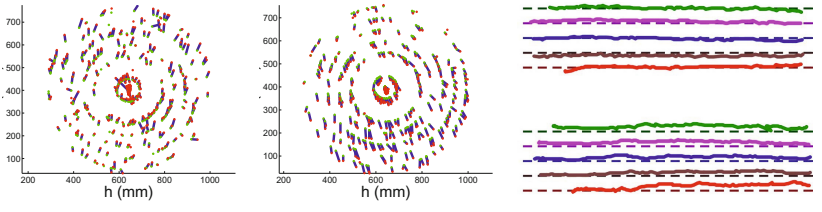

Fig. 6. Interaction experiments. *Left:* Two plots of the projected dots and the estimated contact locations for $\theta = 30°$ and $\theta = -40°$. *Right:* Two plot of the projected lines (straight dashed lines) and traced contact trajectories for the same posture. Both examples are shown in M's reference frame.

In the second experiment, the users were presented with 5 vertical, parallel line segments, which were instructed to trace with their fingertips (see Fig. 6, right). The stripes appeared at a distance of 3 cm from each other. The experiment was repeated for each of the same 5 elevation angles of the previous experiment (see above). The trajectories were recorded and using the stripes as ground truth, the Multiple Object Tracking Accuracy (MOTA) metric [24] was calculated to

evaluate the accuracy of multiple simultaneous contacts to be 1 in the whole range of system operation, that is when $\phi \in [0°, 50°]$.

6.3 Pilot Application

To test the system in a realistic setting, through a set of representative user tasks, an interactive application was developed. The application supports the exploration of ancient artifacts in 360°. The data of the application include datasets where ancient artifacts have been placed on a turntable and photographed from the side, in 360 steps of 1°. For each step, the direction of illumination was mechanically modulated to follow an arc trajectory above the artifact in 20 steps. Using images corresponding to the same set of illuminations, a sparse reconstruction of the artifact was obtained using [25] and regions of interest were defined upon this point, corresponding to "hotspots" on the surface of the artifact.

In the application, a photograph of the artifact is initially projected on the surface of the metal disk, as can be seen from the angle that the metal disk is rotated. By rotating the disk around the vertical axis (θ), the user can see 360 different views of the artifact, as if the actual object was placed behind the disks surface, thus creating a 3D visualization effect. By tilting the disk surface, the user can access alternative lighting settings, revealing different details of the artifact. When the user touches the metal surface, hotspot areas of the current view are presented. Upon touching a hotspot, related information is presented. Additionally, using two fingers, the user can zoom in the image.

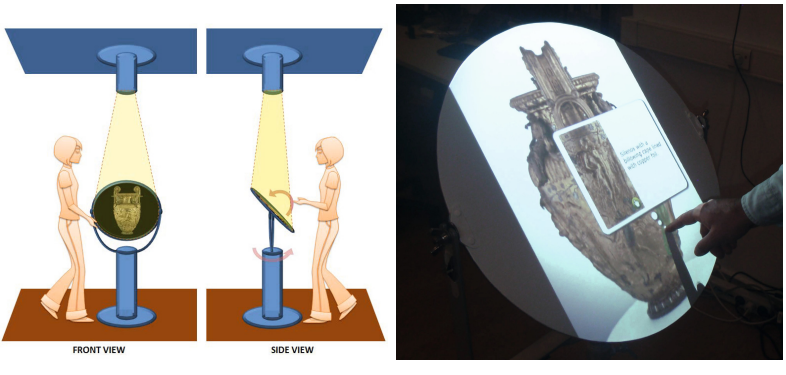

Fig. 7. Pilot application.*Left:* Schematic overview of the pilot application installation *Right:* Working prototype of the pilot application

7 Conclusion and Future Work

A visual approach that creates an interactive multitouch surfaces upon a 2-DOF rotating surface and its implementation have been presented and evaluated.

The results from experiments show that disk pose is estimated very accurately despite the presence of sensor noise and user hand interaction. Correspondingly,

an accurate metaphor of a display upon it is created. Furthermore, user interface input originating from angles ϕ and θ can be reliably used for fine operations. For example, in the pilot application modulation of viewpoint and illumination occurred smoothly and accurately when users moved the disk.

The accuracy of multiple contact detection and localization is sufficiently accurate for conventional multitouch screen interaction. Indeed, this accuracy decreases at oblique angles of the disk, and originates mainly from the reduction in area that the disk undergoes in the sensor's (depth) image. At such angles the projector's limitations are also reached, as less pixels can be used to form an image upon the disk surface. Improvements could use steerable projectors (i.e. [12]) and sensors to compensate for this obliqueness, by rotating in accordance to the elevation (ϕ) of the disk.

A limitation of the approach is met in multitouch and multi-hand interaction and in particular when using more than one hands upon the surface. In such cases, user digits that may be in contact with the disk are occluded from the sensor and missed. Though this topic could be addressed with additional sensors a topic of future work is to investigate whether a more retentive tracking of user hand position would suffice user requirements.

Another topic of future work is a more robust way to disambiguate the azimuth (θ) of the disk when it is approximately parallel to the ground plane. The contour of a disk with less symmetries (i.e. an ovaloid, or an irregular shape) could provide a more global and, thus, more robust orientation cue than the localization of disk joints.

References

1. Rowell, L.: Scratching the surface. NetWorker 10, 26–32 (2006)
2. Rekimoto, J.: Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In: CHI, pp. 113–120 (2002)
3. Streitz, N., Tandler, P., Muller-Tomfelde, C., Konomi, S.: Roomware: Towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds (2001)
4. Wilson, A.: Playanywhere: a compact interactive tabletop projection-vision system. In: UIST, pp. 83–92 (2005)
5. Han, J.: Low-cost multi-touch sensing through frustrated total internal reflection. In: UIST, pp. 115–118 (2005)
6. Gross, T., Fetter, M., Liebsch, S.: The cuetable: cooperative and competitive multitouch interaction on a tabletop. In: CHI, pp. 3465–3470 (2008)
7. Gaver, W., Bowers, J., Boucher, A., Gellerson, H., Pennington, S., Schmidt, A., Steed, A., Villars, N., Walker, B.: The drift table: designing for ludic engagement. In: CHI, pp. 885–900 (2004)
8. Microsoft (Microsoft surface), `http://www.surface.com`
9. Dietz, P., Leigh, D.: Diamondtouch: a multi-user touch technology. In: UIST, pp. 219–226 (2001)
10. SMART: Smart table (2008), `http://www.smarttech.com/`
11. Pinhanez, C.: Using a steerable projector and a camera to transform surfaces into interactive displays. In: CHI, pp. 369–370 (2001)

12. Kjeldsen, R., Pinhanez, C., Pingali, G., Hartman, J., Levas, T., Podlaseck, M.: Interacting with steerable projected displays. In: FG (2002)
13. Jones, B., Sodhi, R., Campbell, R., Garnett, G., Bailey, B.: Build your world and play in it: Interacting with surface particles on complex objects. In: ISMAR, pp. 165–174 (2010)
14. Harrison, C., Benko, H., Wilson, A.: Omnitouch: wearable multitouch interaction everywhere. In: UIST, pp. 441–450 (2011)
15. Song, P., Winkler, S., Tedjokusumo, J.: A tangible game interface using projector-camera systems. In: HCI, pp. 956–965 (2007)
16. Grammenos, D., Michel, D., Zabulis, X., Argyros, A.: Paperview: augmenting physical surfaces with location-aware digital information. In: TEI, pp. 57–60 (2011)
17. Reitmayr, G., Eade, E., Drummond, T.: Localisation and interaction for augmented maps. In: ISMAR, pp. 120–129 (2005)
18. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395 (1981)
19. Zhang, C., Zhang, Z.: Calibration between depth and color sensors for commodity depth cameras. In: ICME, pp. 1–6 (2011)
20. Vezhnevets, V., Velizhev, A., Chetverikov, N., Yakubenko, A.: GML C++ camera calibration toolbox (2011), `http://graphics.cs.msu.ru/en/science/research/calibration/cpp`
21. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
22. Wilson, A.: Using a depth camera as a touch sensor. In: ACM Int. Conf. on Interactive Tabletops and Surfaces, pp. 69–72 (2010)
23. Argyros, A.A., Lourakis, M.I.A.: Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 368–379. Springer, Heidelberg (2004)
24. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. Journal of Image and Video Processing, 1–10 (2008)
25. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH, pp. 835–846 (2006)