2013 Second International Conference on Advanced Computing, Networking and Security

Projection and Interaction with Ad-hoc Interfaces on Non-planar Surfaces

Dere Kundan S.
Department of Information Technology
National Institute of Technology Karnataka,Surathkal.
Managalore, India
Email: kundansdere@gmail.com

G. Ram Mohana Reddy
Department of Information Technology
National Institute of Technology Karnataka,Surathkal.
Managalore, India
Email: profgrmreddy@gmail.com

***Abstract*—Projector-based display systems have been used in area of computer interaction as Ad-hoc interface in recent time. The mobile hand-held projectors are becoming more popular. Many human centric user interfaces with the human wearable computer are being developed. Most of such system uses daily objects for projection and the interaction. But most of ignores the fact that these object surfaces are not planar. Hence such interfaces suffers from the distortion due to non-planar projection surface. Besides this projection quality also suffers from the radiometric distortion as well. Further more the interaction proposed with such interfaces bound to the planar surface only. Hence this paper is targeted to address the geometric distortion free projection of and interaction with such interfaces on non planar surfaces. Kinect is used as depth sensor for 3D scenario acquisition. We use image-per warping to mesh from Kinect. We use colored fingertip gloves for interaction. Here our system aims any day to day object surface for distortion free projection such as human body, curved wall, room corners, curtain's and many more objects.**

***Keywords*-Projection on Non-planar surface, Interaction with Non-planar surfaces, Kinect, Adaptive projection, Projection Based Augmented reality, Real world object Interaction, Wearable Interface, Object Augmentation.**

I. Introduction

The recent advent of computer vision and display technologies has encouraged the development of Ad hoc projection based interfaces and interaction with same. Such systems have changed the way of interaction. Popularity of mobile projectors and Ad hoc interfaces has taken the mobile computing to new stage. A lot of research work is carried out in area of such digital interfaces. In such systems, projector is output device while for interaction RGB camera is used as the feed back device, to adapt the projection and to have the interaction. As the projection is targeted toward the hand-held objects the projector and the camera need to synchronized perfectly, to follow the object movement. This procedure is known as the projection mapping. In earlier day where depth maps were not at their best, researcher used the "Structured Light Patterns" for acquisition of surface geometry of scenario. These Light Patterns usually gets embedded with original imagery. Synchronized cameras captures the distortion of these patterns and produce the triangular mesh of surface. As the pattern images got embedded to the original imagery it reduces the quality of projection. Here we uses the depth cameras for acquisition of 3D information of surfaces, discarding the need of embedding patterns in imagery. Hence the quality of projection is preserved. Further in old procedure it takes some computation time to generate the 3D mesh from the pattern, as this is heavy image processing. While in case of depth camera the 3D description of the scenario is readily available for use avoiding the heavy image processing, giving the better response for interaction. Depth sensor is time-of-flight camera features a near-infrared pulse illumination component.

II. Related Work

A lot of effort's are taken in the area of distortion free projection on non planar surfaces. Different methodologies are used in order to acquire the 3D mesh of surface to be project on. As well as so many interaction methodologies are proposed for user interaction with projector based ad hoc interfaces.

A. Acquisition 3D Description of Surface

1) Structured Light Pattern Approach: System proposed by Sugimoto M, et. al. [1] uses the Pro-Cam system consisting of the a hand-held projector and a RGB camera. It uses an Impressible Structured Light Pattern of black white check bored pattern which gets embedded in projection image making slight but unrecognizable changes in image. With synchronized Projector and camera, system able capture 3D geometry in single shot Image. With the help of recognized pattern the Feature points and Feature lines are drawn making the complete 3D mesh. System proposed by Yang, R. and Welch, et al. [2] present an iterative approach to automatically determine the display surface geometry, without human intervention, unobtrusively and continuously while the system is being used for real work. System use Extended Kalman filter to get the 3D description of projection surface. But this system also uses same methodology of structured light, that reduces the image quality. In relatively same area work by Sukthankar et. al. [3], Won et. al. in [4], Ashdown et. al. in [5], [6], [7] proposes different methods for geometric compensation in

978-0-7695-5127-2/13 $31.00 © 2013 IEEE
DOI 10.1109/ADCONS.2013.41

direct projected reality. [8] proposes the geometric as well as the radiometric compensation based on the pro-cam system. [9] propose the projected augmentation.

2) Depth Sensor Approach: System proposed by Chris Harrison, et al. [10] is a wearable depth sensing camera and projector system that enables interactive application's on everyday surfaces. System allows the wearer to use their hands, arms and wall as the graphical interactive surfaces. System provides the interaction with such surfaces with multi-touch capability without any calibration. It uses depth map in order to classify the touch event.

B. User Interaction

Projector based interactive system's like Sixth Sense, Interactive Dirt makes use of color finger gloves of IR reflective as finger tip markers for figure tracking. But Chris Harrison, et al. [10] proposed the interactive environment without any extra element like any marker or sensor added. System by Chris Harrison, et al. [10] proposes a unique approach to ad hoc finger tracking.

System proposed by Pranav Mistry et al. [11] makes use of head mounted Camera-projector collaboration, giving user facility to take his mobile computational power outside and interact with. System uses the small colored fingertip gloves in order to detect the finger tip, as the color segmentation give easy way for same.

Jochen Huber et. al. [12] uses Everyday Objects as Tangible Controls. It does not map one particular object to a certain digital functionality. [12] advocate mapping the unique affordability of everyday objects such as rotating to unique digital functions. In system Jochen Huber et al. [12], projection surfaces, currently considered flat surfaces of 3D objects. System model them as 2D planes in 3D space.

System proposed in [13] uses hardware trigger for interaction while system propoed in [14] uses invisible IR markers for feed back.

III. System Overview

Here is system with interaction and adaptive projection of ad hoc interface on non-planar surface. System is devided in two modules one is Adaptive projection by Geometric compensation and User-Interaction.

A. Geometric Compensation

We use image pre-warping for the geometric correction of image. Basically the the triangular mesh of the projection surface is created. warping image to this mesh will imitate geometrical distortion on physical surface, while the wrapped image on the mirror in $Z-Axis$ of such mesh and projection on physical surface nullifies the distortion effect. Geometric compensation module follows some steps which are explained below.

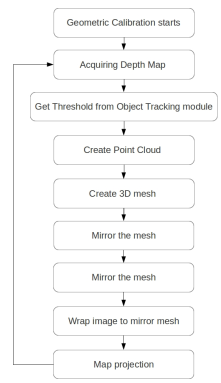


Figure 1. Geometric Compensation module Flowchart

1) Object Tracking: As the in scenario of hand-held projector is considered, the selection object to project on from scenario is important. We consider the scenario where user is wearing such system as pendant or as shoulder mounted device. For object selection to project on, depth map is used. In depth graph sudden drop is observed in non-decreasing ordered depth values, as the difference in physical distance of object from background as shown in Fig. 2. This value is considered as the threshold for background subtraction. The red rectangle point outs the sudden drop in the depth values. Algorithm 1 describes all steps for threshold selection.

Algorithm 1 Background Subtraction Threshold Selection

Require: Depth Map

$depth\_Map[\,] = sort\ (\ depth\_Map[\,]\ )$

$depth\_Map[\,] = reverse\ (\ depth\_Map[\,]\ )$ {Setting depth values in non-increasing order}

for $i = 0 \rightarrow depth\_Map\_Length$ **do**

  if $absolute\ (depth\_Map[i] - depth\_Map[i+1]) < 100)$ **then**

    $threshold\_Value = depth\_Map[i]$;face {100 as minimum depth separation threshold}

  end if

end for

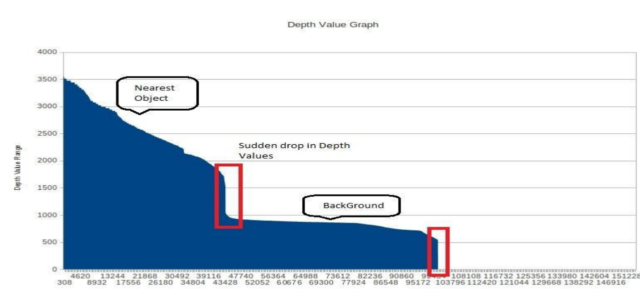


Figure 2. Graph for Depth values of scene in Non-increasing order

2) Point Cloud Creation: Once is the object to be projected on is located next task is to create the point cloud. Original point cloud will be so much dense, considering all points for 3D mesh creation, will increase the computation time for image warping in subsequent stage. Hence we can skip the some points for mesh creation, creating low density mesh.

Algorithm 2 Point cloud Creation

Require: Depth Map and threshold_Value
- **for** $i = 0 \rightarrow depth\_Map\_Length$ **do**
  - **if** $depth\_Map[i] < threshold\_Value$ **then**
    - Discard the pixel
  - **end if**
- **end for**
- **for** $i = 0 \rightarrow depth\_Image\_Width$ **do**
  - **for** $j = 0 \rightarrow depth\_Image\_Height$ **do**
    - $Point = real\_World\_Map[i, j]$
    - {Here each depth value gets transferred to 3 dimension co-ordinates system}
    - Draw $Point$ if it is $step$ ahead of earlier point.
    - {Increase / Decrease $step$ to change point density.}
  - **end for**
- **end for**

Here $step$ is user defined predetermined value to control the mesh density and eventually control the quality of image warping and computation speed.

3) 3D Mesh Generation: This module draws the triangular mesh from point cloud. Four neighboring points are considered for drawing two triangles.

Algorithm 3 3D Mesh Generation

Require: Depth Map
- **for** $i = 0 \rightarrow depth\_Image\_Width$ **do**
  - **for** $j = 0 \rightarrow depth\_Image\_Height$ **do**
    - $Point1 = real\_World\_Map[i, j]$
    - $Point2 = real\_World\_Map[i, j + step]$
    - $Point3 = real\_World\_Map[i + step, j]$
    - $Point4 = real\_World\_Map[i + step, j + step]$
    - Draw $Triangle\ as\ Point1, Point2, Point3$
    - Draw $Triangle\ as\ Point2, Point3, Point4$
    - {Increase / Decrease $step$ to change point density.}
  - **end for**
- **end for**

4) Mesh Mirroring: We use the methodology of image pre-warping technique for geometric correction. For compensation the mesh should be mirrored in z-Axis. Algorithm shows step wise description of mesh mirroring module.

Algorithm 4 shows mesh reflecting technique. In end it is just change in $Zaxis$ value. Plotting newly generated transferred points generates the mirror for the input mesh.

Algorithm 4 Mesh Mirroring

Require: Depth Map $min\_Z = 0$
- **for** $i = 0 \rightarrow depth\_Image\_Width$ **do**
  - **for** $j = 0 \rightarrow depth\_Image\_Height$ **do**
    - $Point = real\_World\_Map[i, j]$
    - **if** $min\_Z < Point.Z$ **then**
      - $min\_Z = Point.Z$
    - **end if**{To get Minimum values of Z for Mirroring}
  - **end for**
- **end for**
- **for** $i = 0 \rightarrow depth\_Image\_Width$ **do**
  - **for** $j = 0 \rightarrow depth\_Image\_Height$ **do**
    - $Point1 = real\_World\_Map[i, j]$
    - $Point2 = real\_World\_Map[i, j + step]$
    - $Point3 = real\_World\_Map[i + step, j]$
    - $Point4 = real\_World\_Map[i + step, j + step]$
    - $Point1.Z = 2 * min\_Z - real\_World\_Map[i, j].Z$
    - $Point2.Z = 2 * min\_Z - real\_World\_Map[i, j + step].Z$
    - $Point3.Z = 2 * min\_Z - real\_World\_Map[i + step, j].Z$
    - $Point4.Z = 2 * min\_Z - real\_World\_Map[i + step, j + step].Z$ {Here each depth value gets transferred to 3 dimension mirror co-ordinates system}
    - Draw $Triangle\ as\ Point1, Point2, Point3$
    - Draw $Triangle\ as\ Point2, Point3, Point4$
    - {Increase / Decrease $step$ to change point density.}
  - **end for**
- **end for**

This module result in compensated image, to be projected on non-planar surface, to be appear as un-distorted. In the end module use the homography concept to wrap image to triangle.

5) Projection Mapping: The co-ordinate system differs in three different domains like real world, Camera and Projector. In order to map these co-ordinate system the intrinsic and extrinsic parameter acquisition is necessary.
In [15] methods of camera and projector intrinsic as well as extrinsic parameter calculation is explained. Method uses method proposed by Zhang et. al.[16] chessboard pattern for calculation of parameters.

B. User Interaction

In order to facilitate user with interaction on such interface, we use depth map. This module is designed to give user real touch feel for interaction, rather than asking for some inconvenient gesture in air. Following two modules explains step wise interaction module and working, While in subsequent module we designed and developed Paint and Movie player application to test the usability of the interaction module. This module is designed on the model proposed by [10] , we uses the fingertip markers for accurate tip position detection.

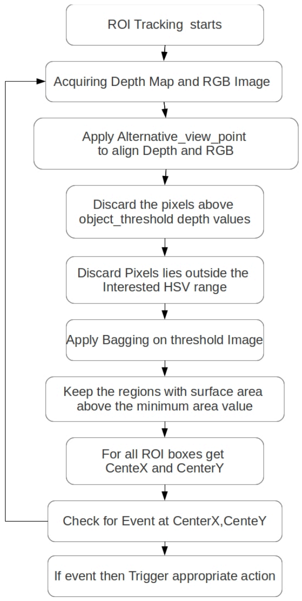


Figure 3. Interaction Module Flowchart

Figure 3 explains step wise module implementation for Interaction module. Pixels are filtered on two layers, first for depth values with object_Threshold obtained in object_tracking module and in second stage with HSV color range to track the color markers in RGB image. Once markers at fingertips are located, the algorithm checks for the event in all such positions.

1) Fingertip Detection: As the touch will be at fingertip, recognizing the fingertip accurately is very important. In [10] derivatives of the depth image and proximity of finger slice is used to locate the fingertip. But due to proximity technique becomes so constrained. In this module we combine the fingertip marker and depth map technique for best results in 3D object interaction. We use the red color marker gloves on fingertip for accurately detecting the fingertips. Figure 3 explains the steps involved in fingertip detection module.

2) Click Event Classification: Once the fingertip is detected, flood fill algorithm is applied on depth image with tolerance of 10mm. As the flood fill area will suddenly increases on physical touch, this event is considered as the touch event, and appropriate function will get triggered as used in [10].

IV. Results and Discussion

Prototype system is developed on i7 computer 8GB of RAM and with Microsoft Kinect sensor as depth camera. Implementation is carried out in the Processing [17] and contributed libraries for processing like Simple-OpenNi, Bob-Scanner. As in interaction module uses colored fingertip

Algorithm 5 Click Event

Require: Depth Map ,click_X, click_Y face

flood_Count = $food\_Fill(click\_X, click\_Y, depth\_Map[])$

if $flood\_Count > threshold * hangOver\_Flood\_Count$ **then**

  Trigger Click Event

end if

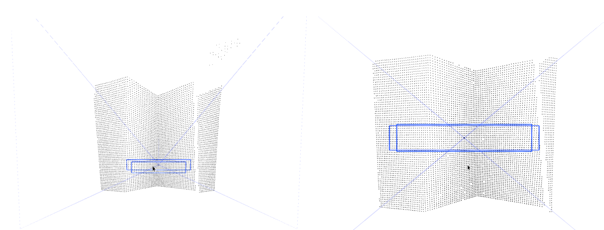
markers for locating fingertip, light conditions are very much important. As the projector itself works as light source, selecting the appropriate UI color scheme is very much important. As fingertip glove may itself remain undetected due to radiometric distortion.

- **IDE** : Processing-(Java)
- **Kinect driver**: OpenNI Drivers - Simple-openNi (Processing wrapper)
- **Image Processing**: OpenCv, JavaCv, BobScanner etc.
- **Calibration**: RGBDemo Calibration.[16].

A. Geometric Compensation

Figure 4. Nearest Object Detection and background Subtraction

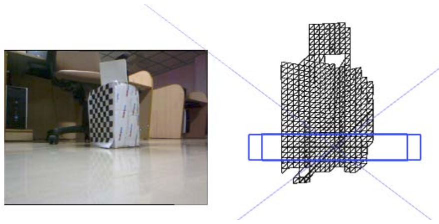
1) Object Tracking: Figure 4 is snapshot is taken for scenario of the nearest object selection module output. In scenario the cardboard box is observed as the nearest object.



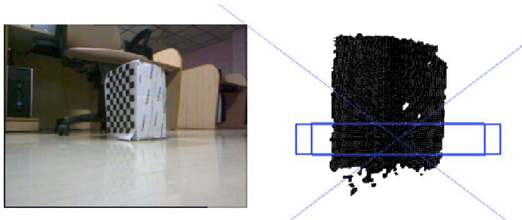
(a) (b)

Figure 5. Point Cloud Creation

2) Point Cloud Creation: Figure 5 is snapshot is taken for output of point Cloud Creation module where wall surface depth map is input to the algorithm. Blue lines is Kinect camera Frustum rectangles. Density of points is predefined one. Consideration of more points in point cloud will increase the image warping quality, increasing computation and eventually reducing the efficiency.



(a)



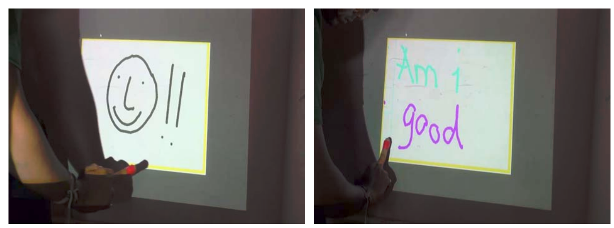
(b)

Figure 6. 3D Mesh of Tracked Object

(a) (b)

(c) (d)

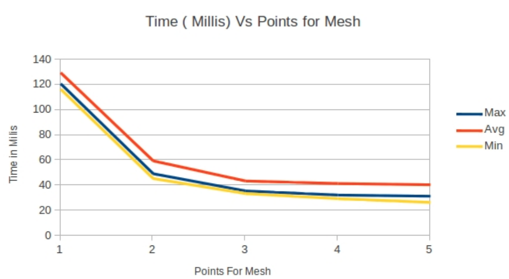
Figure 8. Geometric Compansation



(a) (b)

Figure 9. Paint Application Snapshot

Algorithm 3 describes detailed method followed. Figure 7 shows the performance analysis of drawing mesh with difference point densities.



(a)

Figure 7. 3D mesh creation performance

Figure 8[a] and [c] shows distorted images due to non-planar surface, while [b] and [d] are compensated images. We used OpenGL with Processing platform support for rendering the warping and rendering the image to mesh.

B. User Interaction

In next stage mirrored mesh is textured with user defined image, with process of homography. All modules are sufficiently works faster to be used in real time environment. For performance analysis perpose we developed two applications A movie player and simple paint.Figure 9 and 10 are the snapshots of the UI developed for such projected interface. Figure 11 shows the test output of click performance. As we use colored finger markers, we use centroid of detected area as anchor point. But due to light conditions color detection varies. Two users took click test of clicking the 4X4 grid button's. Figure 11 shows the clicks by two users.

V. Conclusion

System developed works fine as per as constraints are considered. As the problem observed in fingertip detection we uses the color markers for accurately detecting the fingertips. V In Future user should be provided with glove free interaction as developed by [10].

As the Geometric compensation module work fine in real time for portable projectors, same module can be used with traditional projectors as well. Use of proposed module with traditional projector will increase the projector usability by eliminating the projector position and surface geometry constraints. With this modules traditional projectors will be able to project more efficiently from any angle of projection from anywhere. This will make efficient projection non planar surface as well.

References

[1] M. Sugimoto, V. N. Dao, and E. Seki, "An adaptive projection technique for monochromatic non-flat surfaces using a hand-held projector–camera system," in *MP 2 Workshop CHI'11*.

[2] R. Yang and G. Welch, "Automatic and continuous projector display surface calibration using every-day imagery," in *Proceedings of 9th International Conf. in Central Europe*

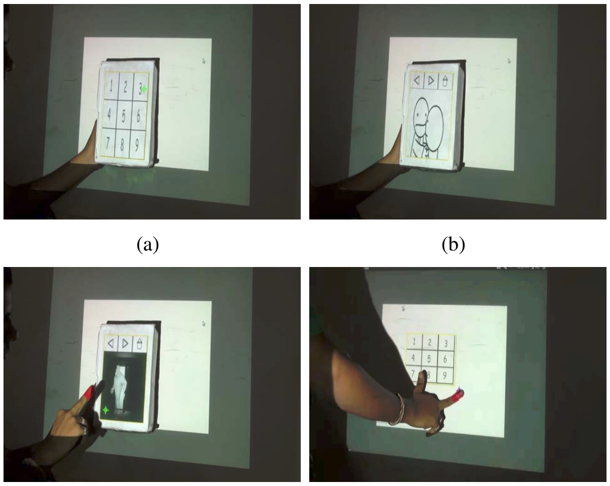


Figure 10. Movie Player Snapshots

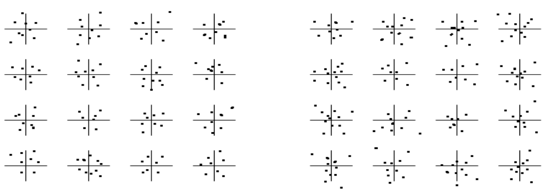


Figure 11. Interaction Analysis

in Computer Graphics, Visualization, and Computer Vision WSCG, 2001.

[3] R. Sukthankar, R. G. Stockton, and M. D. Mullin, "Smarter presentations: Exploiting homography in camera-projector systems," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 247–253.

[4] J. Won and J. Cho, "Geometric image compensation for a portable projector based on prewarping and optimal projection area generation," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*. IEEE, 2011, pp. 873–874.

[5] M. Ashdown, M. Flagg, R. Sukthankar, and J. M. Rehg, "A flexible projector-camera system for multi-planar displays," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–165.

[6] K. Fujii, M. D. Grossberg, and S. K. Nayar, "A projector-camera system with real-time photometric adaptation for dynamic environments," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 814–821.

[7] O. Bimber, R. Raskar, and M. Inami, *Spatial augmented reality*. AK Peters Wellesley, 2005.

[8] H. Park, M.-H. Lee, S.-J. Kim, and J.-I. Park, "Surface-independent direct-projected augmented reality," in *Computer Vision–ACCV 2006*. Springer, 2006, pp. 892–901.

[9] H. Benko, R. Jota, and A. Wilson, "Miragetable: freehand interaction on a projected augmented reality tabletop," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 199–208.

[10] C. Harrison, H. Benko, and A. D. Wilson, "Omnitouch: wearable multitouch interaction everywhere," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 441–450.

[11] P. Mistry, P. Maes, and L. Chang, "Wuw-wear ur world: a wearable gestural interface," in *CHI'09 extended abstracts on Human factors in computing systems*. ACM, 2009, pp. 4111–4116.

[12] J. Huber, J. Steimle, C. Liao, Q. Liu, and M. Mühlhäuser, "Lightbeam: nomadic pico projector interaction with real world objects," in *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*. ACM, 2012, pp. 2513–2518.

[13] X. Cao, C. Forlines, and R. Balakrishnan, "Multi-user interaction using handheld projectors," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007, pp. 43–52.

[14] K. D. Willis, I. Poupyrev, S. E. Hudson, and M. Mahler, "Sidebyside: Ad-hoc multi-user interaction with handheld projectors," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 431–440.

[15] "Rgbdemo,kinect calibration," http://labs.manctl.com/rgbdemo/, accessed: 2013-04-20.

[16] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[17] C. Reas and B. Fry, *Getting Started with Processing*. Make, 2010.