Active Location Tracking for Projected Reality Using Wiimotes

Siam Charoenseang and Nemin Suksen

Institute of Field Robotics, King Mongkut's University of Technology Thonburi,
126 Pracha-u-thit Bangmod Tungkru 10140 Bangkok, Thailand
siam@fibo.kmutt.ac.th, naymin_suksen@msn.com

Abstract. Some addressed issues in projected reality are location acquisition, limited work space, and geometric distortion. This paper proposes a low-cost, robust, fast, and simple method for handling addressed problems using infrared camera in Nintendo's Wiimotes and a pan-tilt camera head. Two Wiimotes are attached on both horizontal and vertical axes of a portable projector mounted on a pan-tilt camera head. Hence, it can detect 4 infrared LEDs on the corners of a display surface in perspective projection volume. The augmented images are wrapped to fit the display area. To increase the system workspace, a pan-tilt camera head is used to track the display surface. While the display surface or the projector moves, a proposed fast location tracking algorithm between two Wiimotes is implemented. Experimental results demonstrate the ability of real time location tracking at 97 fps that is more than the refresh rate of typical projector. Finally, the active location tracking using the pan-tilt camera head can give workspace more than 36 times of the normal perspective projection workspace.

Keywords: Perspective Location Tracking, Projected Reality, Augmented Reality.

1 Introduction

In modern augmented reality application, projected reality approaches are not limit to represent augmentations information. It can enhance human vision sense by changing the texture on surface such as projection surfs on the fall, projection of moving wheel on a static plate. Existing research works in the field of computer vision, human computer interface, augmented reality communities introduce tracking methods for finding projected location such as rotating mirror with video camera [1], attach sensor with magnetic or optical tracker [2], infrared-reflector markers with camera [3], passive marker with rotatable camera [4], 3D surface modeling with a ray/triangle intersection algorithm [5], and structure light patterns with optical fibers [6, 7]. While far from a complete list, those methods manifest the value of flexible image projection to any surface rather than traditional use of projector on passive flat screen.

In this paper, the potential of solution of using the perspective projection of input/output geometry relationship between camera and projector is explored as shown in Figure 1. In an ideal case, camera and projector have the same horizontal

R. Shumaker (Ed.): Virtual and Mixed Reality, Part II, HCII 2011, LNCS 6774, pp. 309–317, 2011.

and vertical field-of-view (FOV), and alignment. This yields equality between the location of targeted object on camera's plane and projection's plane. However, it is impossible to place camera and projector at the same pose. Hence, this paper proposes a method to make virtual camera in ideal case by using two Nintendo's Wiimotes placing closely to the projector on horizontal and vertical axes.

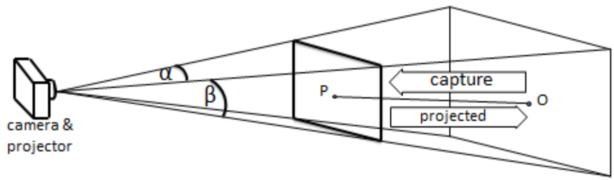

Fig. 1. Ideal case, camera and projector are placed at the same pose. α and β are field of view horizontal and vertical angles, respectively. P is a coordinate of (x,y) on camera and projector planes. O is a coordinate of (x,y,z) on the world space.

2 System Overview

The purposed location tracking system includes a portable projector, four diffusion infrared LEDs, two Nintendo's Wiimotes, a pan-tilt camera head, and a computer for rendering wrapped images to the projector as shown in Figure 2. The configuration of projector and two Wiimotes are placed on the pan-tilt head that looks at infrared LEDs on the display surface. This configuration is designed to prove our algorithm which tracks four LEDs at the same time. To project an image onto a display surface, calibration between camera and projector for creating the virtual camera using image warping [8] needs to be done first. This virtual camera is created to match corresponding points between two Wiimotes' coordinates and convert those points into a projector's coordinate. Wiiuse library[9] is implemented for querying Wiimote's data in this prototype. Secondly, the targeted surface must be inside within the projector's frustum. To do that, it is very common to choose four corners of a display surface to be tracked. Figure 3 shows the locations of all four infrared LEDs in the display surface. The virtual camera then gives all corners' positions in 2D space to the system. OpenGL library is used to render textured graphics to fit the whole display surface.

Fig. 2. System overview of the proposed projected reality

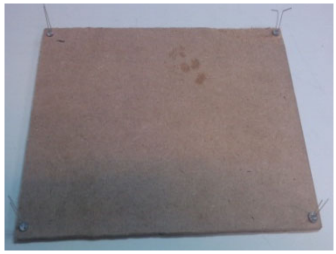

Fig. 3. Four diffusion infrared LEDs embedded on the corners of surface display

2.1 Nintendo's Wiimote

The Wiimote is one of the most common hacked game controllers developed by Nintendo Corporation. It includes many sensors such as accelerometer, infrared camera, and buttons. In this system, only the camera and Bluetooth connectivity are used.

The Wiimote can track up to the most brightest four infrared points at 100 Hz at 1024 × 768 pixels using multiobject tracking (MOT) engine [10]. The camera sensor has 45 degree horizontal FOV. In comparison with webcam at the same price, it gives more of 2.56 times resolution and 3.3 times refresh rate. Wiiuse library is used to retrieve camera data from the Wiimote via Bluetooth communication [9].

2.2 Pan-Tilt Head

The pan-tilt head is used for increasing workspace of perspective projection volume. In comparison with projector's FOV, it gives more of nine and four times on horizontal and vertical FOV, respectively. It consists of two digital servo networked motors (Robotis Dynamixel AX12+) which are connected to the computer via TTL to USB interface. Pan-tilt head is assembled in a simple direct drive form. To track the object, the origin of camera image on left top corner is changed to the center of image to find the offset which is determined as error on x and y axes. Centroid of four circles is also determined as delegated tracking point. The error in pixels is converted to an angular offset shown in [4] as following equations:

$$\Delta Pan = \tan^{-1}\left(\left(X_{pos} - \frac{X_{res}}{2}\right) \times \frac{\tan\frac{FOV_{hor}}{2}}{\frac{X_{res}}{2}}\right) \tag{1}$$

$$\Delta Tilt = \tan^{-1}\left(\left(Y_{pos} - \frac{Y_{res}}{2}\right) \times \frac{\tan\frac{FOV_{hor}}{2}}{\frac{Y_{res}}{2}}\right) \tag{2}$$

Where (X_{pos}, Y_{pos}) represents the centroid, X_{res} and Y_{res} are the size of image resolution. The angle for motors is obtained from the sum of the resultant values ΔPan and $\Delta Tilt$ and each present motor position. The absolute pan and tilt angles are sent to the motor controller when angular offset is more than a threshold to reduce unnecessary update. The speed of motor is varied between 2 rpm and 30 rpm accordingly to the motor's angular offset.

2.3 Virtual Camera

The simple model of location tracking system is shown in Figure 1. The real world object's position is at point O. It appears at point P on the camera plane. If the projector generates graphics at point P, the graphics is also displayed at point O. One of research's goals is to make a virtual camera with three constraints: i) this virtual camera is at the same as a projector's location and alignment, ii) horizontal and vertical FOVs are the same as projector's FOVs, iii) image plane has the same x and y axes as projector plane's axes. It can done by measuring object's position on x axis by using image obtained from camera which is translated on the vertical direction as shown in Figure 4. The previous process is also done for object's position on y axis in the same way. Two corresponding obtained points from two cameras are matched and used to determine a point in the virtual camera plane.

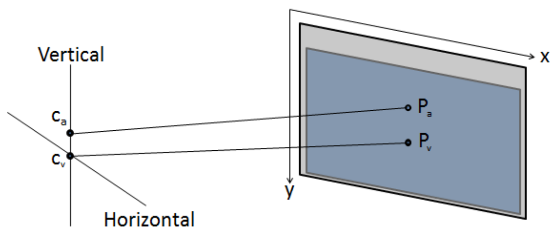

Fig. 4. C_v is position of the virtual camera. C_a is position of the actual camera. P_v and P_a are projections of the same object in the virtual and actual camera planes, respectively.

The first camera is placed closely to the projector with a little bit translation on the vertical axis as shown in Figure 4. The image from the vertical translated camera is warped and clipped to fit the projector's plane. Therefore, the projection points of the same object from the different camera planes, P_a and P_v, have the same coordinate in x axis for any depth. This process is also done for the second camera that is translated on horizontal axis. This gives coordinates in y axis on the virtual camera plane.

In the complete system with two Wiimotes as shown in Figure 5, P_a and P_b are points on the vertical camera and horizontal camera, respectively. (x_a, y_a) and (x_b, y_b) are coordinates of P_a and P_b, respectively. Equation 3 is used to calculate P_v which is the corresponding coordinate of the virtual camera plane.

$$P_v = (x_a, y_b) \tag{3}$$

Fig. 5. Two Wiimotes attached closely on both horizontal and vertical axes of portable projector

2.4 Image Matching

In [11], experimental results of fundamental matrix matching and square matching methods were compared. They took two sets of four points to determine the correspondence between the two images into the matched pair. The square matching method is simpler and has less complexity than the fundamental matrix matching which calculates twenty-four possible pairing for four points. Square matching method is also implemented in this paper. However, this algorithm is broken down when the points are rotated to the vicinity of the quadrant boundaries or rhombus quadrilateral. For carry this problem, all points need to be rotated a little bit around the centroid. The improvement of square matching method is implemented in this research to maintain its simplicity and can be adapted to any quadrilateral. First, the centroid, $C = (C_x, C_y)$, is calculated from a set of four points. If points are rhombus quadrilateral, they are rotated around the centroid by 5 degrees. Secondly, point id, $N = \{0,1,2,3\}$, is assigned for every point $P = (P_x, P_y)$ as shown in Equation 2.

$$N = u + v \tag{4}$$

$u = \{0,1\}$ is used to divide points into left or right sides of the centroid by a vertical line. If P_x is less than C_X, u is 0 otherwise u is 1. In the same way, $v = \{0,2\}$ is used to separate points into below and above sides of the centroid by a horizontal line. if P_y is less than C_y, v is 0 otherwise v is 2. The perpendicular of horizontal and vertical lines at the centroid divides each of the four IR points into unique quadrants by distinct point id. The same id of each image is corresponding point between images from two cameras. This algorithm has a limit to give up to accurate 4 points. As the matching points increase, fundamental matrix matching can be implemented to work with any number of points.

2.5 Coordinate Calibration between Two Wiimotes and a Projector

The virtual camera is the result of collection and calibration of two quadrilaterals from Wiimotes onto targeted quadrilateral of projector using projective mapping method[8]. It maps two Wiimote cameras' coordinate onto a projector's coordinate as shown in Figure 6. For example, if infrared LEDs appear in the projection frustum, the infrared LED positions are used to compute corresponding projector pixels projected on the positions of infrared LEDs. This method requires four triplets of corresponding points that are collected by four-point calibration process as shown in

Figure 6. It is a typical calibration process for any touch-screen system. First, four known locations are projected as the crosshairs on four corners of projector's plane. Secondly, an infrared LED is pointed at each of these crosshair's locations to obtain triplets of corresponding points. Finally, two warping matrices are computed by solving Gaussian elimination. In projective geometry, a 2D real point (x, y) is represented by the homogeneous vector $p = (x', y', w)$ where $(x, y) = (x'/w, y'/w)$ for $w \neq 0$. Given horizontal and vertical warping matrices, M_h and M_v, a vertical camera point $P_v = (s, t) = (s', t', 1)$, and a horizontal camera point $P_h = (u, v) = (u', v', 1)$. Hence, the projector point, P_0, can be obtained using Equations 3-5:

$$(x_v', y_v', w_v) = (s', t', 1)M_v \tag{5}$$

$$(x_h', y_h', w_h) = (u', v', 1)M_h \tag{6}$$

$$P_0 = (x_v, y_h) = (x_v, /w_v, y_h/w_v) \tag{7}$$

For a projector's coordinate in Equation 5, x and y values are obtained from the vertical and horizontal cameras, respectively.

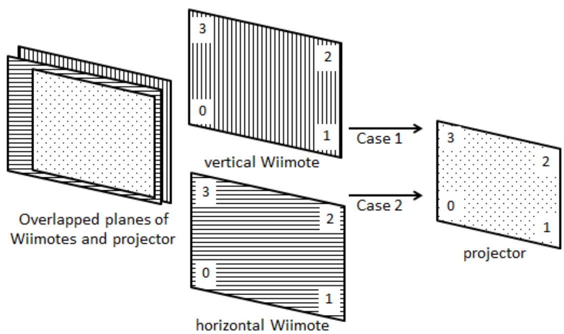

Fig. 6. Overlapped planes from horizontal Wiimote, vertical Wiimote, and projector plane

2.6 User Interface

The user interface (UI) is designed for proofing the concept of the virtual camera. The UI is implemented using OpenGL library. It has two main windows which are a full screen graphics window for projector and a command line window for receiving user's requests. Before using the system, user presses the "C" key to calibrate Wiimotes and a projector using four-point calibration method. User then chooses augmented data such as static image and video as shown in Figure 7. When the display surface is activated in the workspace, the full screen graphics image will be displayed to provide augmented data selected by user's commands. If the object is moved far away from the center of image, the pan-tilt head will move to track that object.

Fig. 7. From left to right: video image projected on moving display. Address information augmented onto a post box.

3 Experimental Results

Active location tracking system is implemented using a portable projector (3M MP150), two infrared cameras (Wiimotes), and a pan-tilt head (Robotics Dynamixel AX12+). The images are at a resolution of 640 by 480 pixels. Four 940 nm diffusion infrared LEDs are attached to four corners of the display surface. All system components are interfaced with a Microsoft's Windows 7 laptop with an Intel Core i5 M460 at 2.53GHz. The location tracking method is evaluated in three areas: i) Geometric Proof ii) Projection Accuracy iii) Processing throughput.

3.1 Geometric Proof

Figure 8 is drawn using Thai Geometer's Sketchpad. It represents boundary of projection volume, object in real world space, object position in image plane separately for each axis. If camera is translated on the vertical axis, only image on y axis has an error as shown in the right side of Figure 8. In this condition, images between projector and camera are parallax. The intersection region can use camera for measuring value in x axis as shown in the left side of Figure 8. In the other word, if camera is translated on the horizontal axis, it can measure the location in y axis.

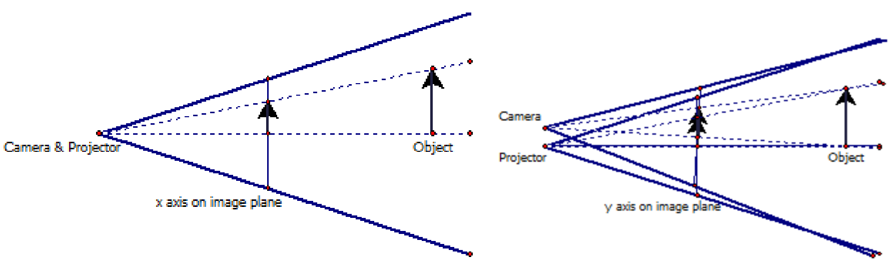

Fig. 8. Camera's translation on vertical axis. Left: top view shows the image on x axis projected on the same position as camera's. Right: left view shows the image on y axis overlapping between projector and camera.

3.2 Projection Accuracy

Projection accuracy is critical for projected reality. In experiments, the accuracy is evaluated by the error from the location tracking using projection crosshair pattern onto the target in Figure 9. The target consists of multiple radius circles with shared center which is infrared LED. There are 10 circles. The first one has a radius of an infrared LED's which is 0.25 cm. The range of each circle interval in target is 0.25 cm. The projection volume is separated into a quadrant which has the origin in the center of image. The test point is assigned randomly for 10 points per quadrant and is measured the error using the circle interval that shows the center of the crosshair. This experiment was run three times with different distances from projector to the plane at 60, 80, and 100 cm. From experimental results, most of projected points are close to the center of the target and placed in the first circle interval shown in Table 1. The class interval arithmetic mean is 0.25 cm.

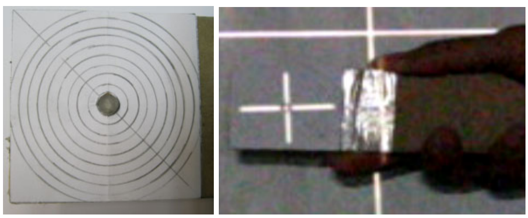

Fig. 9. Left: diffusion LED is pitched at the center of class interval. Right: the error is measured by projected crosshair.

Table 1. Projection Error

Error Intervals (cm)	Frequency(f)
0.00-0.25	99
0.25-0.50	21
0.50-1.00	0

3.3 Processing Throughput

The expected throughput is more than the typical projector refresh rate at 60 fps. This is important for real-time projected reality with moving object. Two sections affect the processing throughput which are input and output section. The effect of input section depends on the refresh rate of the camera and the latency of matching. The effect of output section depends on the latency of warping and rendering graphics in OpenGL. In first section, the camera refresh rate is fixed at 97 fps. The matching algorithm is used only when the new infrared is activated or disappeared. In the other word, it is typically executed only when the surface display comes into the workspace for the first time. Hence, this overhead does not change the camera throughput significantly. For the output section, Table 2 shows the number of frames processed per second when it renders different kinds of graphics. The throughput decreases as the complexity of graphics texture increases. However, the throughput is still higher than the projector's refresh rate. This proposes system has ability to do a real time location tracking.

Table 2. Latency of texture rendering

Texture Type	Frame Rate (fps)
4 crosshair (non-texture)	196
Image	147
Video	123

4 Conclusions and Future Work

This system can augment graphics image onto the moving object using two Wiimotes and a portable projector with a pan-tilt head. Two Wiimotes are placed a little bit of translation on horizontal and vertical axes to track LEDs placed in each corner of the display surface. The system obtains 2D position from two Wiimotes, matches corresponding points, computes the 2D position of point in the projector's plane, and displays texture accurately fit to the surface. This research shows the use of two camera calibration algorithm to directly map 2D camera's coordinate into projector's coordinate using perspective projection geometry property. The camera inside the Wiimote has a limit of tracking capability for only four LEDs which is insufficient to track the complex surface. This would be an interesting avenue for future work to deal with complex surface tracking.

Acknowledgments. This research work is financially supported by the National Science and Technology Development Agency, Thailand.

References

1. Pinhanez, C.: The everywhere displays projector: A device to create ubiquitous graphical interfaces. LNCS, pp. 315–331 (2001)
2. Bandyopadhyay, D., Raskar, R., Fuchs, H.: Dynamic shader lamps: Painting on real objects. Citeseer (2001)
3. Karitsuka, T., Sato, K.: A wearable mixed reality with an on-board projector, p. 321. IEEE Computer Society, Los Alamitos (2003)
4. Ehnes, J., Hirota, K., Hirose, M.: Projected augmentation-augmented reality using rotatable video projectors, p. 35. IEEE Computer Society, Los Alamitos (2004)
5. Park, H., Lee, M.H., Kim, S.J., Park, J.I.: Surface-independent direct-projected augmented reality. In: Computer Vision–ACCV 2006, pp. 892–901 (2006)
6. Lee, J.C., Dietz, P.H., Maynes-Aminzade, D., Raskar, R., Hudson, S.E.: Automatic projector calibration with embedded light sensors, pp. 123–126. ACM, New York (2004)
7. Lee, J., Hudson, S., Dietz, P.: Hybrid infrared and visible light projection for location tracking, pp. 57–60. ACM, New York (2007)
8. Heckbert, P.: Projective Mappings for Image Warping. Image-Based Modeling and Rendering 15-869
9. Laforest, M.: Wiiuse API Overview (2008)
10. Lee, J.C.: Hacking the nintendo wii remote. IEEE Pervasive Computing 7, 39–45 (2008)
11. Wang, D., Huang, D.: Low-Cost Motion Capturing Using Nintendo Wii Remote Controllers