

A Tracking Method for 2D canvas in MR-based interactive painting system

Sandy Martedi^{1*}, Maki Sugimoto, Hideo Saito

Department of Computer Science
Keio University
Yokohama, Japan

^{1*}e-mail: sandy@hvrl.ics.keio.ac.jp

Mai Otsuki^{2*}, Asako Kimura, Fumihisa Shibata

Department of Computer Science
Ritsumeikan University
Shiga, Japan

^{2*}e-mail: otsuki@rm.is.ritsumei.ac.jp

Abstract— We have proposed a mixed reality based painting system. In this paper, we tackle a problem in our previous painting system that fully relies on magnetic sensor that is attached on the canvas; the users needed to detach and attach a sensor on the canvas during painting when they want to switch the canvas. Instead of that, in this paper, we aim to automatically detect the shape of the canvas for registration purpose. Using the shape or region detection method such as MSER (maximally stable extremal regions), we detect and track the shape on the canvas on the captured camera image. We then compute the camera pose for virtually overlay the painting result. Using the brush device, we can draw and paint freely on the tracked canvases. We show that using visual based tracking method, we can generate the equivalent result compared to the result of using the sensor.

Keywords- tracking, mixed reality, augmented reality, interaction

I. INTRODUCTION

Mixed reality (MR) is the technology which merges the real and virtual worlds in real time. Interacting with an MR space is a fruitful field, since users can see their hands, and touch actual objects and interactive devices directly.

We have developed the MR based painting system [11] (Figure 1(a)) which enables the user to do a digital painting on the real objects while directly holding the object. It makes digital painting more closely resemble real-world painting, because the brush shaped device provides the similar sensation of an actual painting brush using tip bending.

In previous system, we used the magnetic sensors to track the position and orientation of HMD (user's head), brush device, and the canvas (real object). Although we designed our system to support multiple canvases, because we used magnetic sensors, we need to detach one sensor and attach it to the other canvas every time, otherwise, prepare magnetic sensors as many as the number of canvases, that is cumbersome. To avoid this problem, we try to switch the tracking method of the canvases from physical sensor to vision-based tracking. We also consider the following requirements:

- Because user holds and moves the canvas and the brush device moves on the canvas during



(a) using a sensor on the canvas (round plate) (b) removing sensors on the canvas

Figure 1: Interactive painting system

painting; the tracking method needs robustness against occlusion.

- Small size of the fiducial markers is desirable to avoid interrupting user's creativity.

To satisfy these requirements, we apply a vision based tracking for tracking canvases that has a specific shape (Figure 1(b)). We developed a vision-based tracker that utilizes the outline of shapes that is printed in the canvas or the shape of the canvas itself. In this paper, for the initial step, we aim to achieve a robust tracking 2D canvas even if the canvas is occluded by the device or/and user's hands.

II. RELATED WORK

There have been many studies in the field of MR; however, IR (Infrared) markers, magnetic sensors and/or visual markers are/is used for tracking real object, device and the user's hands [6-8, 10, 12, 13, 15]. Vision-based virtual painting interaction has been demonstrated using Kinect Fusion [14]. Kinect fusion is able to reconstruct the 3D scene and the user can draw and paint virtually over the reconstructed scene. However, in the case using Kinect Fusion, only 3D surfaces can be identified as the painting canvas. Identifying 2D canvas such as paper will be an issue since Kinect considers the depth to identify and track objects. Another approach emphasized the tangible interaction using a projector to paint real objects (equipped with visual tracker) as introduced in Dynamic shader lamps [1].

For vision-based tracking, although ARToolKit and, ARToolKitPlus [15] are widely used, in our painting system, such marker interrupts user's creativity.

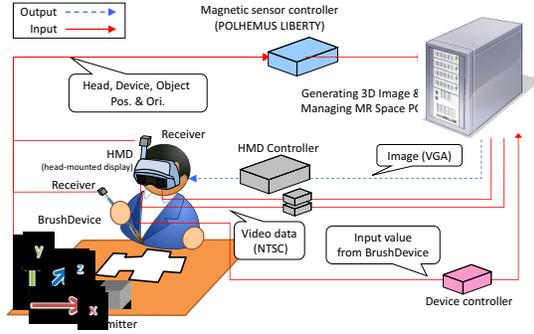


Figure 2: System configuration

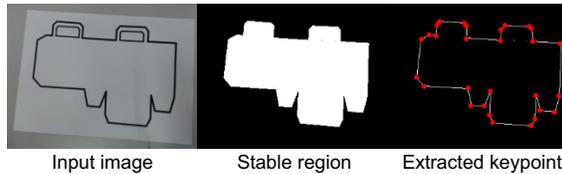


Figure 3: Keypoint extraction

While keeping the tangibility of our system, we aim to automatically detect real both 2D and 3D objects as the canvas. As for 2D, we use shapes or outlines tracking method in order to identify the canvas. Shapes registration methods for augmented reality has been explored actively in recent years. Bergig et al. have developed an application for augmented reality that recognizes 2D hand drawing in real time [2]. Their method recognizes 2D drawings and virtually overlays its corresponding 3D shapes. Explorations on arbitrary shapes registration is also done by Hagbi et al.[5]. Similarly, Donoser et al. proposed a shape tracking method using MSER (maximally stable external regions) detector [3]. Their tracking results are impressive, but the concavity is required to describe a shape. We aim to use more generic representation of shapes regardless the concavity of the shape by introducing a local shape descriptor and matching procedure.

III. PROPOSED METHOD

A. Setup

We modify our previous painting system [11] by removing the magnetic sensor that is attached on the canvas and instead we use planar canvas that has specific shape. The shape is used as the cue for the vision-based tracking. The system configuration is illustrated in Figure 2. The specifications for the main PC are as follows: Microsoft Windows XP OS, Intel Core i7 2600 CPU, and 3062 MB of RAM. We also use a binocular video see-through HMD (Canon VH-2002), which enables users to perceive depth. The HMD is connected to a video capture card (ViewCast Osprey-440), which captures input videos from the cameras built into the HMD. The NVIDIA GeForce GTS 250 graphics processor is used for image processing. The positions and orientations of the HMD and BrushDevice are tracked using Polhemus LIBERTY, a six-DOF tracking system that uses magnetic sensors. A transmitter is also used as a reference point for the

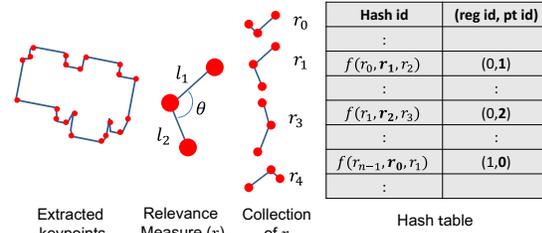


Figure 4: Descriptor matching

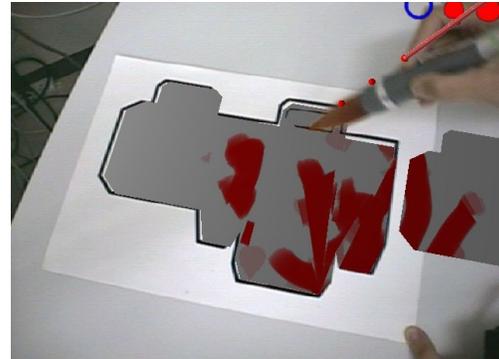


Figure 5: Painting result on a canvas that has a specific shape

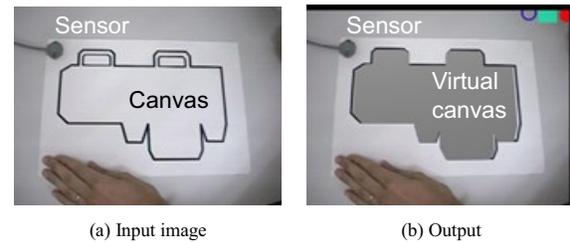


Figure 6: Evaluation setup

sensors. The brush device is connected to the main PC through an input/output (I/O) box. The I/O box retrieves data from the devices and sends them to the main PC.

All the codes in the system are written in C++/CLI in .NET Framework. We used OpenGL and the OpenGL Utility Toolkit (GLUT) for the graphics API. In creating the MR space, we first set the videos captured by the Osprey-440 as the background and then create a virtual viewing point in OpenGL by obtaining the position and orientation of the HMD from Polhemus LIBERTY. In done so, users feel as if they are manipulating virtual objects in the real world.

B. Contour Tracker

1) Descriptor

The first step in our method is extracting the region of the canvas. The popular and recent method such as MSER can be applied. We also can use the simple color segmentation for simple outline as illustrated in Figure 3. The next step is extracting the outline by using contour estimation. The outline that may contain many points is simplified using DP polygon simplification [4]. The remaining points are used as keypoints. The next step is

computing the descriptor r (relevance measure) that is computed using three consequent points in the simplified polygon. r is defined as

$$\frac{\theta l_1 l_2}{l_1^2 + l_2^2} \quad (1)$$

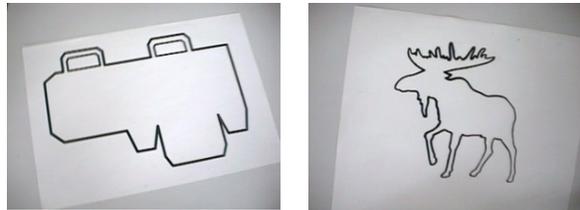
where l_1 and l_2 are the length of two connected segments (lines) and θ is the angle between where l_1 and l_2 ($0-\pi$). r depends on θ and the ratio of both segments. We assume these properties will not change drastically due to scale and rotation changes. More specifically, θ will not change due to rotation and the ratio will not change due to scale changes. However, these properties may change due to perspective change. We handle the perspective change by tracking step that will be explained later on the registration method subsection. Computing descriptors of the shape is done both off-line and on-line. In off-line process, the descriptors are stored in the database whereas in on-line process, the descriptors of unknown shape are matched with the descriptors in the database (see Figure 4).

2) Registration

During the shapes registration, the keypoints of unidentified shape are extracted. The sequences of relevance measures are then computed and the corresponding tuple (region or shape id, keypoint id) is looked up in the hash table (see Figure 4). Since the hash table is many-to-one relationship, in order to get the matched region and keypoint, histogram matching (voting) is performed. This process yields keypoints correspondences between a shape captured in the camera and a shape in the database. We choose three neighboring relevance measure values to represent a keypoint of a shape. In this case, one keypoint is actually described by its four neighbor's keypoints. For example r_0, r_1, r_2 represents a keypoint with id=1. In our implementation, we increase the number of relevance measure to four in order to create distinctive representation of a keypoint for example r_0, r_1, r_2, r_3 for keypoint with id=1. During runtime, when a shape is matched to one in the database, sequences of relevance measure of the matched shape are added to the hash table so that the registration becomes robust against perspective change.

3) Pose Estimation

The camera pose is estimated using homography that is calculated using at least four keypoints correspondences as the result of the shape registration. The outliers from the keypoints are removed using the inverse homography. The camera pose is then optimized using Levenberg-Marquardt [9] by minimizing the re-projection error that is the distance between the projected keypoints from the shape database and the extracted keypoints in the captured frames. The camera pose is then refined by considering the keypoints correspondence to the detected shape in previous frame. These two optimizations produce a stable camera pose.



(a) Craft (b) Reindeer

Figure 7: Two markers for evaluating the processing speed

Table 1: Computation cost

Task	Craft (33 pts)	Reindeer (79 pts)
Detection and tracking (msec)	23.74	36.03
Painting (msec)	11.39	9.5

4) Unifying the Coordinate System

One issue to switch from a magnetic sensor to a visual tracker is the unification of coordinate system. Since in the vision-based tracking, the coordinate system is independent from the sensor coordinate system, the transformation from the camera coordinate system into the sensor coordinate system is necessary. This unification is also required in order to enable the painting interaction since the brush and eraser device are located in the sensor coordinate system.

The unification is quite straightforward since the tracking is done using the camera attached in the HMD. Note that a magnetic sensor is also attached in the HMD which make the unification becomes simple. Therefore, the unification is done by computing the transformation matrix for any object in the camera coordinate system. The transformation matrix is computed by multiplying the view matrix of the HMD by the rotation and translation matrix retrieved using homography as the result of the shape registration. As a result, the virtual canvas can be painted using the brush device that has a magnetic sensor on it. The result of the painting can be seen in Figure 5.

IV. EVALUATION

We evaluate how accurate the registration results from the vision-based tracking. In order to check the accuracy of vision-based tracking method in our system, we compare the positions produced by magnetic sensor and vision-based tracking.

A. Setup

We attach a magnetic sensor on the origin of the canvas in the tracker coordinate system (camera coordinate system) as seen in Figure 6. We retrieve the position information from the sensor and from the tracking by camera. We then observe the value of these positions in several frames. We do not observe the rotation produced by the magnetic sensor and the canvas since we assume that the rotation will be exactly same because the magnetic sensor is attached to the canvas.

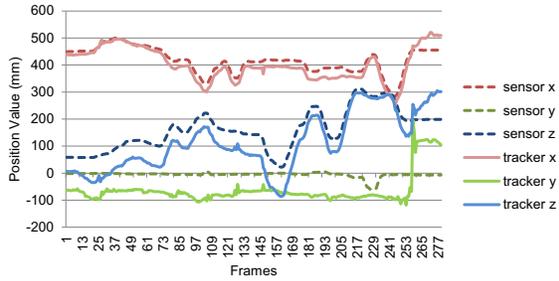


Figure 8: Comparison of the position value of the magnetic sensor and the proposed tracker

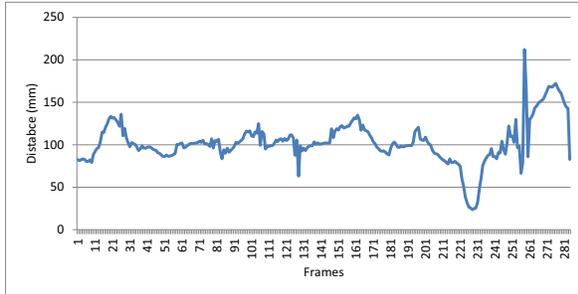


Figure 9: Euclidean distance between the sensor and the proposed tracker

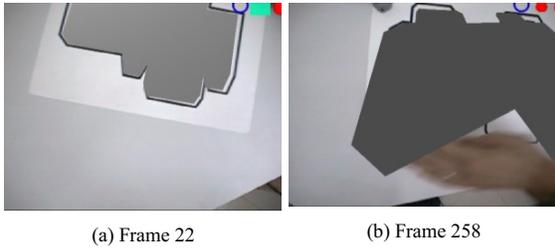


Figure 10: High error due to extreme occlusions and rotations. The grey area is the virtual canvas that is overlaid above the shape printed in a paper. Some keypoints are located outside the capture area which makes the accuracy decreases (a) or the tracking failure (b).

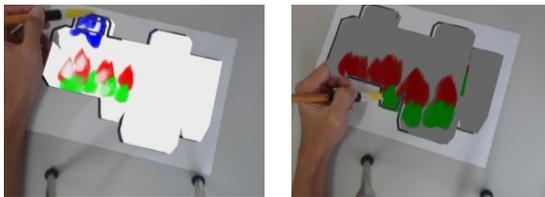


Figure 11: Prototype of the painting system using monocular setup.

B. Results

1) Accuracy

We plot the difference between the positions in Figure 8. As plotted in the graph, the position of the magnetic sensor is shifted in the y axis due to the calibration error. We compensate this issue by applying a local transformation on the camera coordinate system to finalize the unification of the coordinate system. The Euclidean distance between the magnetic sensor and the proposed vision-based tracking method become high due to occlusions around frame 22 and tracking failures

around frame 237-284) as shown in Figure 9 and Figure 10.

2) Processing speed

We captured two shapes for testing the speed of the visual tracking on the painting system. We provided two shapes (Figure 7) that have different number of points in the outline. Theoretically, the number of points in the polygon contributes to the computation cost since our method involves the matching features on the database that is done in the keypoint level. This is proven by experiments that a shape that has 79 points (reindeer shape) is processed longer than shape with 33 points (craft) as listed in Table 1. In order to get a constant processing time, the polygon can be simplified into polygon that has a particular number of keypoints (for example 33 keypoints).

V. DISCUSSION

A. Limitations

The proposed method is for replacing the sensor attached on the canvas object into a visual tracker. The visual tracker is beneficial since the sensor is no longer necessary for each canvas object. However, the visual tracking has some limitations. Visual tracking only relies on the camera for recognizing the object. Therefore, the disadvantage such as occlusions is unavoidable. As stated in the evaluation, our proposed method will fail when many occlusions occur. Even though, partial occlusions can be compensated.

The accuracy of the system also decreases when the tracking fails. It occurs when the camera is in extreme position and orientation. In addition, rapid motion of the camera will create blur area on the image that makes the contour and keypoints become difficult to extract.

Furthermore, when the outlines of the shape are not detected, the location information becomes inaccessible. On the other hand, on sensor-based system the sensor information is continuously transmitted. Therefore, even when the object is invisible by the camera, the location of the object is still accessible. However, in this system, we assume that the user always see the canvas for painting, generally that makes the outlines of the shape always detectable.

B. Monocular-based system

A prototype of the proposed method can be implemented as a monocular system where a web camera can be used as the alternative of the HMD. Instead of using brush device with magnetic sensors, the canvas tracking can be done entirely using vision-based method. In this case, it is necessary to track the brush or eraser using vision method such as color detection. A prototype using monocular system with color detection as the replacement of the sensor in the brush is shown in the Figure 11.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a mixed reality-based painting system that uses vision-based tracking for replacing a magnetic sensor that is attached on the canvas. We use a visual tracker that uses region or shape as

tracking feature. Our proposed system has improved the conventional system that uses sensors in each canvas for painting. As a result, when we use multiple visual trackers, we will be able to switch the various canvases freely without detaching and attaching the physical sensors.

In this paper, we used the asymmetry 2D object as the tracking marker as the canvas, but in the future, we are planning to extend our method to 3D objects. Using 3D objects as the marker, it is possible to paint on any object that of course must be detectable objects. In addition, we currently track one canvas in this research; we are planning to add an ID to each canvas so that we can use multiple canvases simultaneously.

REFERENCES

- [1] Bandyopadhyay, D. Raskar, R., and Fuchs, H. "Dynamic shader lamps: Painting on movable objects." Proc. ISMAR 2001, pp. 207-216. 2001.
- [2] Bergig, O., Hagbi, N., El-Sana, J., and Billinghurst, M. "In-place 3d sketching for authoring and augmenting mechanical systems." Proc. ISMAR 2009, pp., 87-94, 2009.
- [3] Donoser, M., Kotschieder, P., and Bischof, H. "Robust planar target tracking and pose estimation from a single concavity." Proc. ISMAR 2011, pp. 9-15, 2011.
- [4] Douglas, D H., and Peucker, T. K.. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." Cartographica: The International Journal for Geographic Information and Geovisualization, Vol.10, No. 2, pp. 112-122, 1973.
- [5] Hagbi, N., Bergig, O., El-Sana, J., and Billinghurst, M. "Shape recognition and pose estimation for mobile augmented reality." IEEE Trans. Visualization and Computer Graphics, Vol. 17, No. 10, pp. 1369–1379, 2009.
- [6] Kato, H. and Billinghurst, M. "Marker tracking and HMD calibration for a video-based augmented reality conferencing system." Proc. IWAR 1999, pp. 85-94, 1999.
- [7] Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. "Virtual object manipulation on a table-top AR environment." Proc. ISAR 2000, pp. 111–119, 2000.
- [8] Lee, G. A., Kim, G. J., and Billinghurst, M. "Immersive authoring: What you experience is what you get (wyxiwyg)." Comm. ACM, Vol. 48, No. 7, pp. 76-81, 2005.
- [9] Lourakis, M. "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++." [web page] <http://www.ics.forth.gr/~lourakis/levmar/>, Jul. 2004. [Accessed on 23 Aug. 2013.].
- [10] Ohshima, T., Satoh, K., Yamamoto, H., and Tamura., H. "AR2 Hockey: A case study of collaborative augmented reality." Proc. VRAIS '98, pp. 268-275, 1998.
- [11] Otsuki, M, Sugihara, K., Kimura A., Shibata, F., and Tamura, H. "MAI painting brush: An interactive device that realizes the feeling of real painting." Proc. UIST2010, pp. 97-100. 2010.
- [12] Ryu, C., Cho, Y., and Chai, Y.. "Development of an Immersive VR display system for 3d digital art". Proc. CGIV '04. pp. 93-98, 2004.
- [13] Schkolne, S., Pruett, M., and Schröder, P. "Surface drawing: creating organic 3D shapes with the hand and tangible tools." Proc. CHI '01, pp. 261-268, 2001.
- [14] Shahram, I., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, D., Kohli, P., Jamie Shotton et al. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." Proc. UIST 2011, pp. 559-568, 2011.
- [15] Wagner, D. and Schmalstieg, D. "ARToolkitPlus for pose tracking on mobile devices." Proc. CVWW'07, pp. 139-146. 2007.