電気学会論文誌 C（電子・情報・システム部門誌）
IEEJ Transactions on Electronics, Information and Systems
Vol.133 No.3 pp.672–679 DOI: 10.1541/ieejeiss.133.672

Paper

Feature-based Alignment Method for Projecting Virtual Content on a Movable Paper Map

Sandy Martedi[∗a)] Non-member, Maki Sugimoto[∗] Non-member
Hideo Saito[∗] Member, Bruce Thomas[∗∗] Non-member

(Manuscript received April 13, 2012, revised Oct. 16, 2012)

In this paper we propose a feature-based alignment method for visualizing geographical content on a movable paper map using non-calibrated projector-camera pair. In order to project correctly aligned geographical content on a movable paper map, we detect two kinds of features: printed features and projected features on a paper map. First, we print landmark points as map features on the paper map for registration purpose. We then project landmark points onto the paper map. We then detect these printed and projected features simultaneously. The transformation from the projector coordinate system to the image coordinate system is estimated using the feature correspondences. This transformation is used to warp the geographical content before the corrected projection. In succeeding frames, the features are tracked in order to update the transformation. We evaluated our method against the motion of the paper map, the camera and the projector with average projection error of 9.5 pixels.

Keywords: spatial augmented reality, random dot marker method, geographical information system

1. Introduction

In order to display additional information on a paper map, virtual contents can be overlaid using a technique so called augmented maps. Traditional augmented maps use either a monitor or a head mounted display. When monitors are employed, the user can only move around the monitor display. Hence, it limits the user freedom and is impractical for outdoor system. On the other hand, a head mounted display is required for each user to view the virtual content at the same time.

Recently, projectors are used to augment virtual contents onto a real surface by projection mapping method called spatial augmented reality (3) (14). Recent investigations removed the limitation of spatial augmented reality and allowed the user movement using a portable projector attached on a mobile phone (5) (8) (15) (16).

A spatial augmented reality system usually places a target surface and a projector stationarily. In order to realize a portable system where user can hold the surface and move it freely, projecting contents on a movable surface becomes necessary. A planar tracker is usually used for registering a movable surface and projecting virtual contents in aligned position on the surface (6) (9).

We propose a feature-based method for aligning virtual contents on a movable planar paper map using a projector-camera pair without any special devices such as light sensors, motion sensors, or infrared cameras (see Fig. 1). In our wearable and surveillance setup, camera, projector, and target surface can move independently. The benefit of our method in those setups is the automatic calibration (camera and projector pose estimation) during runtime by detecting the target surface and the projected content simultaneously.

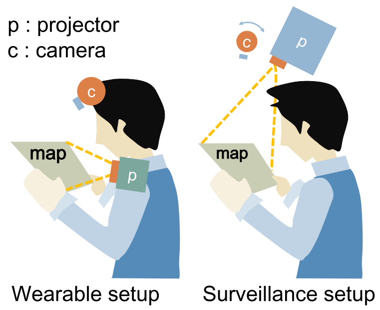


Fig. 1. A wearable and surveillance setup allow the arbitrary paper movement.

In this paper, we also aim to develop a method that is applicable for pre-calibrated (integrated) and non-calibrated setup without losing a generality. Therefore, we mention that any setup such as integrated setup or free moving projector-camera pair could be realized using our method. However, in order to explain further about the benefit of our method, we present two kinds of applicable setup using free-moving projector and camera (non-calibrated) dedicated for movable map applica-

a) Correspondence to: Sandy Martedi. E-mail: sandy@hvrl.ics.keio.ac.jp
[∗] Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
[∗∗] University of South Australia
Mawson Lakes, SA 5095, Australia

tion. Note that the non-calibrated setup is desirable and ideal setup that has been the goal of recent researches in spatial augmented reality field. Regardless fixing our setup beforehand (pre-calibrated) would be technically feasible and easier to implement.

Compared to the integrated setup where the camera and projector are fixed and precalibrated, our method allows the user to move arbitrarily while holding the target surface. In this case, the camera detects the target surface and projector automatically projects the aligned content. The user movement and the unintentional geometrical changes between projector and camera in each frame will thus require re-calibration. Our method can automatically compensate these changes and estimate the camera and projector pose in real time.

We apply the random dot marker method for registering features in a movable paper map because it is robust to rotation, perspective, scale changes and partial occlusion. First, we print map features on the paper map as geographical layer. During runtime, we project the same map features in different color from the printed ones. We detect both printed and projected features. We then estimate each homography of the printed and projected features relative to map features in database. We calculate a transformation matrix between these two homographies. The transformation matrix is used to warp virtual contents so that it will be projected aligned to the paper map.

Our novelty lies on the non-calibrated projection mapping method that uses the random dot marker printed and projected on a movable paper. To the best of our knowledge, there is no method that explores a random dot marker as a replacement of light sensor or structured light for aligning virtual contents on a movable paper yet. Conventional methods depend on precalibration and special devices such as light sensors, motion sensors, or infrared cameras whilst in our method those requirements are not necessary.

The rest of the paper describes some related works in Section 2. Section 3 describes our proposed method. Section 4 describes the implementation of our method. The accuracy of our method is evaluated in Section 5. We discuss our future works in Section 6. Section 7 concludes this paper.

2. Related Works

Generally, spatial augmented systems use camera or sensors for registering a movable surface and aligning projected contents. Marner and Thomas proposed a method using a sensor attached on a digital brush for coloring 3D objects by projecting colors on the objects [9]. Similarly, Lee et al. put a sensor on each corner of a surface and automatically register the position of the surface [6]. Sensor based method is accurate and fast but sensors must be attached on the surface beforehand.

On the other hand, a camera-based system usually depends on a stationary and pre-calibrated projector-camera pair. Ehnes and Hirose used an AR marker for tracking a movable surface [4]. Takao et al. detected surface corners and corrected the projection using a homography matrix [18]. Recently, Audet and Okutomi proposed a method that uses a surface texture and projected pattern for aligning virtual contents [2]. Similarly, a feature based approach using border tracking was proposed by Leung et al. [7].

A combination of a particle filter method and a motion sensor method has been proposed [17]. This method calibrates the camera and projector in each frame so that the mapping between the camera coordinate and projector coordinate can be estimated during runtime.

In this paper, our method only depends on projector and camera setup. We use a random dot marker method for map registration and print the dot marker on a paper map. Our method is similar to the active registration using binary coded patterns [21] and the simultaneous projection of structured light pattern [1]. However, instead of projecting coded patterns, we project dots as the map features.

3. Proposed Method

An alignment of virtual contents on a movable paper map is achieved by image warping before the projection. First, we project landmark points contained in a map (reference dots) in blue color. Note that the same landmark points are printed in the paper map as red icons beforehand.

In each frame we detect the printed red icons and estimate a homography H_c that relates the reference dots in geographical coordinates system into the image coordinate system. We also detect the projected landmark points and estimate a homography H_p that relates the reference dots in geographical coordinate system into the image coordinate system (see Fig. 2).

Using H_c and H_p, we estimate a homography that transforms the geographical content (in the geographical coordinate system) into the aligned position on the paper map in the projector coordinate system as illustrated in Fig. 3. The complete algorithm for estimating the transformation including the initialization and the transformation update is listed Fig. 4.

3.1 Initialization First, we project landmark points as random dots in blue color. We estimate a transformation (homography) between the projected random dots to the printed red icons as follows:

$$X_a = HX_p, \quad \cdots\cdots (1)$$

$$H = H_p^{-1}H_c, \quad \cdots\cdots (2)$$

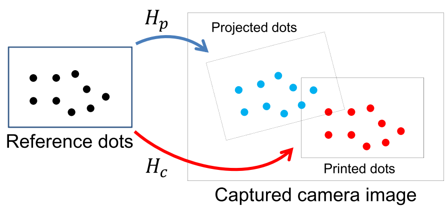


Fig. 2. In a captured image, both the printed red icons and projected blue dots are detected. We then estimate the planarity or homography of the projected blue dots (H_p) and printed red icons (H_c).

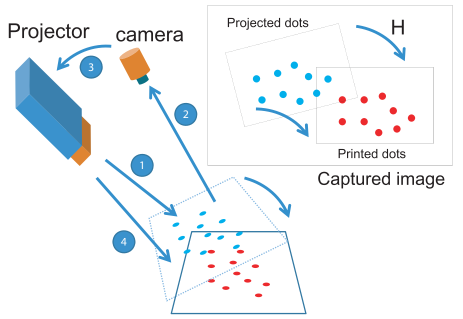


Fig. 3. Transformation estimation. After H_c and H_p are estimated, we calculate a homography (H) that relates the projected blue dots planarity to the printed red icons planarity. We then warp virtual contents using H.

Require: H_p, H_c
  if *init* **then**
    computeH ▷ (*Equation* 2)
  else
    $X_p \leftarrow H_p X$
    $X_c \leftarrow H_c X$
    $e \leftarrow E(X_p, X_c)$ ▷ (*Equation* 3)
    if $e > range_0$ **then**
      $init \leftarrow true$
    else if $e < range_0$ **and** $e > range_1$ **then**
      $H_p \leftarrow H_{p0}$
      computeH ▷ (*Equation* 5)
    else
      computeH ▷ (*Equation* 5)
      $H_{p0} \leftarrow H_p$
    end if
  end if
  return H

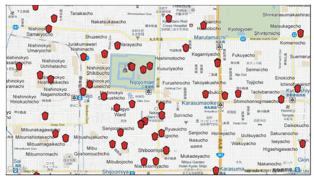
Fig. 4. The algorithm for computing H in every frame.

where the H_c is the homography that maps the reference dots into the printed dots and H_p is the homography that maps the reference dots into the projected dots captured by the camera. X_p is the original contents that we want to project, X_a is the final output of our method which is the aligned virtual contents, and H is the output homography for the alignment.

3.2 Transformation Update After the projected blue dots are detected, we warp the blue dots to the location of the corresponding red icons. We define this condition as aligned state. In each succeeding frame, it is necessary to continue warping virtual contents into the aligned coordinate system. Therefore, the blue dots are re-detected and the homographies (H_c and H_p) are re-estimated. Before computing the transformation (warping) function (H), the error between two homographies is computed using the following error function:

$$E(X'_p, X'_c) = \frac{1}{4}\sum_{i=1}^{4} d(X'_{pi}, X'_{ci}) \cdots\cdots\cdots\cdots\cdots (3)$$

where $d(X'_{pi}, X'_{ci})$ is the euclidean distance between each corresponding four corners of the boundary of the map and the projected dots in the image coordinate system. When $E(X'_p, X'_c)$ is inside a $range_0$ to $range_1$ the H_p will be updated using the old H_p and if it exceeds a threshold, the method reinitializes.



(a) A map image ©2011 Google, ZENRIN

(b) Map labels (c) Navigation (d) Photo

Fig. 5. A map image and geographical contents.

The transformation H is updated as follows:

$$X_{a^{t-1}} = H_t X'_p, \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4)$$

$$H = H_t H_c, \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (5)$$

where the H_t is the homography that maps the captured blue dots in the image (X'_p) into the blue dots in the aligned coordinate in previous frame ($X_{a^{t-1}}$). The homography for transforming the content for the current frame (H) is estimated by multiplying H_t with the homography that relates the red icons captured in the image with the reference dots in database H_c. Thus, we use the correct H calculated from the previous frame.

4. Implementation

In order to detect the projected and printed landmark points, we extend the detection method in our previous works [(10)–(12)].

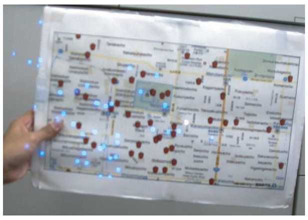
4.1 Map and Geographical Contents We use a paper map that contains landmark points and a background map. We overlay landmark points as red icons on top of the background map as illustrated in Fig. 5 (a). We use geographical contents such as labels, navigation information and photo as virtual contents as illustrated in Fig. 5 (b),(c), and (d).

4.2 Features Extraction In order to extract map features, we use color-based extraction. First, we separate the input image into three color channels R, G, and B. We then subtract the value of R channel with (G+B) channel. We then binarize the result to get blobs of red icons. We compute the center of each blobs as the features.

We use the similar way to extract the blue projected dots by subtracting the value of B channel with (R+G) channel. The result of the feature extraction is illustrated in Fig. 6 (b) and (c).

4.3 Paper Map Registration We use a random dot marker method [(20)] for surface detection and tracking because of its robustness to partial occlusions. Partial occlusions occur because projected contents may alter the captured features in the paper map. The random dot marker method uses the affine invariant descriptor computed from the combination of neighbours dots (Fig. 7).

In an offline phase, we create a descriptor database



(a) A captured image containing red icons and projected blue dots

(b) Extracted red icons (c) Extracted projected blue dots

Fig. 6. A captured image and features extraction.

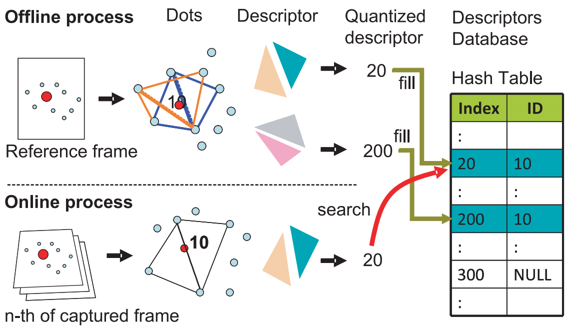


Fig. 7. The map registration. An offline process estimates descriptors and stores them into database. An online process extracts dots from captured image and computes the descriptor followed by searching the keypoint id in the database.

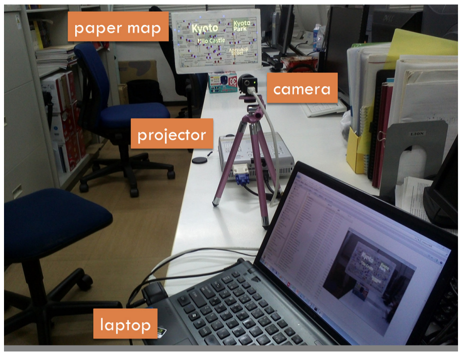


Fig. 8. The experiment setup.

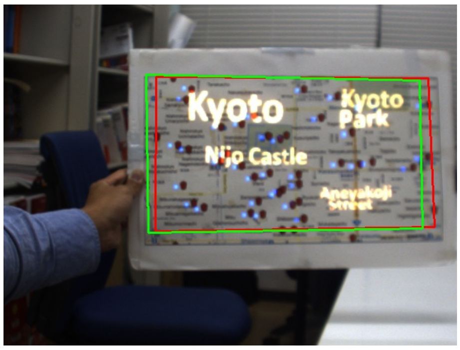


Fig. 9. We estimate the projection error by averaging the distance between four corners of the red border and green border. Red border is the projection result to image coordinate by the H_c. The green border is the projection result to image coordinate by the H_p.

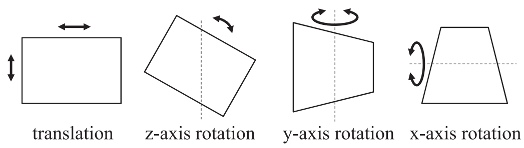


Fig. 10. The paper map motion (translation and rotation).

from a text file containing the location of landmarks (reference dots) in the geographical coordinate system. The paper map is registered using the following steps. The red icons printed on the surface are extracted based on color (Section 4.2). Then the extracted keypoints are matched with the reference dots stored in the database. Finally, we achieve the keypoints correspondences and the homography (H_c) is calculated. The projected blue dots are detected similarly and matched with the same reference dots. The homography H_p is then calculated.

5. Evaluation

5.1 Setup For experiments, we use Intel (R) Core (TM) i7 CPU M 640 2.8GHz 4GB RAM laptop and 640 × 480 pixel Point Grey camera. We use PLUS U4-232h projector in 1024×768 image resolution. We implement our codes in C/C++ with OpenCV [13] and OpenGL for rendering. We print our maps on A3 size paper. We placed a camera behind a projector and put a paper map in front of the projector as illustrated in Fig. 8.

5.2 Accuracy We observed the projection error by calculating the distance between two boundaries projected by H_p and H_c as illustrated in Fig. 9. We examined the first 170 frames and calculated the error function using the Equation 3.

Three conditions were observed based on the paper map motion, the camera motion and the projector motion as listed in Table 1. The projection error is plotted in Fig. 11, 12, and 13 respectively.

Table 1. The projection error.

Condition	Average distance (pixels)	Standard deviation
Paper map moves	11.9	5.3
Camera moves	9.3	8.4
Projector moves	7.4	5.6
Average	9.5	

The average error due to the paper motion such as translation and rotation (see Fig. 10) in the first 170 frames is 11.93 pixels. In Fig. 11, the error suddenly increases in some frames. In frame 16, the blue dots are not easily detected because it is necessary to store information from the previous frames until the tracking becomes stable (less jitter). When the paper map remains stationary in frame 46, the error reaches minimum. The paper map is rotated along z-axis from frame 46 to frame 80. Because of the delay of the projection, the projected blue dots can not be projected at the same time as the paper map moves and the error reaches maximum in

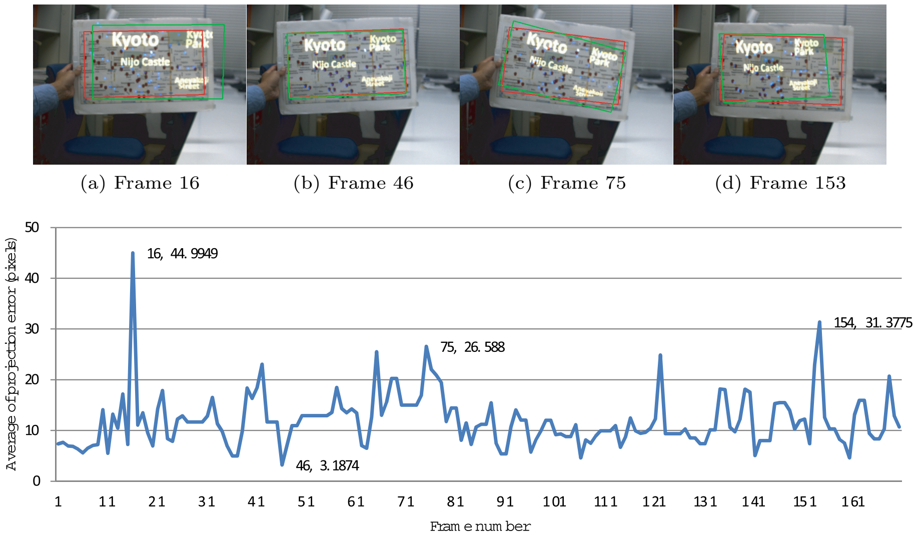


Fig. 11. Projection error in 170 frames due to map motion. Frame 1-16 is the initialization process. The paper map is translated in frame 17-45. The paper map is rotated along z-axis in frame 46-80. The rest is the paper map is rotated along x-axis and y-axis.

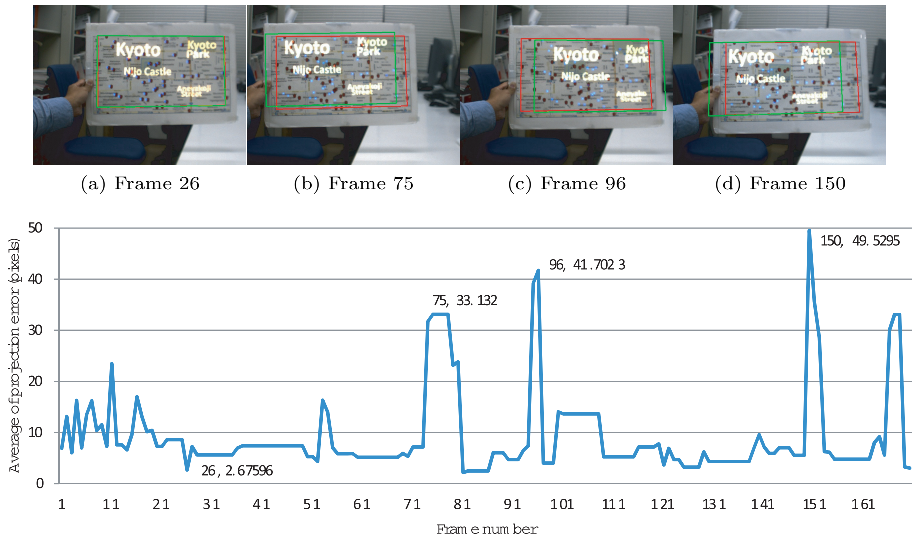


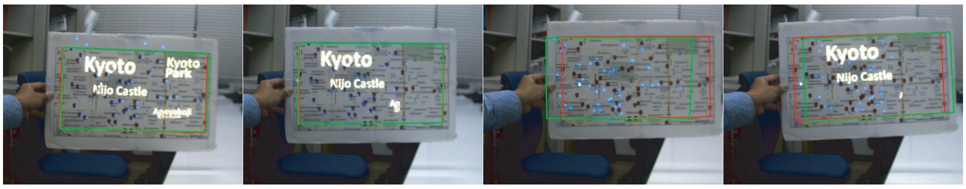
Fig. 12. Projection error in the first 170 frames due to the camera motion (rotation along y-axis). The error is relatively lower than due to the paper map motion.

frame 76. The paper map is rotated along x-axis and y-axis from frame 81 to frame 170 and the error reaches maximum when the detection fails in frame 154. After some detection failures (large projection error), we show on the graph that our method re-initializes the detection for the next alignment.

The mapping error in all cases are caused by the accuracy of H, H_c and H_p. The accuracy of those matrices is influenced by the number of matched features. Note that the number of matched features is influenced by the feature matching (registration) result. In Fig. 11, 12, and 13, the big errors are caused by the abrupt movement of the paper map, camera or projector which make some of the features are not extracted or occluded and thus unmatched. Therefore, in our paper we measure the observable error by evaluating the influence of the keypoint correspondence to the mapping errors. Fig. 11, 12, and 13 show that abrupt movements of the paper, camera or projectors will reduce the number of matched features and thus increase the mapping error. Fig. 11, 12, and 13 also show that even after the big mapping error occurs, our method can reinitialize and reduce the mapping error automatically.

The average error due to the camera motion (see Fig. 12) in the first 170 frames is 9.28 pixels. The error is lower compared to the paper map motion. However the extreme camera motion produces big errors as shown in frame 75, 96 and 150. The average error due to projector motion (see Fig. 13) in the first 170 frames is 7.42 pixels. Similar to the camera motion, the extreme projector motion produces big errors as shown in frame 62 and 126.

5.3 Speed and Alignment Results We calculated the processing time using the same videos from the accuracy experiments. We achieve the average computation time 75.6 msec in each frame. In other words, our



(a) Frame 2 (b) Frame 54 (c) Frame 62 (d) Frame 126

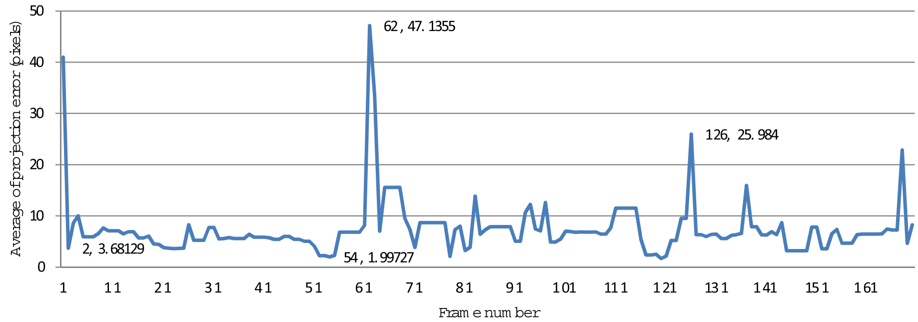


Fig. 13. Projection error in the first 170 frames due to the projection motion (rotation along y-axis).

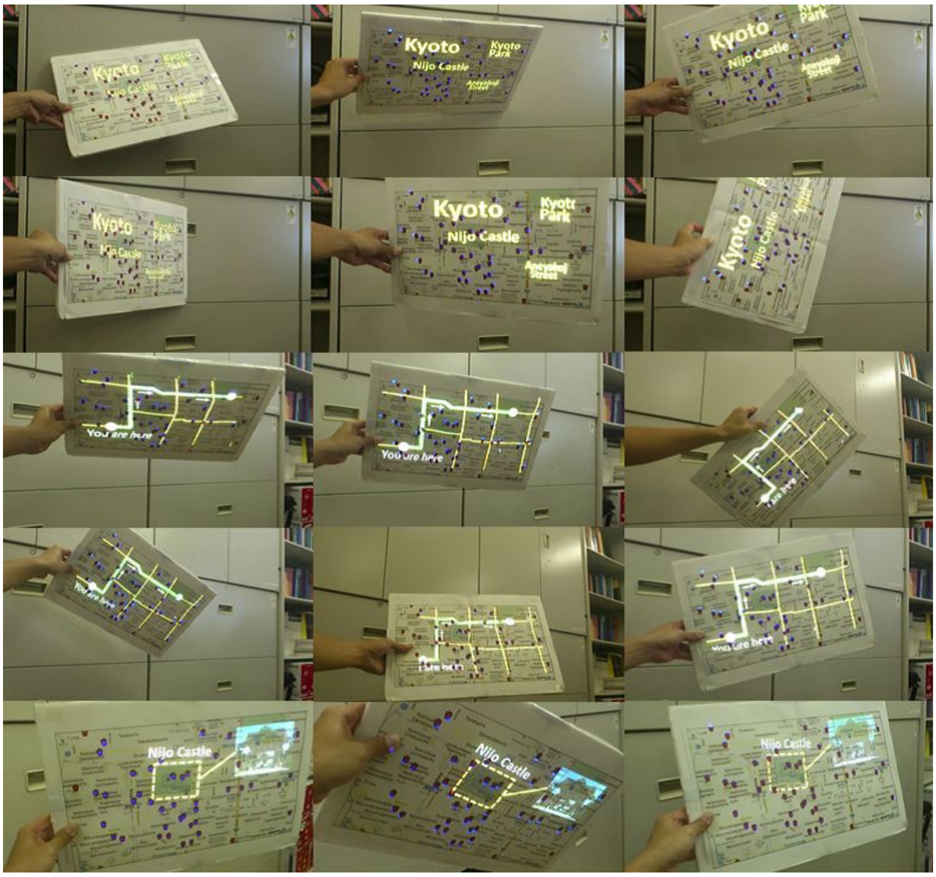


Fig. 14. Projection mapping results. Our method can compensate arbitrary paper map motions.
Row 1 and 2 show the projected text `http://youtu.be/L43wVrn-M_0`
Row 3 and 4 show the projected navigation information `http://youtu.be/tmG7U4I5a0A`
Row 3 and 4 show the projected photo or image `http://youtu.be/31l99RF4lik`

method can run in approximately 14fps. This computation cost is small and thus fast enough for augmented reality application. The results of the alignment on a movable paper map are shown in Fig. 14. Furthermore, the speed of the projection can be examined in the videos provided. As described in previous section, we compute the projection mapping error in each projection. When the error exceeds a threshold due to abrupt motions, our method re-initializes and maintains the correct projection as demonstrated in the result videos as well.

Although the computation time required for alignment is small, transferring the contents into projector and capturing the new projected content takes longer than 75.6 msec. In order to get the correct content from the previous projection, we intentionally add 100 msec delay in each iteration before starting to capture the next frame. This delay value is determined by some trials that show values lower than 100 msec produced false detection.

6. Future Works

For general purpose, the target map can be replaced with a textured surface such as a photo. Annotations or videos can be projected instead of geographical contents. In this case, robust features on a target surface instead of predefined red icons are necessary as previously explored by Uchiyama and Marchand [(19)]. They mentioned that the random dot marker method is proven applicable for a textured surface. The next challenge is improving the performance of our alignment method against a strong projector light. This issue can be solved by taking into account the illumination model on the projected image and the surface image as proposed by Audet and Okutomi [(2)].

Our current implementation runs in approximately 75.6 msec. By removing the delay for compensating the projector latency, the synchronization of the frame and the correct timing of the projection should be handled. This can be solved by projecting and detecting an ID on each frame in order to determine the correct timing for the projection.

7. Conclusion

We have presented an alignment method on a movable paper map using a projector-camera pair. In order to allow the paper map movement, we apply the random dot marker method for registering and tracking the paper map and warp the virtual contents onto the aligned position. Thanks to the update of the transformation in each frame, the geometrical changes between the projector and the camera can be compensated. The novelty of our method is the successful aligned projection without depending the calibrated projector-camera pair nor specific devices such as motion sensors, light sensors or infra red cameras. Our proposed method can track the movable paper map and project the aligned contents with projection error of 9.5 pixels.

Acknowledgment

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for high-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan and Grant-in-Aid for JSPS Fellows. We also thank Hideaki Uchiyama for providing the random dots marker source code.

References

(1) K. Akasaka, R. Sagawa, and Y. Yagi: "A Sensor for Simultaneously Capturing Texture and Shape by Projecting Structured Infrared Light", In Proc. 3DIM, pp.375–381 (2007)

(2) S. Audet, M. Okutomi, and M. Tanaka: "Direct image alignment of projector-camera systems with planar surfaces", Image (Rochester, N.Y.), pp.303–310 (2010)

(3) O. Bimber and R. Raskar: "Spatial Augmented Reality: Merging Real and Virtual Worlds", Vol.6 (2005)

(4) J. Ehnes and M. Hirose: "Projected Reality - Content Delivery Right onto Objects of Daily Life", IJVR, Vol.5, No.3, pp.17–23 (2006)

(5) A. Kushal, J. V. Baar, and R. Raskar: "A handheld projector supported by computer vision", In Proc. ACCV), pp.183–192 (2006)

(6) J. C. Lee, S. E. Hudson, E. Tse, A. Sw, and C. Tc: "Foldable interactive displays", In Proc. UIST, pp.287–290 (2008)

(7) M. C. Leung, K. K. Lee, K. H. Wong, and M. Chang: "A projector-based movable hand-held display system", In Proc. CVPR, pp.1109–1114 (2009)

(8) M. Löchtefeld, W. Str, M. Rohs, and A. Krüger: "LittleProjectedPlanet: An Augmented Reality Game for Camera Projector Phones", In Artificial Intelligence

(9) M. R. Marner: "Digital Airbrushing with Spatial Augmented Reality", Techniques, 904490–904490 (2010)

(10) S. Martedi and H. Saito: "Foldable Augmented Papers with a Relaxed Constraint", In Proc. ISAS, pp.127–131 (2011)

(11) S. Martedi and H. Saito: "Towards Bendable Augmented Maps" In Proc. MVA, pp.566–569 (2011)

(12) S. Martedi, H. Uchiyama, G. Enriquez, H. Saito, T. Miyashita, and T. Hara: "Foldable Augmented Maps", In Proc. ISMAR, pp.65–72 (2010)

(13) OpenCV. OpenCV. http://sourceforge.net/projects/opencvlibrary/

(14) R. Raskar, G. Welch, and H. Fuchs: "Spatially augmented reality", In Proc. IWAR, pp.11–20 (1998)

(15) S. Robinson, M. Jones, E. Vartiainen, and G. Marsden: "PicoTales", In Proc. CSCW, p.671 (2012)

(16) E. Rukzio and P. Holleis: "Projector phone interactions: Design space and survey", In Proc. WCDVI (2010)

(17) T. Siriborvornratanakul and M. Sugimoto: "A Portable Projector Extended for Object-Centered Real-Time Interactions", In Proc. CVMP, pp.118–126 (2009)

(18) N. Takao, J. Shi, S. Baker, I. Matthews, and B. Nabbe: "Tele-Graffiti: A pen and paper-based remote sketching system", In Proc. ICCV, Vol.36, p.750 (2001)

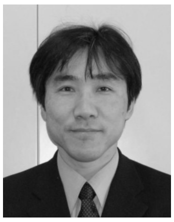
(19) H. Uchiyama and E. Marchand: "Toward augmenting everything: Detecting and tracking geometrical features on planar objects", In Proc. ISMAR, pp.17–25 (2011)

(20) H. Uchiyama and H. Saito: "Random dot markers", In Proc. IEEE VR, pp.35–38 (2011)

(21) S. Zollmann, T. Langlotz, and O. Bimber: "Passive-Active Geometric Calibration for View-Dependent Projections onto Arbitrary Surfaces", Virtual Reality, Vol.4, No.6 (2007)

Sandy Martedi (Non-member) received B. Eng degree from Bandung Institute of Technology, Indonesia in 2005. He received M. Eng degrees in engineering in 2010 and is currently in Ph.D course of Department of Computer Science at Keio University, Japan. His research interests are augmented reality, 3D reconstruction and Geographical Information System (GIS).

Maki Sugimoto (Non-member) received Ph.D. in Engineering from The University of Electro-Communications. He is currently the Assistant Professor of Department of Information and Computer Science in Keio University. He engaged in display-based computing systems and spatial augmented reality systems at The University of Electro-communications and Keio University. He also engaged in Parasitic Humanoid: a wearable robotic human interface at Japan Science and Technology Agency. His research interests include interactive systems and augmented reality environments with actuated interfaces.



Hideo Saito (Member) received the B. Eng., M. Eng., and Ph.D. degrees in electrical engineering from Keio University, Japan, in 1987, 1989, and 1992, respectively. He has been on the faculty of Department of Electrical Engineering, Keio University, Japan since 1992. From 1997 to 1999, he was a visiting researcher at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, where he joined the virtualized reality project. From 2000 to 2003, he was also a researcher with PRESTO, JST. Since 2006, he is a professor in the Department of Information and Computer Science, Keio University. He is a member of IEICE, IPSJ, VRST, ITE, IIEEJ, and SICE. His research interests include computer vision, image processing, augmented reality, and human-computer interaction.

Bruce Thomas (Non-member) received a BA in Physics from George Washington University in 1980, and a MS in Computer Science from the University of Virginia in 1984. He received a Ph.D. in Computer Science from Flinders University in 1998. He is the Deputy Director of the Advanced Computing Research Centre, Director of the Human Embedded Systems Group, and Director of the Wearable Computer Laboratory at the University of South Australia. He is currently a NICTA Fellow, CTO A-Rage Pty Ltd, Senior Member of the ACM, and visiting Scholar with the Human Interaction Technology Laboratory, University of Washington. His current research interests include: wearable computers, user interfaces, augmented reality, virtual reality, CSCW, and tabletop display interfaces.