Automatic Alignment System in Simple Planes for Projection Mapping

Tomohiro Kaneda, Yasue Mitsukura and Nozomu Hamada
Graduate School of Science and Technology, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
Phone/FAX: +81-45-566-1718
E-mail: kaneda@mitsu.sd.keio.ac.jp, mitsukura@sd.keio.ac.jp

Abstract
We propose an automatic alignment system in simple planes for projection mapping (PM). PM has been used widely and applied in many fields such as the medical, entertainment, and industrial fields. However, there are many problems such as the equipment of PM being too expensive and large. Additionally, we must align the projected images and the target manually in the final step of the mapping procedure. In this study, to solve these problems, we build a compact mobile PM system with automatic alignment. As reported in this paper, we construct the automatic alignment system for simple planes using an inexpensive camera, laptop computer, and mobile projector as the first step towards the mobile PM system. In experiments, we use a plane with a color frame and a nonframe plane (picture-story show) as the simple planes. Using color feature and local feature amounts such as Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and Oriented FAST and Rotated BRIEF (ORB), we estimate the corners of the plane and project an image to planes automatically. Finally, we evaluate the error calculated by the proposed system.

1. Introduction
In recent years, PM, which is one of the projection technologies using computer graphics, has been applied in many fields. PM is the technology of projecting images onto objects such as walls or architecture, and of varying the image in accordance with the shape of the object. PM is used in many fields. For instance, Sugimoto et al. projected the images of organs onto the body surface to assist surgical operations[1]. In the entertainment field, PM synchronizes the real world and the virtual world to interest people. It is also used for interior arrangements or trying on clothes in simulation. PM is expected to be used in more fields in the future.

However, there are some demerits such as its high cost the need for many kinds of equipment. For example, PM performed at Tokyo Station required 46 high-intensity projectors. As another example, in Tokyo Disneyland, it cost more than 15 million dollars to construct its PM system. Moreover, because the alignment of the image and the projection target is conducted manually, if the objects have complex three-dimensional shapes, the task is very time-consuming. Therefore, the problems of high cost, a large amount of equipment, and time-consuming alignment reduce PM’s versatility.

To solve these problems, the miniaturization of equipment and automatic alignment are desired. Therefore, the final goal of this study is to build a system using an inexpensive camera and a mobile projector that recognizes the projection targets automatically. Such a system is yet to be invented. In this study, as the first step towards the final goal, we try to perform PM using a mobile projector, web camera, and laptop computer. In experiments, we use simple planes as the projection targets. Firstly, their corners are automatically detected. Secondly, we align the image corners to the object corners and project an image. If automatic alignment for planes can be implemented correctly, we will expand the objects of the system into three-dimensional objects.

2. Proposed System and Experiments
We use two types of planes: with a color frame and with no frame. A nonframe plane is a picture-story show. We will explain the alignment system for them separately.

2.1 Simple plane with color frame
To find the corners of the simple plane, we track the color feature using the colorimetric value in the Commission Internationale de l’Éclairage (CIE) L*a*b* color system. The camera image is binarized on the basis of the threshold value for the frame color. Using this binary image, we estimate the corners of the plane by proposed methods 1 and 2 (see Figs. 1 and 2). In method 1, the origin is set at the center of the area of the tracking color and the distance between the frame color pixels and the origin is measured in each quadrant. We assign the pixel with the longest distance as a corner (Fig. 1). However, this method is not applied when the plane is tilted. This is because we confirmed that the true corners may not have the longest distance depending on the quadrants in the experiments. This leads us to propose method 2.
When the plane tilts, the corners are located on maximum or minimum x and y coordinates (Fig. 2). Using this characteristic, the pixels having maximum or minimum coordinates at x or y are defined as corners. Although we are able to track the corners when the plane is tilted, this method is unstable when the plane is set horizontally. This is because many pixels have maximum or minimum coordinates in this situation. Therefore, we combine these two methods to overcome each demerit. We focus on the change in the aspect ratio of the blob when the plane tilts. The blob is a rectangle linked with the maximum x and y and minimum x and y coordinate values. In this paper, the aspect ratio is defined as

\[ a_{\text{blob}} = \frac{B}{L} \]  

where \( a_{\text{blob}} \) denotes the aspect ratio of the blob, \( B \) is the breadth of the blob, and \( L \) is the length of the blob. The blob size and aspect ratio change as shown in Fig. 3 following plane rotation. Therefore, we set the threshold \( t \) according to the plane shape and change the method in accordance with the threshold as follows:

\[ \text{method} = \begin{cases} 
\text{method 1}, & a_{\text{blob}} \leq \frac{1}{t}, t \leq a_{\text{blob}} \\
\text{method 2}, & \frac{1}{t} < a_{\text{blob}} < t 
\end{cases} \]  

(2)

t is the maximum aspect ratio minus 0.1. We build the automatic alignment system for a plane with the color frame by these methods.

2.2 Nonframe plane

For a nonframe plane, we use SIFT, SURF, and ORB to detect the corners and compare their results. At the beginning, we describe them simply. In SIFT, difference-of-Gaussian (DoG) images are used to detect the feature points and to define 128-dimensional vectors as the feature amount[2]. DoG images are difference images between object images convolved with the Gaussian at various scales \( \sigma \). The Gaussian is defined as

\[ G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right) \]  

(3)

In SURF, processes of convolution with the Gaussian in SIFT are substituted for filtering with box filters[3]. Moreover, 64-dimensional vectors are defined as the future amount so that the processing speed is faster than that of SIFT.

ORB uses the feature points detected by FAST, which is one of the fastest corner detection methods[4]. The feature amounts are defined as multidimensional vectors in SIFT and SURF. On the other hand, the binary code is used as the feature in ORB. Figure 4 shows examples of feature extraction using SIFT, SURF, and ORB.

After obtaining the feature amounts, we compare the distances between the target object and the image in each feature point. After this procedure, we detect which feature point in the image corresponds to a point in the target using the Euclidean distance in SIFT and SURF and the Hamming distance in ORB. At the end, we estimate the homography matrix by random sample consensus (RANSAC) based on the correspondences of feature points. Then, we are able to calculate the corner coordinates of a nonframe plane in the images.
2.3 Experiments

Firstly, we rotate the planes by multiples of each 10° from −90° to 90° for rolling and −80° to 80° for yawing and pitching. Secondly, we apply the proposed system to each situation and evaluate the accuracy quantitatively using the errors between estimated corners and true corners. The true corners are measured manually using an image editor. In the evaluation, we convert each error value in pixel units to millimeter units to elucidate the magnitude of the error in a real environment. The computer environment used in this experiment is as follows: an Intel Core i7 (2.80GHz) CPU, 6GB memory, and a Windows 7 OS.

3. Results and Discussion

3.1 Plane with color frame

Figure 5 shows the errors in rolling, yawing, and pitching when we applied the proposed system to a plane with a frame. In these figures, bar charts indicate the mean error of four corners, and blue curves indicate the area of rectangles formed by joining the true corners. As shown in Fig. 5(a), the proposed system could trace the target corners in the case of rolling.

In the case of yawing, our system was able to detect corners at −10° ≤ β ≤ 10°. On the other hand, at ±20°, ±30°, and ±40° the error became large (Fig. 5(b)). This is because method 1, chosen on the basis of the aspect ratio, was unsuitable for this situation. At these angles, because many pixels were located on nearby maximum or minimum x and y coordinates, method 1 could not detect the corners. At ±50°, method 2 was chosen on the basis of the aspect ratio and found to be suitable. Although method 2 was even used at −80° ≤ β ≤ −60° and 60° ≤ β ≤ 80°, the right (or left) frame was hidden and tracking failed. Therefore, the corners were detected incorrectly. However, when the plane was tilted more and the breadths of the rectangle became shorter, the errors decreased.

In the case of pitching, method 1, chosen on the basis of the aspect ratio, could be applied to −30° ≤ γ ≤ 30° and detected the corners correctly. At −80° ≤ γ ≤ −40° and 40° ≤ γ ≤ 80°, although the method was not changed, the errors became large. This is because the upper (or lower) frame was hidden and not recognized. As in yawing, when the plane was tilted and the lengths of the rectangle became shorter, the errors decreased.

Figure 6 shows the result of PM using this system in the case of rolling. Regardless of whether the plane was tilted or not, the system estimated the corners correctly and projected the image with the adjustment of these corners.

3.2 Nonframe plane

Table 1: Processing speeds

|        | SIFT | SURF | ORB |
|--------|------|------|-----|
| Rolling [s] | 1.01 | 0.461 | 0.071 |
| Yawing [s]  | 1.01 | 0.456 | 0.114 |
| Pitching [s]| 1.02 | 0.452 | 0.109 |

Table 1 shows the processing speed using each feature amount in each type of rotation. Figure 7(a) shows that every method could detect corners, and SIFT yielded almost perfectly correct results in rolling. On the other hand, in terms of the processing speed, SIFT and ORB were the slowest and the fastest, respectively.

In the case of yawing, at −60° ≤ β ≤ 60°, each method could detect the corners sufficiently accurately (Fig. 7). However, the errors were large at −80° ≤ β ≤ −70° and 70° ≤ β ≤ 80°. Because the feature points of the characters on the right or left (Fig. 4) was hidden at these angles, matching between the target object and the image was not implemented correctly.

In the case of pitching, although the errors were small at −60° ≤ γ ≤ 60°, the corners were detected incorrectly at −80° ≤ γ ≤ −60° and 60° ≤ γ ≤ 80°. This is because feature points were hidden. Moreover, errors were not symmetric at positive and negative angles because the number of feature points on the upper side was larger than that on the lower side.

4. Conclusions

The final goal of our study is to construct a mobile system for PM. In this work, as the first step towards the final goal, we built an alignment system for PM using mobile equipment and applied it to simple planes with the three types of rotation to verify its effectiveness. As a result, we were always able to detect the corners of planes in case of rolling and part of the time in the case of yawing and pitching. However, we encountered some problems. Firstly, the method chosen on the basis of the threshold we set was unsuitable for some situations. Secondly, matching was not able to be performed because the feature points were hidden by some rotations. In future works, for a plane with a color frame, we will reset the threshold of the aspect ratio or use other indicators to choose between method 1 and method 2. Also, for a nonframe plane, we must reconsider the method. If automatic alignment for planes can be implemented correctly, we can expand the objects of the system to three-dimensional objects.

Acknowledgment

This work was supported by JSPS KAKENHI (C) Grant Number 26330230.
Figure 6: PM result

(a) System outline  
(b) Successful example  
(c) Example of failure

Figure 5: Errors in a plane with the color frame

(a) Rolling error  
(b) Yawing error  
(c) Pitching error

Figure 7: Errors in a nonframe plane

(a) Rolling error  
(b) Yawing error  
(c) Pitching error

References

[1] M. Sugimoto et al.: Image overlay navigation by markerless surface registration in gastrointestinal, hepatobiliary and pancreatic surgery, Journal of Hepato-Biliary-Pancreatic Sciences, Vol. 17, No. 5, pp. 629-636, 2010.

[2] D.G. Lowe: Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.

[3] H. Bay, T. Tuytelaars and L.V. Gool: Surf: Speeded up robust features, Computer Vision ECCV 2006, Springer, pp. 404-417, 2006.

[4] E. Rublee et al.: ORB: An efficient alternative to SIFT or SURF, Computer Vision ICCV, 2011 IEEE International Conference, pp. 2564-2571, 2011.