An Interactive Image Rectification Method Using Quadrangle Hypothesis

Satoshi Yonemoto

Graduate School of Information Science, Kyushu Sangyo University
Yonemoto@is.kyusan-u.ac.jp

Abstract. In this paper, we propose an interactive image rectification method for general planar objects. Our method has two interactive techniques that allow a user to choose the target region of interest. First, with a user-stroke based cropping. Second, with a box based cropping. Our method can be applied to non-rectangular objects. The idea is based on use of horizontal and vertical lines with the target object. We assume that such lines can be richly detected. Practically, at least two horizontal lines and two vertical lines must be observed. Our method has the following procedures: First, detect primitive line segments, and then select horizontal and vertical line segments using baselines. Next, make a quadrangle hypothesis as a combination of 4 line segments. And then, evaluate whether re-projected line segments will be horizontal (vertical) or not. The quadrangle hypothesis with max goodness is the final solution. In our experiments, we showed promising cropping results for several images. And we demonstrated real-time marker-less tracking using the rectified reference image.

Keywords: image rectification, marker-less tracking, text detection.

1 Introduction

Localizing text area in video frames has been researched for various tasks such as information retrieval and license plate identification [4][6]. As text detection methods in natural scene, two main approaches are taken. The first is vanishing point estimation [2] [9]. This approach is often used for road plane rectification. It is assumed that reliable long edges can be stably observed. However, with imprecise short edges, calculation error may give a serious problem. The second is quadrangle estimation [7]. A quadrangle is warped into a rectified text area (i.e., a rectangle), correcting any perspective distortion. In document images used in OCR, text lines are regularly aligned in paragraph, so rich horizontal lines are stably extracted. Two text lines and the vertical stroke boundaries are often used as a quadrangle clue [2].

Our method includes text detection in natural scene, and is extended to general planar objects. As mentioned above, many text detection methods have already been proposed. However, since their methods are based on estimation of foreground and background color pixels, they are not applied to general planar objects. We assume that target image has rich horizontal and vertical lines. Hence, our method is not applied to edge-less objects and slanted texts such as italic fonts.

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 51–60, 2013.

In recent augmented reality (AR) application, robust tracking algorithm is used, which does not use any artificial markers. It is known as marker-less tracking [3]. Reference image generation in AR is very similar to text detection in natural scene. Most marker-less tracking approaches force user to capture the front view of a target object. Most planar objects in natural scenes are observed in non-frontal view. In offline video use, reference image must be acquired from one of the video frames. Also, in online use, user cannot always capture a front view image (e.g., large sign-board and building surface). Therefore, we propose an image rectification method for non-frontal view of a planar object. In case of a rectangular object such as book and poster, their corner points are clearly observed, so the reference image is easily acquired from their points [10]. Our method can also be applied to non-rectangular objects such as circular objects. To crop such objects is not discussed so far. The idea is use of horizontal (vertical) lines with target object, that is, these lines always become horizontal, correcting any perspective distortion. We assume that at least two horizontal lines and two vertical lines would be observed. Our method is one of quadrangle estimation methods. A quadrangle with a target object is constructed by picking up 4 points (or 4 lines) in an image. The projection of a rectangle is a quadrangle. Our approach to find such quadrangle is summarized as follows:

- Specify the target object of interest in an interactive way.
- Extract horizontal and vertical line segments in the target image.
- Make a quadrangle hypothesis from their line segments. Next, warp the quadrangle into a rectangle [5]. Then, evaluate whether re-projected line segments will be horizontal (vertical) or not. The goodness function is defined by total error with the transformed horizontal and vertical lines. The important point is whether detected lines will be transformed horizontal (vertical) or not.

As a result, the underlying problem is equal to a search problem for a combination of 4 line segments.

2 Interactive Image Rectification

We have proposed an interactive image rectification method that has two interactive techniques. The first is a user-stroke based cropping that can specify the horizontal baseline. The second is a box-based cropping that specifies the whole target region of interest. Fig. 1 illustrates the user-stroke based cropping.

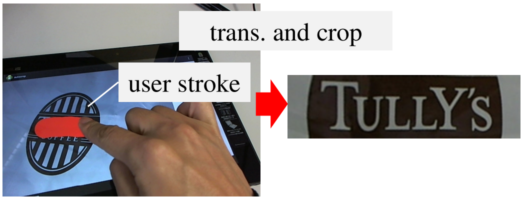

Fig. 1. User-stroke based cropping

2.1 User-Stroke Based Cropping

User can directly paint a stroke(s) on the target image. User can also select pen size in advance. We also support multiple strokes. In this cropping, it is assumed that user stroke direction is parallel to a horizontal baseline. We call a horizontal baseline Hb, and we call a vertical baseline Vb. Fig. 2(1a)(2a) shows input images (TULLY and FIGURE) in which user stroke is drawn. Two user strokes are painted in Fig. 2(2a). The calculated bounding box and the baselines are drawn in Fig. 2(1b)(2b).

2.2 Box Based Cropping

User can specify a bounding box for the target object. In this case, user does not define the baselines clearly. To find valid baseline Hb and Vb, we take a hypothesis and test approach. Fig. 3(1a)(2a) shows input images (HOUSE and P6T) in which the user-specified bounding box is drawn. Reference baselines are drawn in Fig. 3(1b)(2b). Reference baselines are candidates for the baseline Hb.

2.3 Line Segments

Examples of the extracted line segments in the bounding box are shown in Fig. 2(1c)(2c) and Fig. 3(1b)(2b). $\{hl_m\}$ and $\{vl_n\}$ are horizontal and vertical line segments respectively. Short red lines indicate the horizontal lines parallel to Hb (long red line). Short blue lines indicate the vertical lines roughly parallel to Vb (long blue line).

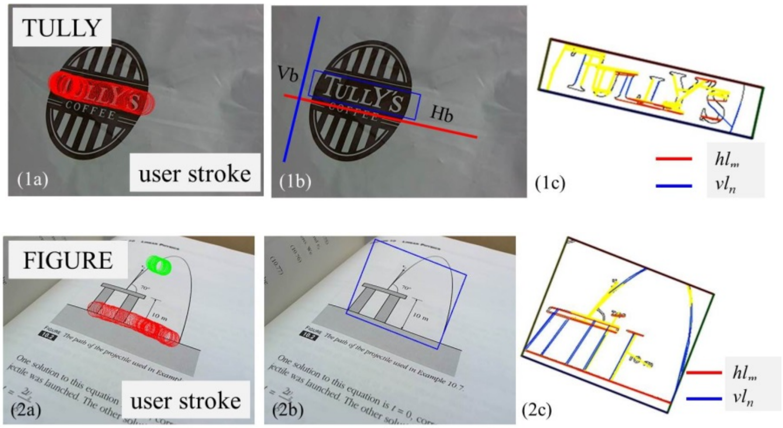

Fig. 2. User stroke and extracted line segments. (1a)(2a) input image (TULLY and FIGURE) and user stroke (1b)(2b) bounding box and baselines (1c)(2c) extracted line segments.

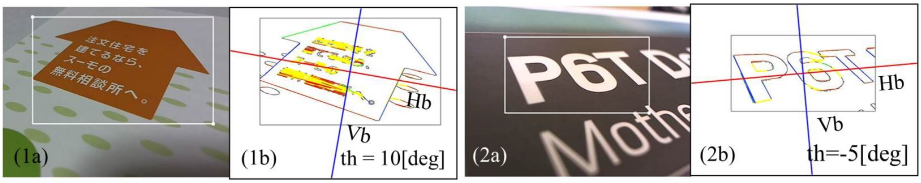

Fig. 3. Box and extracted line segments. (1a)(2a) input image (HOUSE and P6T) and bounding box (1b)(2b) reference baselines and extracted line segments.

3 Line Segments Extraction

Baseline Definition

To detect edges, Canny edge detector is used [1]. The primitive line segments are extracted by probabilistic Hough transform [8]. To select horizontal and vertical line segments from them, baseline Hb and Vb are used. The task is to select line segments parallel to each baseline. In user-stroke based method, baselines are automatically detected from the bounding box of the user stroke. In box based method, reference baseline has to be generated as a hypothesis. Angle *th* with reference baseline is defined, and baseline Hb is used as a basis. Vertical baseline Vb lies at right angles to Hb. Examples of the reference baseline are shown in Fig. 3(1b)(2b). In the examples, angle *th* is set to 10 [deg] and -5 [deg] respectively. Although the reference baseline is not true horizontal line, it is used to “collect” the horizontal line segments.

3.1 Horizontal and Vertical line Segments

Next, line segments parallel to baseline Hb or Vb are selected from the primitive line segments. To avoid similar line confliction, overlapped line segments are rejected. In such a case, shorter line segments can be rejected. In Fig. 2(1c)(2c) and Fig. 3(1b)(2b), the rejected line segments are drawn as yellow or green lines. Vertical line segments may be not parallel to baseline Vb. Therefore candidates for the vertical line segments are widely collected.

4 Image Rectification Algorithm

The estimation task is to determine a quadrangle, namely, 4 corner points set in the target image. If a quadrangle is the projection of a rectangular object, the quadrangle will be geometrically transformed into a rectangle by projection geometry. Our estimation algorithm consists of the following steps:

- [**Step 1**] Sort line segments. Fig. 4 illustrates our idea to avoid generating unnecessary hypothesis. First, sort horizontal line segments $\{hl_m\}$. They are sorted by a sort key y_m. y_m is y-coordinate of a center point of a line segment hl_m. The coordinate system is defined, where baseline Hb and Vb are used as a basis. Next, sort vertical line segments $\{vl_n\}$. They are sorted by a sort key x_n. x_n is x-coordinate of a center point of a line segment vl_n. Data structures of the sorted line segments are represented as the following buffers:

hlbuf [M] ... sorted $\{hl_m\}$ m=1,...,M *vlbuf* [N] ... sorted $\{vl_n\}$ n=1,...,N

- [**Step 2**] Set angle *th* with reference baseline. The search parameter *th* is defined in the limited range ([-45, +45]). The parameter *th* is updated by step size (+5 [*deg*]).
- [**Step 3**] Select 4 line segments {hu, hb, vl, vr}. Run the following procedures: Select hu and hb from *hlbuf,* and select vl and vr from *vlbuf.* The line segment hu is an upper line segment in *hlbuf* [i] (i.e., hu is i$^{\text{th}}$ element). hb is a bottom line segment in *hlbuf* [M-j] (hb is M-j$^{\text{th}}$ element). vl is a left line segment in *vlbuf* [k]. vr is a right line segment in *vlbuf* [N-l]. The search range for i, j, k, l are limited by constant value N_h and N_v. In short, 4-tuple {hu, hb, vl, vr} is defined as follows:

  hu = *hlbuf* [i], hb = *hlbuf* [M-j], vl = *vlbuf* [k] and vr = *vlbuf* [N-l].

- [**Step 4**] Next, generate a quadrangle hypothesis q_{th}(i, j, k, l) from 4-tuple {hu, hb, vl, vr} (see Fig. 5). The quadrangle hypothesis can be generated by intersecting each extended lines. Their lines are obtained by extending the line segments. The following steps are executed:

  4-1. Reject illegal combination of {hu, hb, vl, vr}, keeping geometrical constraints.
  4-2. Warp the quadrangle into a rectangle, correcting any perspective distortion.
  4-3. Verify the hypothesis q_{th}(i, j, k, l), evaluating 'goodness' (see Fig. 6).
  4-4. Update max goodness.

- [**Step 5**] for all *th* and {i, j, k, l}, find the best quadrangle q_{th}(i, j, k, l) with max goodness. Note that in user-stroke based method, the parameter *th* is not estimated. The best quadrangle is the final solution. Finally, geometrical image transformation is applied to the best quadrangle. Then target image is automatically cropped using the bounding box information.

An example of 4-tuple {hu, hb, vl, vr} is drawn as thick lines in Fig. 5. The quadrangle hypothesis is overlaid as a black polygon with cross lines. Note that any corner points of the quadrangle are not directly estimated from the image. The goodness function is defined by total error with the transformed line segments:

$$goodness = \frac{M + N}{M + N + \sum_{i}^{M+N} \| d_1^i - d_2^i \|} \tag{1}$$

where, d1 and d2 are terminal points of a line segment (see Fig. 6). The position error ||d1-d2|| is calculated and then summed for all line segments. Total number of line segments is M+N. In the horizontal line segments, ||y1-y2|| is used, and in the vertical case, ||x1-x2|| is used.

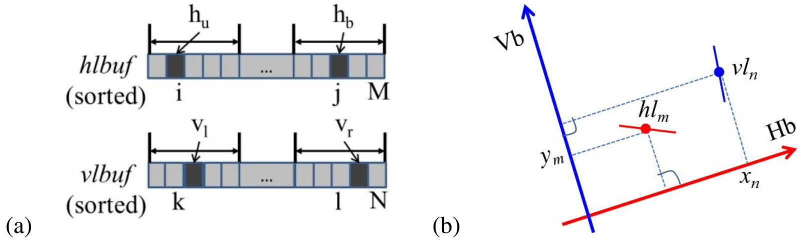

Fig. 4. Data structure of line segments. (a) Two buffers (*hlbuf*, *vlbuf)*. (b) Sort key calculation.

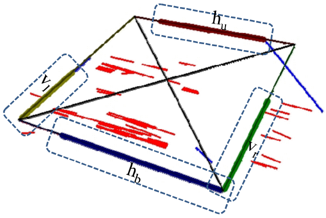

Fig. 5. An example of 4-tuple {hu, hb, vl, vr} with max goodness (HOUSE)

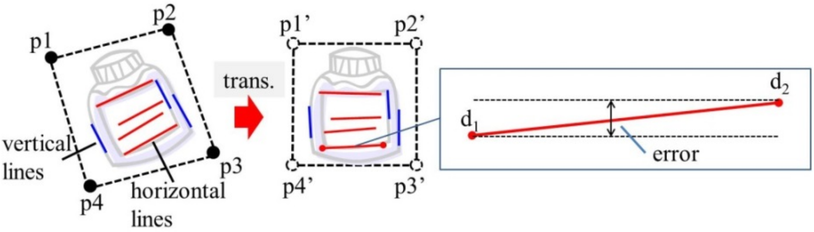

Fig. 6. Geometrical transformation and line segment evaluation

5 Experiments

5.1 Implementation Details

We have implemented our prototype tool on a laptop PC (OS: windows 7, CPU: Intel Core i5, memory: 4GB). The experimental input images were captured using a web camera (*Logicool* C910). We also developed the tiny tool run on a tablet PC.

5.2 User-Stroke Based Cropping

Fig. 7 shows the estimation results for test images TULLY and FIGURE. The user stroke was shown in Fig. 2. In the left figure, the best quadrangle hypothesis {i,j,k,l}={0,2,0,0} is shown. In the right figure, the warped fronto-parallel image is shown. Edges for text ‘T’,’L’ and ’Y’ are automatically selected as vl, hb and hu. Note that the vertical lines in the stripe pattern are not used in this estimation process. In the bottom figure, the best quadrangle hypothesis {i,j,k,l}={3,1,4,3} is shown. The figure edges are selected as vl, hb and hu. vr is the straight part of the curve. Using multiple user strokes, our method enables auto-cropping in the wider range.

5.3 Box Based Cropping

Fig. 8 shows the estimation results for test images HOUSE and P6T. The best quadrangle hypothesis (*th*=10[deg], {i,j,k,l}={0,0,0,1}) is shown. Contour edges are selected as hu, vr, hb and vl. In P6T, the best quadrangle hypothesis (*th*=-5[deg], {i,j,k,l}={5,1,0,2}) is shown. Edges for text ‘P’,’T’ are selected as vl, hb and vr. The estimation results for test images INTEL and TULLY are shown in Fig. 9. TULLY is the same image used in Fig. 2. In this case, the whole region of logo is the target region of interest. The best quadrangle hypothesis and the extracted line segments are shown in the right figure.

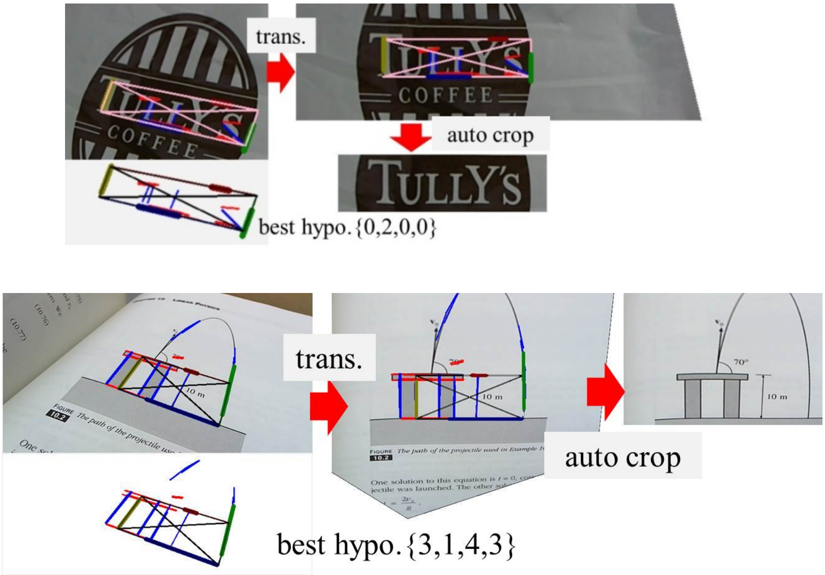

Fig. 7. User-stroke based cropping result (TULLY and FIGURE)

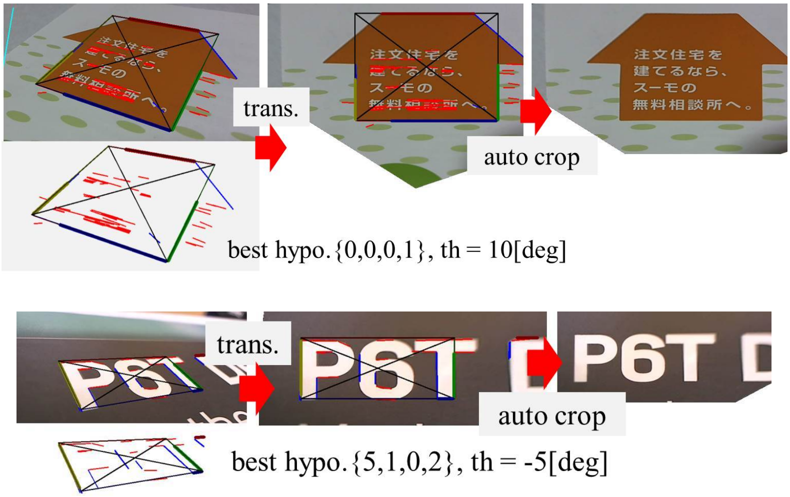

Fig. 8. Box based cropping result (HOUSE and P6T)

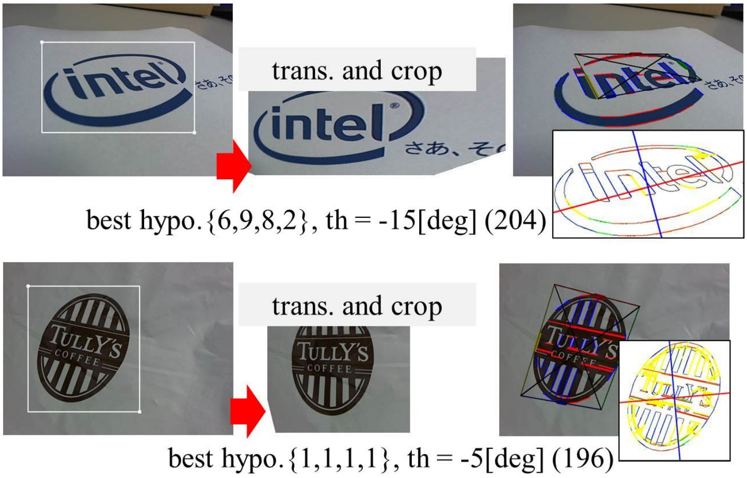
intel
trans. and crop
best hypo.{6,9,8,2}, th = -15[deg] (204)
TULLY'S COFFEE
trans. and crop
best hypo.{1,1,1,1}, th = -5[deg] (196)

Fig. 9. Box based cropping results (INTEL and TULLY)

5.4 Experimental Evaluation

Fig. 10 shows several candidates for reference baselines with different angle *th*. In this example, the best quadrangle hypothesis is estimated at *th* = -5[deg] (see Fig. 8). Since line is distorted by perspective projection, baseline Hb is not perpendicular to Vb. However, the task is to collect the horizontal line segments. Therefore, we consider use of reference baselines valid. Fig. 11 shows three quadrangle hypotheses (goodness scores are 15, 65 and 327 respectively). Principally, many combinations can be taken as alternative solutions. Max goodness score is 327 point. In this image, slanting line segments of text ‘A’ are selected. Our method may be not suitable for such a case. However, the goodness score for quadrangle hypothesis that includes slanting line segments is obviously lower than the score for the best hypothesis. Fig. 12 shows an example of a combination with a text line detector [4] [7]. The bounding box calculated from user-stroke is drawn. Two text lines (i.e., hu and hb) are detected by the detector. In this case, only vertical line segments (vl and vr) are extracted. Edges for text ‘H’ and ’N’ are automatically selected as vl and vr.

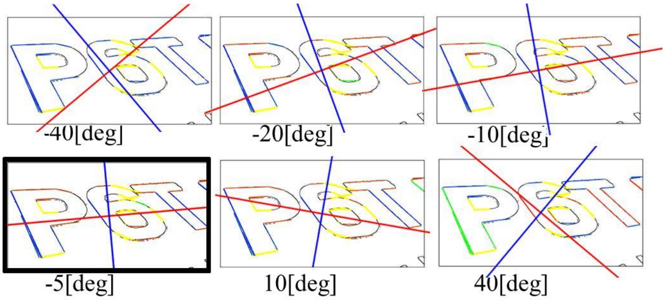

Fig. 10. Candidates for reference baselines (P6T)

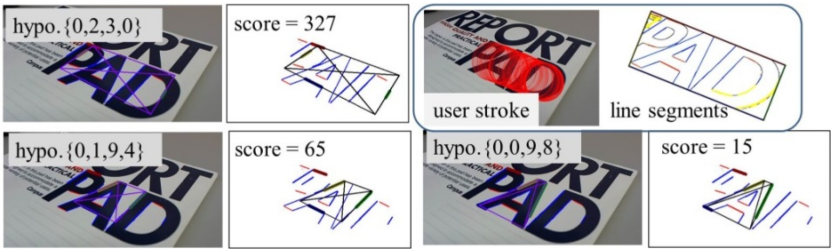

Fig. 11. Three quadrangle hypotheses (PAD). Goodness scores are described

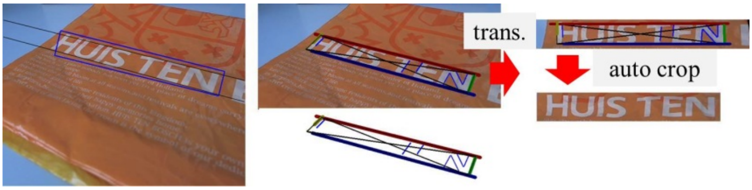

Fig. 12. Combination with text line detector. (left) two detected lines and bounding box. (right) cropping results.

5.5 Application

Fig. 13 shows several shots of marker-less AR tracking, where the rectified reference image TULLY (Fig. 9) is used and virtual graffiti is superposed on the live video. Tracked quadrangle is also projected.

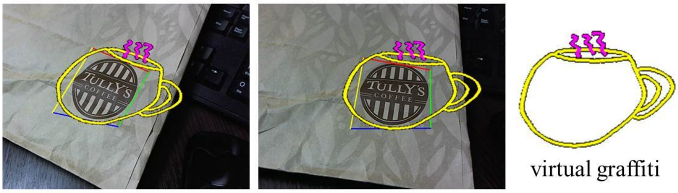

Fig. 13. Marker-less AR application. Virtual graffiti is supuerposed on live video frames.

6 Conclusions

We have proposed an interactive image rectification method that can be applied to non-rectangular objects. In user-stroke based cropping, user can specify the horizontal baseline. While, in box based cropping, user specifies the whole target region of interest. The idea is based on use of horizontal and vertical lines with target object. This estimation problem is solved as a search problem with the best 4-tuple combination of line segments, evaluating whether re-projected line segments will be horizontal (vertical) or not. In our experiments, we showed promising cropping results. And we demonstrated real-time marker-less tracking using the rectified reference image.

Future work includes performance evaluation for various target objects and full implementation for mobile devices such as tablet PC and smart phones.

References

1. Canny, J.A.: computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) 8, 679–714 (1986)
2. Clark, P., Mirmehdi, M.: Estimating the orientation and recovery of text planes in a single image. In: Proceedings of the 12th British Machine Vision Conference (2001)
3. Fue, P., Vincent, L.: Vision based 3D tracking and pose estimation for mixed reality. In: Emerging Technologies of Augmented Reality: Interfaces and Design, pp. 1–22 (2005)
4. Fragoso, V., et al.: TranslatAR: A Mobile Augmented Reality Translator. In: Proceedings of IEEE Workshop on Applications of Computer Vision (WACV), pp. 497–502 (2011)
5. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, New York (2001)
6. Jain, A., Yu, B.: Automatic text location in images and video frames. In: Proc. 14th Intl. Conf. on Pattern Recognition, vol. 2, pp. 1497–1499 (1998)
7. Lu, S., et al.: Perspective rectification of document images using fuzzy set and morphological operations. Image and Vision Computing 23(5), 541–553 (2005)
8. Matas, J., Galambos, C., Kittler, J.V.: Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. CVIU 78(1), 119–137 (2000)
9. Yin, X.-C., et al.: Robust Vanishing Point Detection for MobileCam-Based Documents. In: International Conference on Document Analysis and Recognition. IEEE (2011)
10. Zhang, Z., He, L.-W.: Note-taking with a camera: whiteboard scanning and image enhancement. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 3 (2004)