Machine Vision and Applications
DOI 10.1007/s00138-016-0807-1

ORIGINAL PAPER

# Rectification of planar targets using line segments

Jaehyun An[1] · Hyung Il Koo[2] · Nam Ik Cho[1]

Received: 7 May 2015 / Revised: 13 March 2016 / Accepted: 29 August 2016

**Abstract** This paper presents a method that performs the rectification of planar objects. Based on the 2D Manhattan world assumption (i.e., the majority of line segments are aligned with principal axes), we develop a cost function whose minimization yields a rectification transform. We parameterize the homography with camera parameters and design a cost function which encodes the measure of line segment alignment. Since there are outliers in the line segment detection, we also develop an iterative optimization scheme for the robust estimation. Experimental results on a range of images with planar objects show that our method performs rectification robustly and accurately.

**Keywords** Rectification · Planar target · Line segment

## 1 Introduction

Since planar surfaces are common and important targets in many computer vision applications, robust and efficient rectification of surfaces has been a widely researched topic. Some examples where the 2D rectification algorithm plays a crucial role to the overall performance are object tracking, optical character recognition (OCR), and augmented reality [1–3,14,19,23]. In these applications, many of the algorithms exploited application-specific features, and hence one cannot be straightforwardly applied to other applications. Also, some of them are based on sophisticated segmentation or optimization schemes, which cannot be used in time-critical applications. In this paper, in order to alleviate these limitations, we present an efficient and effective method that can be applied to various kinds of planar objects.

### 1.1 Rectification of planar objects

The conditions for metric rectification of planar targets have been extensively studied in the literature [8,15]. However, automatic rectification of planar targets is still a challenging problem. Specifically, among a variety of conditions for the metric rectification (e.g., known rectangles, angles, and parallel curves), it is not easy to find the right exact conditions for the given images and to select appropriate algorithm: Some images have dominant rectangles, others have parallel curves, and so on. Therefore, rather than developing a general rectification algorithm, many researchers focused on individual cases. Probably, the most widely studied case is the rectification of rectangle target, which is a common and useful target in many images. For example, Hua and Liu [9] developed a rectification method for rectangle targets: They first segmented business card image patches by applying a segmentation method [10], and the segmented region is fitted to a quadrangle. Under the similar assumptions, four boundaries of rectangles were detected with Hough transform in [7].

Also, there are rectification algorithms for other conditions such as the existence of coplanar repeated patterns and parallel planar curves [4,20]. They showed interesting theoretical results; however, these application areas are somewhat limited because such patterns are not commonly found in natural images. Some researchers tried to circumvent this problem by using additional hardware or user interactions: Lee et al. [14] addressed the rectification problem with the help of inertial sensors and others simply assumed that the users provide a fronto-parallel view at the first frame [23].

✉ Hyung Il Koo
hikoo@ajou.ac.kr

[1] INMC, Seoul National University, Seoul, Korea

[2] Ajou University, Suwon, Korea

Published online: 10 September 2016

### 1.2 Rectification based on self-calibration

The rectification problem is also addressed in a different way: Self-calibration algorithms [6,12,13,16,21,22,25–27] yield the pose of the camera and the rectification transform of the planar targets can be computed from the calibration results. For instance, Zhang et al. [26,27] developed a camera calibration method based on the properties of low-rank textures, which can remove radial distortions as well as perspective distortions in the images. However, this method has some limitations in that it works only for low-rank textures, it requires user interaction, and its computation cost is high. Recently, self-calibration of a camera in a (3D) Manhattan world received a lot of attention [6,12,13,16,21,22,24,25], which can also be used for the 2D metric rectification. Actually, one of these methods provides some successful results on planar targets [13]. However, these algorithms were developed for the 3D world, and it may be sub-optimal for the 2D rectification problem. In Experimental section, we will compare our method with the state-of-the-art 3D self-calibration methods in terms of accuracy and computation cost for the 2D problems.

### 1.3 Our approach

Our approach is based on the 2D Manhattan world assumption, where a profusion of lines is aligned either horizontally or vertically on the plane. Based on this assumption, we develop a cost function that evaluates the alignments of lines and find the rectification transform by minimizing the cost function. Since our method is based on (a kind of) the Manhattan world assumption and the cost function is formulated with camera parameters, it can be considered a self-calibration method. Main difference from the conventional 3D calibration methods is that the proposed algorithm is focused on the alignment of lines [instead of the estimation of vanishing points (VPs)]. In particular, consistency measures between VPs and extracted edges were used in [21,24] to estimate VPs and these methods can perform rectification with the estimated VPs. However, the proposed method can be simpler than this two-step approach because our method does not need the sophisticated VP estimation step. Also, the proposed method provides improved rectification results as shown in Experimental section. It is because the proposed line alignment metric is designed to reflect the rectification quality.

The most similar approach to our work may be the orthogonality based method in [25], which finds the rectification transform by using the pairs of orthogonal line segments. However, our cost function is based on line alignments rather than pairwise orthogonality, and our cost function can be minimized efficiently. Experimental results on a range of planar objects (including building facades, documents, and signposts) show that the proposed method not only is efficient but also yields robust rectification performance compared with the conventional methods [13,25, 26].

## 2 Proposed rectification model

We solve the 2D rectification problem by estimating the internal and external parameters of a camera. For this, we first extract line segments in the image and find the homography that makes the majority of these line segments aligned with principal axes. In this section, we first present our parametric model for the homography (i.e., the rectification transform).

### 2.1 Homography between the reference plane and the camera-captured image

We assume that the image is captured with a standard pinhole camera and consider the pose of camera as an element in the special Euclidean group $SE(3)$. To be specific, let us denote the internal matrix of a camera and its pose (external parameters) as

$$\mathrm{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

and

$$\begin{bmatrix} \mathrm{R} & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in SE(3), \tag{2}$$

respectively, where $f$ is the focal length, $R \in \mathrm{SO}(3)$ is the rotation matrix, and $\mathbf{t} \in \Re^{\mathbf{3}}$ is a translation vector. Then, the camera matrix of the image is given by

$$P = \mathrm{KR}\left[\mathrm{I}_{3\times 3}| - \mathbf{t}\right], \tag{3}$$

and the relationship between a point $[X, Y, 0]^\top$ on the surface and its corresponding point on the image $p = [x, y]^\top$ is given by

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathrm{KR}\left[\mathrm{I}_{3\times 3}| - \mathbf{t}\right] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{4}$$

$$= \mathrm{K}\left[\mathbf{Re}_1 \ \mathbf{Re}_2 \ -\mathrm{R}\mathbf{t}\right] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{5}$$

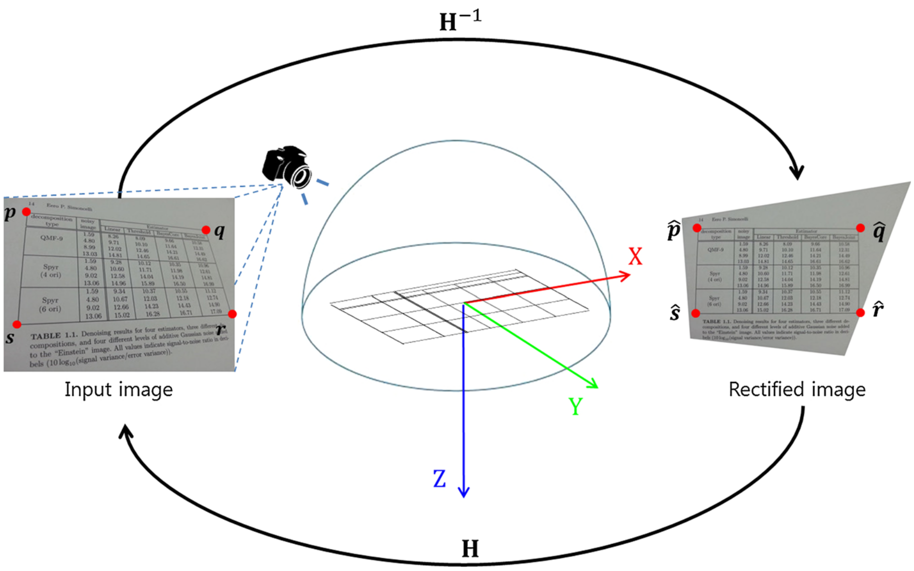

**Fig. 1** When the majority of line segments are aligned with principal axes, we can rectify the camera-captured images by finding a transform that makes the line segments horizontally or vertically aligned

for some $\lambda$, where $e_i$ $(i = 1, 2, 3)$ are unit vectors representing each axis [8]. Thus, the homography H between the image and the reference plane is given by

$$\mathrm{H} = \mathrm{KR}\left[\mathbf{e}_1\ \mathbf{e}_2\ -\mathbf{t}\right] \tag{6}$$

$$= \mathrm{KR}\left\{\mathrm{I} - (\mathbf{t} + \mathbf{e}_3)\,\mathbf{e}_3^{\top}\right\}. \tag{7}$$

By applying the matrix inversion lemma to (7), its inverse transform (the rectification transform) is given by

$$\mathrm{H}^{-1} = \left\{\mathrm{I} - \frac{(\mathbf{t} + \mathbf{e}_3)\,\mathbf{e}_3^{\top}}{\mathbf{e}_3^{\top}\mathbf{t}}\right\}\mathrm{R}^{\top}\mathrm{K}^{-1}, \tag{8}$$

which is the rectification matrix as illustrated in Fig. 1.

### 2.2 Parametrization

For the rotation matrix, we adopt an exponential representation:

$$\mathrm{R} = \exp\left([\boldsymbol{\theta}]_{\times}\right), \tag{9}$$

for a vector $\boldsymbol{\theta} \in \Re^3$. Also, we set the translation vector $t$ to

$$[0, 0, -\max(w, h)]^{\top} \tag{10}$$

without loss of generality, where $w$ and $h$ are the width and height of the input image, respectively. Then, the metric rectification is equivalent to estimating $f$ and $\boldsymbol{\theta} \in \Re^3$ from the detected line segments.

## 3 Proposed method

In this section, we derive a cost function based on the alignments of line segments to principal axes and develop its optimization method.

### 3.1 Optimization-based framework

Similar to other methods [13,25], we extract line segments using the LSD (line segment detector) method in [5]. Let us denote detected line segments as

$$\mathcal{L} = \{(\mathbf{u}, \mathbf{v})\}, \tag{11}$$

where $u$ and $v$ are the homogeneous representation of two endpoints of a line segment. Then, we formulate the rectification as an optimization problem defined on $\mathcal{L}$:

$$(\hat{\boldsymbol{\theta}}, \hat{f}) = \underset{\boldsymbol{\theta}, f}{\operatorname{argmin}}\, E(\boldsymbol{\theta}, f; \mathcal{L}) + \lambda F(\boldsymbol{\theta}, f), \tag{12}$$

where the first term $E(\boldsymbol{\theta}, f; \mathcal{L})$ evaluates alignments to the principal axes when we apply $\mathrm{H}^{-1}$ to $\mathcal{L}$ (Fig. 2), the second term $F(\boldsymbol{\theta}, f)$ helps us to avoid trivial solutions by imposing the constraints on the focal length, and $\lambda$ is a factor to control the balance between the two terms.

### 3.2 Cost function based on line segment alignments

We define the first term in (12) as

$$E(\boldsymbol{\theta}, f; \mathcal{L}) = \sum_{(\mathbf{u}, \mathbf{v}) \in \mathcal{L}} w(\mathbf{u}, \mathbf{v}) \times d_{\mu}^{2}(\mathrm{H}^{-1}\mathbf{u}, \mathrm{H}^{-1}\mathbf{v}), \tag{13}$$

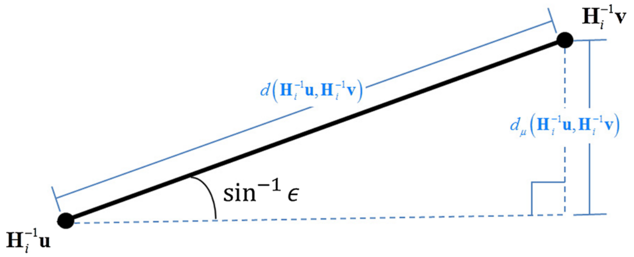

**Fig. 2** A line segment with large angle, i.e., $\sin^{-1}\epsilon(\mathrm{H}^{-1}, \mathbf{u}, \mathbf{v}) \gg 0$, is considered an outlier

where $\mathrm{H}^{-1}$ is given by (8) and $w(\cdot, \cdot)$ is the normalized weight (that sums to unity) of each segment. Since longer lines are more informative for estimating the rectification transform, the weights are defined as:

$$w(\mathbf{p}, \mathbf{q}) \propto d^2(\mathbf{p}, \mathbf{q}), \tag{14}$$

where $d(\cdot, \cdot)$ is a geometric distance between two homogeneous points ($\mathbf{p} = [p_1, p_2, p_3]^\top$ and $\mathbf{q} = [q_1, q_2, q_3]^\top$):

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1/p_3 - q_1/q_3)^2 + (p_2/p_3 - q_2/q_3)^2}. \tag{15}$$

In order to evaluate their alignments, we also define the alignment error as

$$d_\mu(\mathbf{p}, \mathbf{q}) = \min(|p_1/p_3 - q_1/q_3|, |p_2/p_3 - q_2/q_3|). \tag{16}$$

Even though (13) successfully measures the alignments of lines, its direct minimization may result in a trivial solution (by making the focal length $f$ very large). Hence, we impose a constraint that the focal length $f$ should be similar to the image scale [13], i.e., the second term is designed as

$$F(\boldsymbol{\theta}, f) = \left(\frac{\max(a, f)}{\min(a, f)} - 1\right)^2, \tag{17}$$

where $a = \max(w, h)$ similar to the method in [13]. However, $F(\boldsymbol{\theta}, f)$ is a regularization term and experimental results show that the choice of this term is not critical (e.g., $(f - a)^2$ yields similar results).

### 3.3 Optimization

When the set of LSD results (11) does not contain outliers, the cost function (12) can be easily minimized via the Levenberg–Marquardt algorithm, because it consists of sum of square terms [18] and its Jacobian can be computed analytically (detailed derivations can be found in Appendix). However, as shown in Fig. 3, LSD results usually contain lots of outliers, and hence we develop an iterative approach that incrementally filters out these outliers.

To be precise, for a current solution $\mathrm{H}_i^{-1}$, we collect a set of (potential) inliers for the $(i+1)$-th iteration, by finding the lines that have small $d_\mu(\mathrm{H}_i^{-1}\mathbf{u}, \mathrm{H}_i^{-1}\mathbf{v})$ compared to $d(\mathrm{H}_i^{-1}\mathbf{u}, \mathrm{H}_i^{-1}\mathbf{v})$ (refer to Fig. 2) as

$$\mathcal{L}_{i+1} = \left\{(\mathbf{u}, \mathbf{v}) \in \mathcal{L} \mid \epsilon(\mathrm{H}_i^{-1}, \mathbf{u}, \mathbf{v}) < \tau_i\right\}, \tag{18}$$

where

$$\epsilon(\mathrm{H}_i^{-1}, \mathbf{u}, \mathbf{v}) = \frac{d_\mu(\mathrm{H}_i^{-1}\mathbf{u}, \mathrm{H}_i^{-1}\mathbf{v})}{d(\mathrm{H}_i^{-1}\mathbf{u}, \mathrm{H}_i^{-1}\mathbf{v})}. \tag{19}$$

From $\mathcal{L}_{i+1}$, we can estimate $\mathrm{H}_{i+1}^{-1}$ by minimizing (12) defined on $\mathcal{L}_{i+1}$, with the Levenberg–Marquardt algorithm [18].

We set $\mathcal{L}_1 = \mathcal{L}$, and $\tau_i$ in (18) is selected so that only the minority of line segments are classified into outliers at each iteration:

$$\tau_i = \max\left(\sin\left(\frac{\pi}{60}\right), \min\left(\mu_i + k \times \sigma_i, \sin\left(\frac{\pi}{10}\right)\right)\right), \tag{20}$$

where $\mu_i$ and $\sigma_i$ are the mean and standard deviation of (19), respectively, on $\mathcal{L}_i$. Our method is iterated until the number of inliers becomes stable. Although our greedy approach does not guarantee the global optimum, it works well for a variety of inputs as will be demonstrated in the next section.

## 4 Experimental results

We evaluate our method on a variety of images and compare it with three state-of-the-art techniques (i.e., Upright [13], SfAR [25], and TILT [26]). Performances of the algorithms are evaluated in terms of the computation time and the amount of remaining geometrical distortions after the rectification [17]. In all the experiments, we set $\lambda$ in (12) to 0.1, and $k$ in (20) to 2.

### 4.1 Evaluation metrics

Geometric distortions are measured with four criteria: orthogonality, diagonal ratio, and length ratios for opposite sides. Let us assume that we have the rectification homography $\mathrm{H}^{-1}$ and four (manually annotated) corner points of an object in input image $p$, $q$, $s$, and $r$ as illustrated in Fig. 1. Then, the diagonal ratio $r_d$ is defined as the length ratio between two diagonals:

$$r_d = \max\left(\frac{d(\mathrm{H}^{-1}\mathbf{q}, \mathrm{H}^{-1}\mathbf{s})}{d(\mathrm{H}^{-1}\mathbf{p}, \mathrm{H}^{-1}\mathbf{r})}, \frac{d(\mathrm{H}^{-1}\mathbf{p}, \mathrm{H}^{-1}\mathbf{r})}{d(\mathrm{H}^{-1}\mathbf{q}, \mathrm{H}^{-1}\mathbf{s})}\right). \tag{21}$$

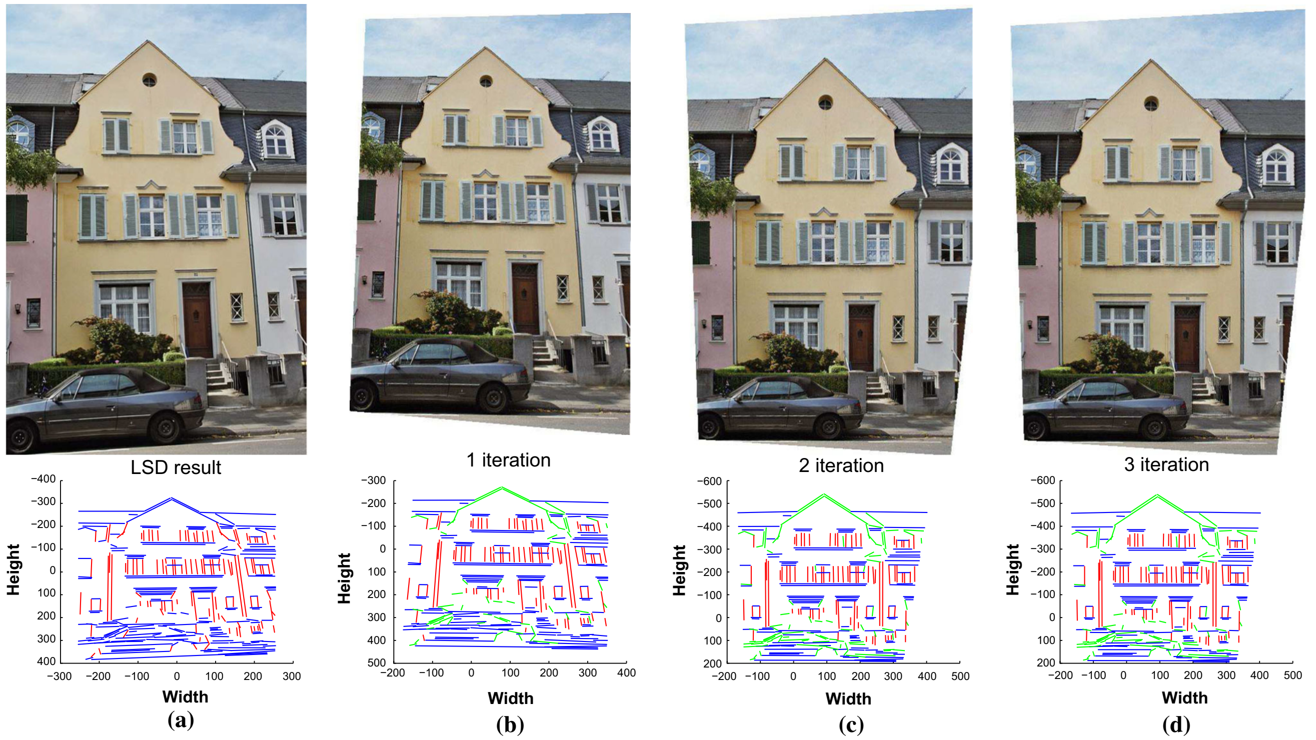

**Fig. 3** Illustration of iterative approach. The first row shows current results (by applying $\mathrm{H}^{-1}$) to the input image, and the images in the second row are corresponding LSD results. *Blue* and *red* are the lines close to the *horizontal* and *vertical axes*, respectively, and green lines are outliers in the iteration. **a** input, **b** after first iteration, **c** after second iteration, **d** after third iteration

Also, two pairs of opposite sides must have the same length, and we evaluate the ratio of two vertical sides as

$$r_v = \max\left(\frac{d(\mathrm{H}^{-1}\mathbf{p}, \mathrm{H}^{-1}\mathbf{s})}{d(\mathrm{H}^{-1}\mathbf{q}, \mathrm{H}^{-1}\mathbf{r})}, \frac{d(\mathrm{H}^{-1}\mathbf{q}, \mathrm{H}^{-1}\mathbf{r})}{d(\mathrm{H}^{-1}\mathbf{p}, \mathrm{H}^{-1}\mathbf{s})}\right). \tag{22}$$

The ratio for the horizontal pair $r_h$ is similarly defined. Finally, we measure the orthogonality of four corners. Specifically, for the upper left corner, we compute

$$\theta_o = \cos^{-1}\left(\frac{\mathbf{l}_1^\top \mathrm{C}_\infty^* \mathbf{l}_2}{\sqrt{(\mathbf{l}_1^\top \mathrm{C}_\infty^* \mathbf{l}_1)(\mathbf{l}_2^\top \mathrm{C}_\infty^* \mathbf{l}_2)}}\right), \tag{23}$$

where $\mathrm{C}_\infty^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ is the conic dual of the Euclidean coordinate system [8], and two lines are given by $\mathbf{l}_1 = (\mathrm{H}^{-1}\mathbf{p}) \times (\mathrm{H}^{-1}\mathbf{q})$ and $\mathbf{l}_2 = (\mathrm{H}^{-1}\mathbf{s}) \times (\mathrm{H}^{-1}\mathbf{p})$.

We use the deviation of these values from the ideal ones as evaluation metrics. Note that the ideal values for $r_d, r_h$, and $r_v$ are 1, and the ideal value for the orthogonality is $90^\circ$, and we evaluate $|r_d - 1|$, $|r_h - 1|$, $|r_v - 1|$, and $|\theta_o - 90^\circ|$ for the performance evaluation. When there are two or more planar targets in an input image, we compute their average as illustrated in Fig. 4.

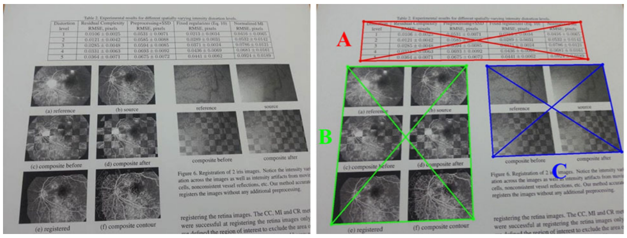

**Fig. 4** When there are two or more rectangles in an input image, we compute the average of errors (i.e., a, b, c *rectangles*)

### 4.2 Quantitative evaluation

We evaluate our method and the conventional methods [13, 25, 26] on four datasets.

- Business card images (*Card*): 100 images
- Document images including tables and/or figures (*Document*): 100 images
- Building facade images (*eTRIMs* [11]): 60 images
- Signpost images (*Signpost*): 30 images

**Table 1** Performance comparison of the proposed method with upright adjustment [13], SfAR [25] and TILT [26]. The comparison is based on four values: mean and median as measurements of center, and standard deviation (SD) and interquartile range (IQR) as those of spread. The boldface represents the best result in each row

| Measure | | Original | Upright [13] | SfAR [25] | TILT [26] | Proposed |
|---|---|---|---|---|---|---|
| Orthogonality | Mean | 6.0090 | 2.6052 | 2.4062 | 3.8459 | **0.9322** |
| | Median | 4.6523 | 1.1117 | 0.9817 | 0.8715 | **0.5175** |
| | SD | 4.8857 | 3.8343 | 5.4250 | 8.1598 | **1.6371** |
| | IQR | 6.7114 | 2.2954 | 1.6388 | 3.1150 | **0.7575** |
| Diagonal ratio | Mean | 0.0390 | 0.0180 | 0.0347 | 0.0291 | **0.0089** |
| | Median | 0.0246 | 0.0065 | 0.0084 | 0.0069 | **0.0059** |
| | SD | 0.0436 | 0.0346 | 0.1381 | 0.0644 | **0.0163** |
| | IQR | 0.0441 | 0.0140 | 0.0126 | 0.0200 | **0.0115** |
| Vertical ratio | Mean | 0.1062 | 0.0630 | 0.0902 | 0.0517 | **0.0156** |
| | Median | 0.0742 | 0.0211 | 0.0207 | 0.0132 | **0.0088** |
| | SD | 0.1028 | 0.1542 | 0.2928 | 0.1332 | **0.0301** |
| | IQR | 0.1426 | 0.0509 | 0.0472 | 0.0299 | **0.0177** |
| Horizontal ratio | Mean | 0.0979 | 0.0415 | 0.0552 | 0.0550 | **0.0117** |
| | Median | 0.0729 | 0.0139 | 0.0148 | 0.0081 | **0.0048** |
| | SD | 0.0884 | 0.0626 | 0.1809 | 0.1300 | **0.0291** |
| | IQR | 0.1085 | 0.0413 | 0.0273 | 0.0303 | **0.0101** |

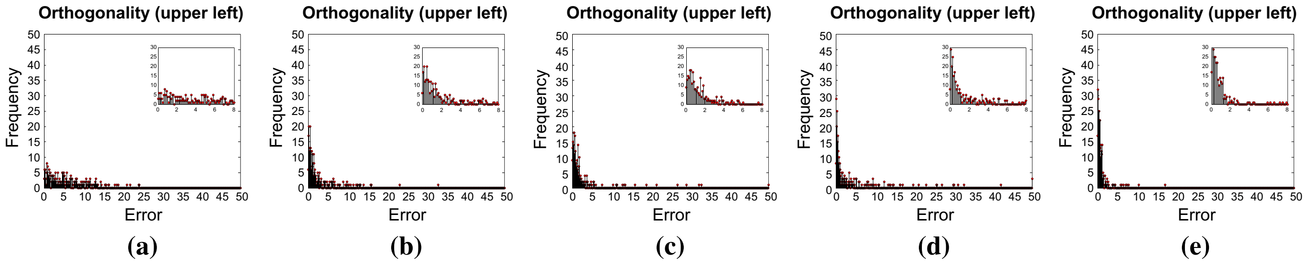

**Fig. 5** Error histograms of orthogonality. **a** original, **b** upright [13], **c** SfAR [25], **d** TILT [26], **e** the proposed method

**Table 2** The execution time (sec). In the case of [13], only a part of source code is available and we evaluate its execution time. The boldface shows the best result

| Method | Card | Document | ETRIMs | Signpost |
|---|---|---|---|---|
| Upright [13] | $\geq 2.02$ | $\geq 2.26$ | $\geq 3.20$ | $\geq 3.81$ |
| SfAR [25] | 8.91 | 42.29 | 53.37 | 37.23 |
| TILT [26] | 5.14 | 6.57 | 7.88 | 6.41 |
| Ours (C++) | **0.13** | **0.17** | **0.18** | **0.21** |

For the conventional methods [13,25,26], we use the source code or programs implemented by the authors. Since SfAR [25] requires the exact focal length, we use the focal length from the camera's EXIF data (for *Card* and *Document*) when the data are available. Otherwise, the automatic camera calibration method [13] is used for the estimation of focal length (for *eTRIMs* and *Signpost*). For TILT [26], we manually provide regions of interest for the initialization of the method. The rectification methods discussed in Sec. 1.1 are not compared with our method, because they are focused on finding individual cases (e.g., the presence of a single dominant rectangle).

Evaluation results are summarized in Table 1, which shows that the proposed method yields the least errors. Since there are some failure cases, we also evaluate the standard deviation and interquartile range that can reflect the spread of values in a distribution. It can be seen that our method shows the smallest values for all datasets. We illustrate the error histogram for the orthogonality in Fig. 5. As shown in the histogram, our method shows sharp peak around zero, and the number of failure cases is smaller than others. The histograms of other measures also show similar patterns. In summary, the proposed method works robustly for the 2D rectification problem, because the line alignments are the salient features for 2D case and the proposed method exploits them.

### 4.3 Computation complexity

The proposed algorithm is implemented with C++, and the computation time is measured on a PC with AMD Phenom(tm) II X6 1055T Processor. The computational cost of our method is compared to the conventional methods [13,25,26] in Table 2, which shows that our method is effi-

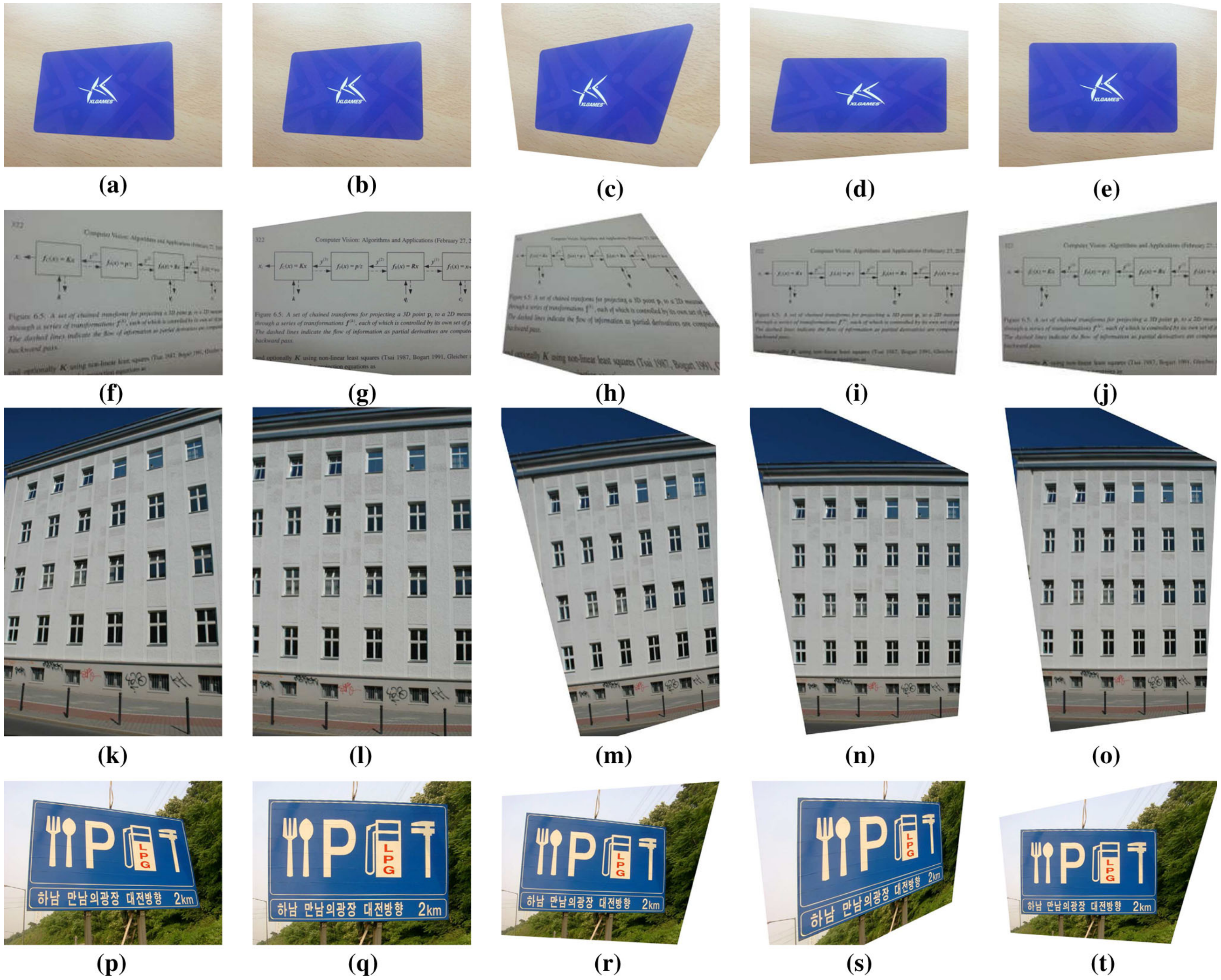

**Fig. 6** Sample results for each dataset. *First column* input images. *Second column* results of [13]. *Third column* result of [25]. *Fourth column* result of [26]. *Last column* results of the proposed method

cient. One might think this comparison is biased, because conventional methods were implemented with MATLAB. However, since (a) LSD [5] takes about 90 % of the computation time in the proposed method and (b) the LSD was also implemented with C-mex in the conventional methods [13,25], it is believed our method is more efficient than conventional approaches.

### 4.4 Qualitative comparisons and limitations

Figures 6 and 7 show the rectification results of conventional methods [13,25,26] and the proposed method for sample images of our dataset and the ones from [13]. Full-resolution of more inputs/results can be found at http://ispl.snu.ac.kr/jhahn/LsdRectification.

Because our method depends on line segments, it sometimes fails when the majority of the extracted line segments are not aligned with principal axes (see Fig. 8). Another limitation is that our method may not perform the exact metric rectification when the focal length largely deviates from our assumption, i.e., (17). However, the exact focal length estimation is impossible from a planar rectangle target [8], and our method yields reasonable results for a variety of inputs as shown in Fig. 6.

## 5 Conclusion

In this paper, we have proposed an efficient and robust method for the rectification of planar targets based on line segments. For the goal, we parameterize the camera parameters with four variables and develop a cost function defined on line segments, which can be easily minimized via the Levenberg–Marquardt algorithm. According to the experiments, the proposed method shows better performance for

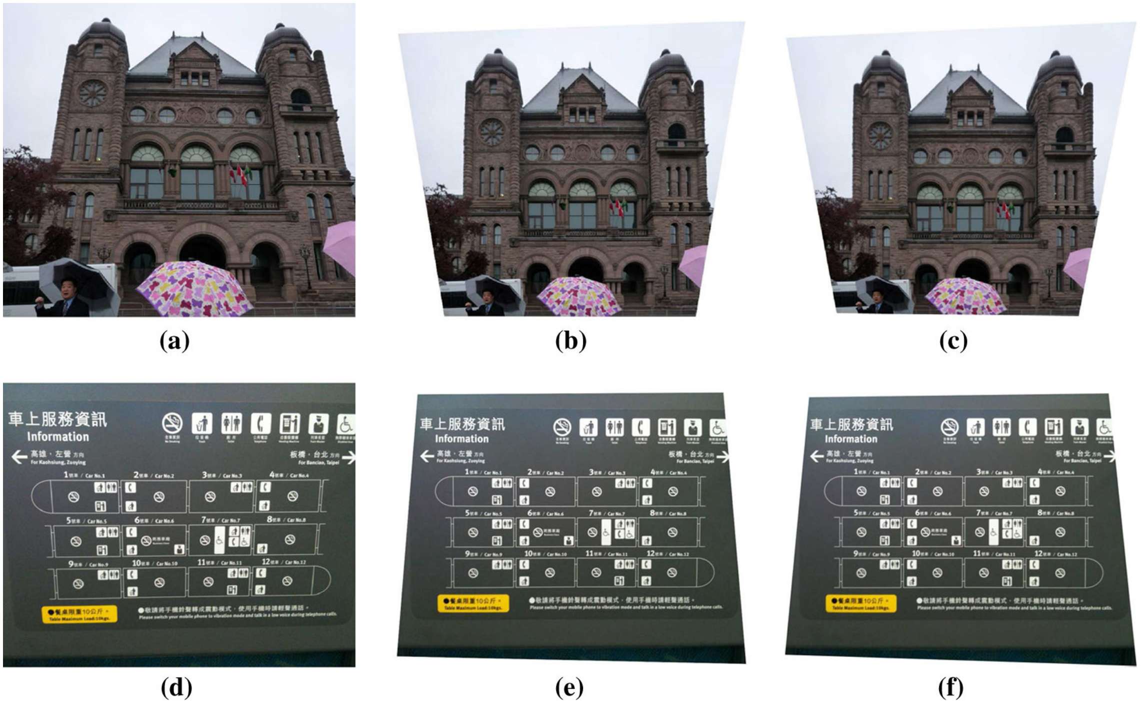

**Fig. 7** Examples of planar target in [13]. *First column* input images. *Second column* results of [13]. *Last column* results of the proposed method

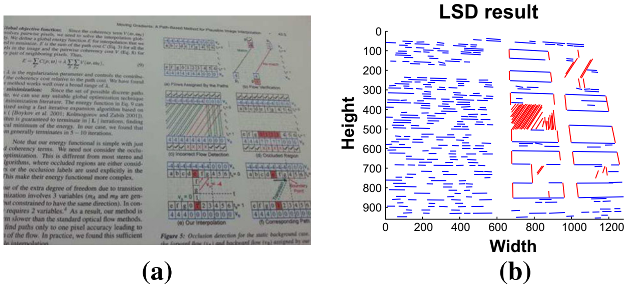

**Fig. 8** A failure example. **a** input image, **b** LSD results

the planar objects compared to the state-of-the-art self-calibration methods, as the proposed method is specialized for the 2D rectification. Hence, it is believed that the proposed method is a competent method for the rectification of name cards, signposts and documents where the rectification of 2D planar objects is the main goal.

**Acknowledgments** This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1016) supervised by the IITP (Institute for Information & communications Technology Promotion).

## Appendix: Jacobian matrix of the proposed cost function

The cost function (12) can be minimized via the Levenberg–Marquardt algorithm. For the efficient implementation of the algorithm, we need the derivatives of

$$d_{\mu}(\mathrm{H}^{-1}\mathbf{u}, \mathrm{H}^{-1}\mathbf{v}) \tag{24}$$

and

$$\frac{\max(a, f)}{\min(a, f)} \tag{25}$$

with respect to four parameters(i.e., $t \in \{\theta_1, \theta_2, \theta_3, f\}$). However, the $\min(\cdot,\cdot)$ function in (24) is not differentiable at some points; therefore, we use a simple approximation:

$$\frac{\partial}{\partial t}(\min(f(\cdot), g(\cdot))) = \begin{cases} \frac{\partial f(\cdot)}{\partial t} & \text{if} \quad f(\cdot) \leq g(\cdot) \\ \frac{\partial g(\cdot)}{\partial t} & \text{otherwise.} \end{cases} \tag{26}$$

Although this approximation introduces ambiguities when $f(\cdot) = g(\cdot)$, this case seldom happens and the approximation works well in practice. We handle $\min(\cdot,\cdot)$ and $|\cdot|$ functions in a similar manner.

### Derivatives of (24)

Let us denote $\hat{\boldsymbol{p}} = \mathrm{H}^{-1}\mathbf{u} = \left[\hat{p}_1 \ \hat{p}_2 \ \hat{p}_3\right]^\top$ and $\hat{\boldsymbol{q}} = \mathrm{H}^{-1}\mathbf{v} = \left[\hat{q}_1 \ \hat{q}_2 \ \hat{q}_3\right]^\top$ and denote their inhomogeneous representation as $\tilde{\boldsymbol{p}} = \left[x_1(\cdot) \ y_1(\cdot)\right]^\top$ and $\tilde{\boldsymbol{q}} = \left[x_2(\cdot) \ y_2(\cdot)\right]^\top$, respectively.

Then, the derivative of (24) with respect to $t$

$$\frac{\partial}{\partial t}\left[d_{\mu}(\hat{\boldsymbol{p}}, \hat{\boldsymbol{q}})\right] = \frac{\partial}{\partial t}\left[\min\left(\left|x_1(\cdot) - x_2(\cdot)\right|, \left|y_1(\cdot) - y_2(\cdot)\right|\right)\right] \tag{27}$$

can be decomposed into four cases:

$$\begin{cases} \frac{\partial}{\partial t}\left(x_1(\cdot) - x_2(\cdot)\right) & \text{if } x_1(\cdot) > x_2(\cdot) \;\&\; |x_1(\cdot) - x_2(\cdot)| \le |y_1(\cdot) - y_2(\cdot)| \\ \frac{\partial}{\partial t}\left(x_2(\cdot) - x_1(\cdot)\right) & \text{if } x_1(\cdot) \le x_2(\cdot) \;\&\; |x_1(\cdot) - x_2(\cdot)| \le |y_1(\cdot) - y_2(\cdot)| \\ \frac{\partial}{\partial t}\left(y_2(\cdot) - y_1(\cdot)\right) & \text{if } y_1(\cdot) \le y_2(\cdot) \;\&\; |x_1(\cdot) - x_2(\cdot)| > |y_1(\cdot) - y_2(\cdot)| \\ \frac{\partial}{\partial t}\left(y_1(\cdot) - y_2(\cdot)\right) & \text{otherwise.} \end{cases} \tag{28}$$

Since we can derive $\frac{\partial \tilde{\boldsymbol{p}}}{\partial t} = \left[\frac{\partial x_1(\cdot)}{\partial t} \; \frac{\partial y_1(\cdot)}{\partial t}\right]^{\top}$ by using a chain rule:

$$\frac{\partial \tilde{\boldsymbol{p}}}{\partial t} = \frac{\partial \tilde{\boldsymbol{p}}}{\partial \hat{\boldsymbol{p}}} \frac{\partial \hat{\boldsymbol{p}}}{\partial t} \tag{29}$$

where

$$\frac{\partial \tilde{\boldsymbol{p}}}{\partial \hat{\boldsymbol{p}}} = \frac{\partial\left[\hat{p}_1/\hat{p}_3 \;\; \hat{p}_2/\hat{p}_3\right]}{\partial\left[\hat{p}_1 \;\; \hat{p}_2 \;\; \hat{p}_3\right]} = \begin{bmatrix} 1/\hat{p}_3 & 0 & -\hat{p}_1/\hat{p}_3^2 \\ 0 & 1/\hat{p}_3 & -\hat{p}_2/\hat{p}_3^2 \end{bmatrix}, \tag{30}$$

it is sufficient to compute $\frac{\partial \hat{\boldsymbol{p}}}{\partial t}$ and $\frac{\partial \hat{\boldsymbol{q}}}{\partial t}$ to get (27). By using (8), we can get $\frac{\partial \hat{\boldsymbol{p}}}{\partial t}$ for each parameter:

$$\frac{\partial \hat{\boldsymbol{p}}}{\partial \theta_1} = \left\{\mathbf{I} - \frac{(\mathbf{t} + \mathbf{e}_3)\,\mathbf{e}_3^{\top}}{\mathbf{e}_3^{\top}\mathbf{t}}\right\} \frac{\partial \mathbf{R}^{\top}}{\partial \theta_1} \mathbf{K}^{-1}\mathbf{u}, \tag{31}$$

$$\frac{\partial \hat{\boldsymbol{p}}}{\partial \theta_2} = \left\{\mathbf{I} - \frac{(\mathbf{t} + \mathbf{e}_3)\,\mathbf{e}_3^{\top}}{\mathbf{e}_3^{\top}\mathbf{t}}\right\} \frac{\partial \mathbf{R}^{\top}}{\partial \theta_2} \mathbf{K}^{-1}\mathbf{u}, \tag{32}$$

$$\frac{\partial \hat{\boldsymbol{p}}}{\partial \theta_3} = \left\{\mathbf{I} - \frac{(\mathbf{t} + \mathbf{e}_3)\,\mathbf{e}_3^{\top}}{\mathbf{e}_3^{\top}\mathbf{t}}\right\} \frac{\partial \mathbf{R}^{\top}}{\partial \theta_3} \mathbf{K}^{-1}\mathbf{u}, \tag{33}$$

$$\frac{\partial \hat{\boldsymbol{p}}}{\partial f} = \left\{\mathbf{I} - \frac{(\mathbf{t} + \mathbf{e}_3)\,\mathbf{e}_3^{\top}}{\mathbf{e}_3^{\top}\mathbf{t}}\right\} \mathbf{R}^{\top} \frac{\partial \mathbf{K}^{-1}}{\partial f} \mathbf{u}. \tag{34}$$

The derivatives of $\frac{\partial \hat{\boldsymbol{q}}}{\partial t}$ can be derived in a similar way.

### Derivatives of (25)

We can get the derivative of (25) with respect to $f$ similarly:

$$\frac{\partial}{\partial f}\left[\frac{\max(a, f)}{\min(a, f)}\right] = \begin{cases} \frac{1}{a}, & \text{if } a < f \\ -\frac{a}{f^2}, & \text{if } a > f \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

## References

1. Buenaposada, J.M., Baumela, L.: Real-time tracking and estimation of plane pose. In: Proceedings of International Conference on Pattern Recognition, pp. 697–700 (2002)
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, pp. 421–430 (2001)
3. Cobzas, D., Jagersand, M., Sturm, P.: 3d ssd tracking with estimated 3d planes. Image Vis. Comput. **27**, 69–79 (2009)
4. Corral-Soto, E.R., Elder, J.H.: Automatic single-view calibration and rectification from parallel planar curves. In: European Conference on Computer Vision, pp. 813–827 (2014)
5. Grompone, V.G.R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. **32**(4), 722–732 (2010)
6. Hanbury, A., Wildenauer, H.: Robust camera self-calibration from monocular images of manhattan worlds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2831–2838 (2012)
7. Hartl, A., Reitmayr, G.: Rectangular target extraction for mobile augmented reality applications. In: Proceedings of International Conference on Pattern Recognition, pp. 81–84 (2012)
8. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
9. Hua, G., Liu, Z., Zhang, Z., Wu, Y.: Automatic business card scanning with a camera. In: IEEE International Conference on Image Processing, pp. 373–376 (2006)
10. Hua, G., Liu, Z., Zhang, Z., Wu, Y.: Iterative local-global energy minimization for automatic extraction of objects of interest. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1701–1706 (2006)
11. Korč, F., Förstner, W.: eTRIMS Image Database for interpreting images of man-made scenes. Tech. Rep. TR-IGG-P-2009-01, Department of Photogrammetry, University of Bonn (2009)
12. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic upright adjustment of photographs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 877–884 (2012)
13. Lee, H., Shechtman, E., Wang, J., Lee, S.: Automatic upright adjustment of photographs with robust camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **36**(5), 833–844 (2014)
14. Lee, W., Pack, Y., Lepetit, V.: Video-based In Situ tagging on mobile phones. IEEE Trans. Circuits Syst. Video Techn. **21**, 1487–1496 (2011)
15. Liebowitz, D., Zisserman, A.: Metric rectification for perspective images of planes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 482–488 (1998)
16. Mirzaei, F., Roumeliotis, S.: Optimal estimation of vanishing points in a manhattan world. In: IEEE International Conference on Computer Vision, pp. 2454–2461 (2011)
17. Monasse, P., Morel, J.M., Tang, Z.: Three-step image rectification. In: The British Machine Vision Conference, pp. 89.1–10 (2010)
18. Mor, J.: The Levenberg-Marquardt algorithm: Implementation and theory. In: Watson, G. (ed.) Numerical Analysis. Lecture Notes in Mathematics, vol. 630, pp. 105–116. Springer, Berlin Heidelberg (1978)
19. Pilu, M.: Extraction of illusory linear clues in perspectively skewed documents. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. I363–I368 (2001)
20. Pritts, J., Chum, O., Matas, J.: Detection, rectification and segmentation of coplanar repeated patterns. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2973–2980 (2014)
21. Tardif, J.P.: Non-iterative approach for fast and accurate vanishing point detection. In: IEEE International Conference on Computer Vision, pp. 1250–1257 (2009)

22. Tretyak, E., Barinova, O., Kohli, P., Lempitsky, V.: Geometric image parsing in man-made environments. Int. J. Comput. Vis. **97**(3), 305–321 (2012)
23. Xu, C., Kuipers, B., Murarka, A.: 3d pose estimation for planes. In: Proceedings of International Conference on Computer Vision Workshops, pp. 673–680 (2009)
24. Xu, Y., Oh, S., Hoogs, A.: A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1376–1383 (2013)
25. Zaheer, A., Rashid, M., Khan, S.: Shape from angle regularity. In: Proceedings of the 12th European Conference on Computer Vision, vol part VI, pp. 1–14 (2012)
26. Zhang, Z., Ganesh, A., Liang, X., Ma, Y.: Tilt: Transform invariant low-rank textures. Int. J. Comput. Vis. **99**(1), 1–24 (2012)
27. Zhang, Z., Matsushita, Y., Ma, Y.: Camera calibration with lens distortion from low-rank textures. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2321–2328 (2011)

**Jaehyun An** received BS degree from the Department of Information Electronics Engineering, Ewha Womans University, and MS and PhD from the Department of Electrical Engineering and Computer Science, Seoul National University (SNU), Seoul, Republic of Korea, in 2009, 2011, and 2015, respectively. Her research interests include image processing and computer vision.

**Hyung Il Koo** received BS, MS, and PhD from the Department of Electrical Engineering and Computer Science from Seoul National University (SNU), Seoul, Korea, in 2002, 2004, and 2010, respectively. From 2010 to 2012, he was a Research Engineer in the Qualcomm Research Korea. He joined the Department of Electrical and Computer Engineering, Ajou University, in 2012, where he is currently an Associate Professor. His research interests include computer vision and machine learning.

**Nam Ik Cho** received BS, MS, and PhD in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1986, 1988, and 1992, respectively. From 1991 to 1993, he was a Research Associate of the Engineering Research Center for Advanced Control and Instrumentation, Seoul National University. From 1994 to 1998, he was with the University of Seoul, Seoul, Korea, as an Assistant Professor of Electrical Engineering. He joined the School of Electrical Engineering, Seoul National University, in 1999, where he is currently a Professor. His research interests include speech, image, video signal processing, and adaptive filtering.