

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Content Authoring Using Single Image in Urban Environments for Augmented Reality

Nan-Hung Cho, Qiang Wu, Jingsong Xu, Jian Zhang

Global Big Data Technologies Centre

University of Technology Sydney, Australia

Nan.H.Cho@student.uts.edu.au, {Qiang.Wu, JingSong.Xu, Jian.Zhang}@uts.edu.au

Abstract—Content authoring is one of essentials of Augmented Reality (AR), which is to emplace an augmented content on a true part of a real scene in order to enhance users' visual experience. For the case of street view single 2D images, the challenge emerges because of clutter environments and unknown position and orientation related to camera pose. Although existing methods based on 2D feature point matching or vanishing point registration may recover the camera pose, the robustness is always challenging because of the uncertainty of feature point detection on texture-less region and displacement of vanishing point detection caused by irregular lines detected on the scene. By taking the advantages of characteristics of the man-made object (e.g. building) widely seen on the street view, this paper proposes a simple yet efficient content authoring approach. In this approach, the building dominant plane where the virtual object will be emplaced is detected and then projected to the frontal-parallel view on which the virtual object can be reliably emplaced. Once the virtual object and the true scene are embedded to each other on the frontal-parallel view, they are able to be converted back to the original view using inverse projection without any distortion. Experiments on public databases show that the proposed method can recover camera pose and implement content emplacement with promising performance.

Keywords—Augmented Reality; single image; dominant planar view; content authoring

I. INTRODUCTION

Augmented Reality (AR) allows an augmented content to be overlaid onto part of an image or video clip from the real world as shown in Fig.1. The biggest virtue about AR is to enhance users' visual experience about the physical world with an extra information relative to the real world environment [1]. One of the most critical aspect about an AR system is to produce a consistent alignment of the augmented content with the real world image [2]. This leads to the problem of solving an unknown position and orientation of the camera for a given image taken with respect to the augmented content we wish to render, known as camera pose estimation [3].

In recent years AR has begun to attract more attention because of the high potential of integration with many human interaction applications. Thus developing a solution that produces an accurate AR view effect has become a popular research topic. However, depending on complexity of scene structure and estimation approach, this process can be computationally expensive, or even requires non-linear optimisation methods. The computational requirement has appeared to be a major challenge point in many AR applications. In particular, for a system that has constraints on its computation power such as mobile phone or tablet device, an approach that reduces

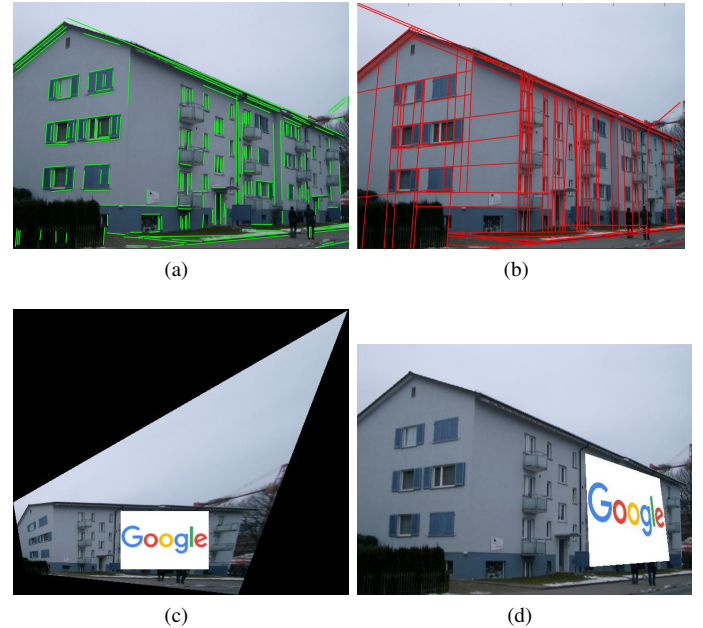


Fig. 1: An overview of our content authoring method. (a) shows the result of line segment detection that outlines the structure of overall building structure. (b) is the line segment result after gap filling. (c) is the result of metric rectification that brings one of the building dominant plane to frontal-parallel view. Users are then allowed to put a content image onto intended augment region. This defined region is then back-projected and replaced with an augmented content in the original view (d).

complexity is required. In this paper, we propose a method to automatically recover building major planes using a single image. Importantly, there is no presumption about global XYZ coordinate system in the image plane.

The contribution of this paper are (i) we propose a simple yet efficient method to find dominant frontal-parallel view of each building facade based on single image. (ii) Through the rectified frontal-parallel view, content authoring (i.e. object emplacement) for AR is successfully carried out without distortion, which shows improved performance compared to the existing methods.

The rest of the paper are organised as follows: Section II reviews current existing approaches related to image-based camera pose estimation and augmented reality. Our main

pipeline of dominant frontal-parallel view recovery is described in Section III. Based on this result, the approach of AR content annotation is described in Section IV. The performance of the proposed method is experimentally evaluated in Section V. Finally, conclusion and future work outline our ongoing works for the next stage in Section VI.

II. RELATED WORK

There are three mainstream approaches aiming to solve the complexity of camera pose estimation for AR applications. One is to make use of the proprioceptive sensors [4] such as GPS or digital compass. The second approach is computer vision based [5] and finally a hybrid approach [4]. In this paper we focus on the computer vision approach. There are couple of advantages of computer vision based AR. One is the fact that the pinhole camera has become a ubiquitous peripheral device as part of many computer systems. Also, the costs are usually cheaper and the system setup and configuration are easier. The fundamental idea is to take advantages or making assumptions within the image plane to extract the geometry or good feature points to estimate camera pose. For feature point matching such as SIFT, these encoded feature points are robust against changes in image scale or even partially occlusion. The review of recent methods can be found in [6], [7], [8]. The feature point extraction relies on searching for potential local extrema and the generation of SIFT descriptor for the subsequent feature matching. There are other works [9], [10], [11] which intend to further improve speed and overall robustness of SIFT based approach for AR application. However, when applying this method in the context AR application, the encoded information has a higher chance to fail in feature matching due to viewpoint changes of the camera or perspective projection of the scene structure.

The other category is to exploit the geometric property such as epipolar lines rectification [12] or to find the rectangle structure in a man-made environment such as building facade, street view or indoor layout. Our proposed method belongs to this category. In regards to rectangle structure, previous works have observed that these man-made environments usually contain a great amount of parallel lines aligned along with some fixed orthogonal. But due to perspective projection, parallel lines in 2D image plane will eventually converge at vanishing point. The parallel lines and their corresponding orthogonal directions can be used to produce a “rectified view” of the image scene, i.e. metric rectification. The purpose of generating this rectified view is to transform a set of consecutive image planes with different and unknown pose to a common ground in order to estimate the homographies relation between each inter-frame.

The presence of such common ground is viewpoint invariant and can be helpful for simplifying the camera calibration application. [13] utilised this approach in their pipeline to recover rectangle structures from in the rectified view in order to realise a single view based 3D reconstruction. [14] also proposed a method of finding vanishing point and camera parameters using mutually orthogonal line segment pattern found in Manhattan structures. [15] took the advantage of vanishing point in Manhattan structure to develop a method for building facade extraction and recovering the camera pose [16], [17] for a typical AR application. However, [15] took

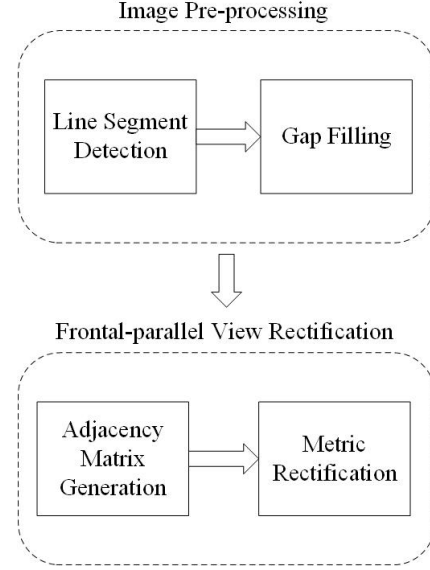


Fig. 2: Dominant frontal-parallel recovery pipeline. Line segments are firstly detected, followed by gap filling between lines if they are closed enough. Then using this line segment information and the orthogonal property of rectangle structure, a set of line-pairs is generated. Finally in the metric rectification stage, we further optimise and label these line-pairs with respect to the dominant plane.

all line segments into account throughout the entire process. This result can be easily distorted if the building structure does not fill up the entire image plane because they rely heavily on the result of vanishing point detection. In addition, it requires manually selection of line segment structure to recover the rectified view.

Inspired by [13], our proposed method adopts the orthogonality property in building structure to guesstimate a set of homography. Such homography can bring dominant facade of the building to frontal-parallel view in a single image frame. Using the inverse relation of this rectified homography we can further recover the unknown camera orientation of each facade in the original view. The advantage of such method is the fact that we do not make presumption about global XYZ orthogonal coordinate system in the image plane. In our experiment, we demonstrated our method can achieve a better result compared to [15].

III. DOMINANT FRONTAL-PARELLEL VIEW RECOVERY

In this section we present the method of dominant frontal-parallel view recovery as illustrated in Fig. 2. Given an image taken in urban environment, our system is to find a number of hypothesis planes in this image as well as generate a set of homography parameters of recovered planes. The overall pipeline need to be performed in the following sequence: (1) Line segment detection and gap filling; (2) Orthogonal line-pair candidate matrix generation; (3) Metric rectification.

A. Line Segment Detection and Gap Filling

The very first step is line segment detection. A line segment is the section of line defined by two definite end points that outline the building structure. In our application we adopt the

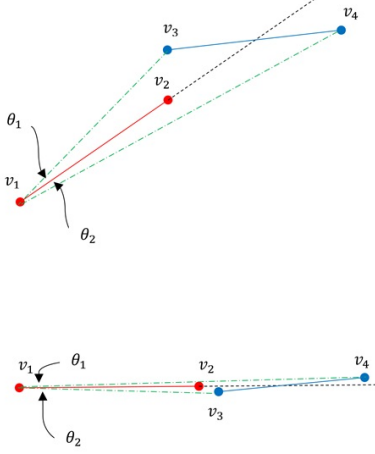


Fig. 3: An illustration of collinearity checking. θ_1 and θ_2 are measured based on the angle difference between l_a to the line segment that joins the two endpoints of l_b . The example on the top shows a bigger angle difference compared to case at the bottom.

existing algorithm described by [18]. An input image is firstly processed by Canny edge detector, and for each detected edge pixel the image derivative will be computed. The orientation is defined as:

$$\theta = \tan^{-1}\left(\frac{\partial I / \partial x}{\partial I / \partial y}\right) \quad (1)$$

I is the image, x and y are the directions to calculate gradient. The gradient direction is then quantised into a predefined number of bins ($k = 8$ in this paper). Edge pixel orientation falls within a range of interval according to the following definition:

$$\text{bin}_i = \frac{\theta}{\pi k} \text{mod}(k) \quad (2)$$

The edge pixels that belong to the same bucket are then grouped together using connected components labelling. A line fitting stage [18] will be performed in order to generate the straight line. During the line fitting, only those larger segment components are considered. Although the threshold length depends on the actual environment, in our case lines with 40 pixels length and above are selected (see green line segments in Fig.1). Then we perform a line gap filling. The reason is that line segment detection breaks down the building structures that belong to the same part into smaller segments. By examining the collinearity of line segments, it is possible to join some of them to produce a longer line segment if they are not too far away from each other and almost collinear with each other.

Given a predefined threshold, the collinear condition is examined by checking the angle between two line segments, $l_a = [v_1, v_2]$ and $l_b = [v_3, v_4]$, where v_i is a column vector which contain the pixel coordinate information. An example is shown in Fig. 3. Here θ_1 and θ_2 are the measurements of angle difference l_a to the line segment that joins the endpoints to l_b . If both measurements are below a predefined threshold θ_1 , i.e. $\theta_1 < \theta_\pi$ and $\theta_2 < \theta_\pi$ (refer to Fig. 3), then it is presumed that

they are almost collinear with each other and therefore these two line segments will be merged into a longer line segment. After this step, the candidate matrix of the orthogonal line-pair can be constructed.

B. Orthogonal line-pair Matrix Generation

In this section, the candidate matrix $C \in R^{n \times n}$ is generated, where n is the number of line segments. In prior to candidate selection, we adopted a fast algorithm [19] to examine if two line segments are intersected. For each entry the index of column and row represent the corresponding line segment index, a Boolean indicator is assigned. If two line segments, l_i and l_j , intersect with each other and also are orthogonal, a '1' will be assigned, otherwise a '0' is assigned. The candidate selection is based on the following criteria:

$$\theta_{adj} = (l_i^T \cdot l_j) \quad (3)$$

$$C_{i,j} = \begin{cases} 1, & \text{if } \theta_{adj} \geq \theta_{ortho} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

θ_{adj} is the angle difference measured in radian, C_i and C_j are the Boolean indicator of the candidate matrix at row i and column j , and θ_{ortho} is the orthogonality criterion. This is only a rough evaluation. The main purpose is to facilitate the subsequent process for random sample selection.

C. Metric Rectification

In many man-made environments the image plane under perspective projection can be rectified to frontal-parallel view. One important thing is that the aspect ratio of 2D quadrilateral needs to remain constant. [3] described a transformation homography (H) for viewing the same 3D plane due to pure rotation, that is

$$H = K R_Z^\gamma R_Y^\beta R_X^\alpha K^{-1} \quad (5)$$

Where K denotes intrinsic camera matrix. We assume square pixels and the image origin is at the camera principle point, thus K is reduced to diagonal matrix $\text{diag}(f, f, 1)$, f is focal length. R_i^ρ represents a rotation of ρ -angle around i -axis. Note that the product of $K R_Z^\gamma$ has no effect and can be omitted since it is a similarity transformation. The metric rectification function can be reduced to:

$$H = K R_Y^\beta R_X^\alpha K^{-1} \quad (6)$$

From this equation we are seeking for a rectifying homography. Given a pair of line segments l_i and l_j , the arbitrary homography will bring these lines, i.e. $H^{-T}l_i$ and $H^{-T}l_j$, to orthogonal basis. We denote the product of $H^{-T}l_i$ and $H^{-T}l_j$ as \tilde{v}_i and \tilde{v}_j , which is normalised to unit norm respectively. Since they must be perpendicular to each other, the resultant dot product $\|\tilde{v}_i \cdot \tilde{v}_j\|$ should be close to 0 as possible. Thus the optimisation of finding this arbitrary homography is to minimise the sum of the dot product result over all line segments pairs, giving the following equation:

Algorithm 1: Metric rectification using orthogonal line-pairs

Input:
 I : An input image with one or more dominant planes;
 C : The orthogonal line-pair matrix;
 d : RANSAC threshold value for cost function evaluation (7);
 t : Minimal number of line-pairs;

1 **Initialization:**
2 $C_{remain} = C$;
3 $S = \{\}$: selected inliers;
4 $H_{xy} = \{\}$: rectified homographies;
5 $m = 0$: number of plane detected
6 $\|\cdot\|$ represents the number of '1' s in the matrix
7 **while** $\|C_{remain}\| > t$ **do**
8 $D = \{\}$: current inliers
9 Randomly pick up two orthogonal line-pairs from C and
 perform iteratively cost function optimisation of finding
 the best transformation parameter W using (7)
10 **for all** C_{remain} **do**
11 Apply homography transformation W to all line
 segment pairs in C_{remain}
12 **if** $(v_i^T \cdot v_j)^2 < d$ **then**
13 $|D_{i,j}| = 1$;
14 **end**
15 **end**
16 **if** $\|D\| > t$ **then**
17 $H_{xy} \leftarrow W$;
18 $S \leftarrow S \cup D$;
19 $m \leftarrow m + 1$
20 **end**
21 Subtract the C_{remain} from C
22 **end**
Output:
 S : selected inliers in m planes;
 $H_{xy} \in R^{3 \times m}$ rectified homography in m planes.

$$\min_{\alpha, \beta} \sum_{i,j} \|v_i^T \cdot v_j\|^2 \quad (7)$$

In the previous step a set of line segment pair candidates are extracted. But it is not clear whether all of them are indeed orthogonal. Here we apply RANSAC [20] to iteratively estimate underlying model within a set of data that might contain some outliers. The process starts with an initial homography of $diag(1, 1, 1)$, and picks up any two set of line segment pairs from the adjacency generated in previous step, and then computes a rectifying homography through a number of iterations. The final result of the homography will be used to extract the inliers by counting the number of line-pairs that have become orthogonal in rectified image plane.

For each line segment pair, we need to determine which one belongs to the horizontal or vertical direction. A common approach is to perform distance or cosine similarity measurement based on vanishing point. In our case, we look at each line segment pair and examine which one is close to vertical direction vector $([1, 0])$ or horizontal direction vector $([0, 1])$, and presume the other line pairs to be other direction. After all line segment pairs are assigned to each direction vector, the similarity transform can be optimised. An example of final result is shown in Fig. 5.

The formulation of the cost function consists of two parts: vertical and horizontal. For each component, we are aiming to

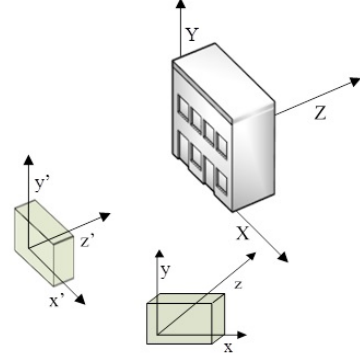


Fig. 4: Illustration of our content authoring technique of a unknown camera orientation (x, y, z) from the frontal-parallel view (x', y', z') .

look for a homography defined by a rotation parameter about Z-axis, denoted as $\gamma \in [\pi/2, \pi/2]$, that will minimise the result of dot product for these two component, i.e. as close as to 0. Therefore, this cost function could be written as follows:

$$\min_{\gamma} (\sum_i |l_i^T \cdot [0, 1]| + \sum_j |l_j^T \cdot [1, 0]|) \quad (8)$$

Where $l' = H_z^T l$. l_i and l_j are the horizontal and vertical line segments respectively. The resultant transformation is then combined with the two homographies obtained, i.e. $H = H_{xy} H_z$. This homography is finally adopted to recover the camera pose with respect to each planar surface detected. The detailed metric rectification method is shown in Algorithm 1.

IV. CONTENT AUGMENTATION METHOD

One of the critical aspect about AR is to develop a system or tools to assist the process of virtual content authoring, especially to achieve a decent accuracy of content emplacement in the AR view. The illustration in Fig. 4 shows a typical building and two camera views. Using our proposed method we obtain an estimate homography that brings the camera viewpoint to frontal-parallel view (x', y', z') . Content authoring then consists of adding content onto this plane and re-project back to the original view (x, y, z) .

Our technique permits a simplified way to achieve the same task rather than exploiting a full detail of 3D model about the environment. An example is shown in Fig. 6, an input image is firstly processed using the proposed technique to produce a set of frontal-views. Content authoring is then taken place to emplace region of interested.

The proposed method does not provide a comprehensive freedom of AR view compared to the previous approaches. However, it is suitable for situation where the content authoring can be done without a complex 3D image structure or point cloud structure. In this way, the proposed method is perfectly for smartphone or tablet application.

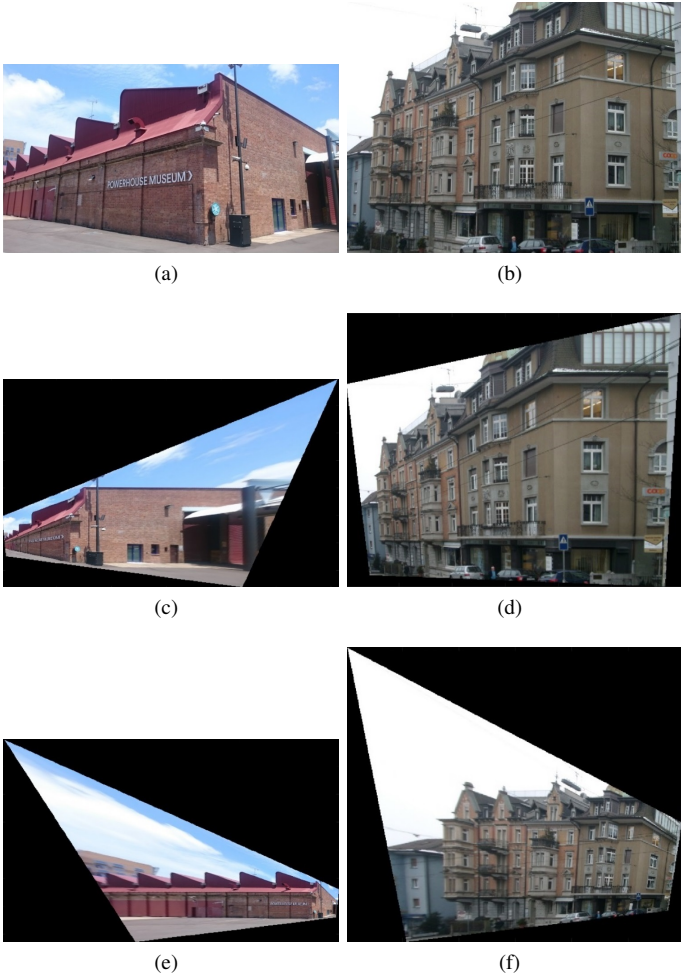


Fig. 5: Sample result of frontal-parallel view using the proposed technique. The first row shows the original input images; the second and third rows are the results of each dominant plane of the building facade in frontal-parallel view.

V. EXPERIMENT RESULT

A. Experimental Setup

The experiment is run on a PC with Intel® i5 2.4 Ghz without GPU or any hardware-accelerated component. In the early stage of development, we use the photo taken from a typical smartphone, with a known intrinsic embedded in the image file. But in the later stage, we also use a public available building databased ZuBUD [21] to examine the versatility of this proposed method.

The configuration in our experiment is setup as follows. When an input image is provided (640×360), the system first attempts to check if the image contains the camera sensor width information. If so then the camera intrinsic parameter $diag[f, f, 1]$ can be calculated analytically [3], otherwise TOV-based approach is applied [22]. Then the system performs self-rectification to frontal-parallel view using the proposed method. At each iteration after RANSAC, the system examines if these inliers have enough line segment pairs to be considered as a dominant surface. The number of inliers must have at

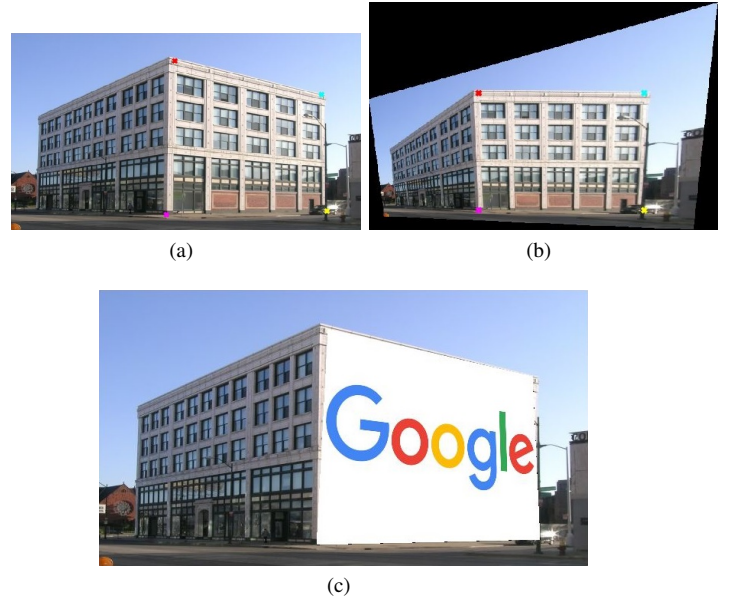


Fig. 6: Example of augmentation result. The augmentation region in (a) is defined by user in frontal-parallel view (b) represented by color points. Then a Google logo is augmented onto this region (c).

least 10% of the total amount of orthogonal line-pairs. In case of searching for two or more dominant surfaces, the corresponding orthogonal line-pairs are subtracted once the inliers are obtained. The remaining line-pairs will be processed again until the sizable line-pairs cannot be found.

B. Qualitative Analysis

The experimental results are shown in Fig. 6. Our proposed technique is effective for buildings where the facade structure is dominantly consisting of planar surface. In case of the presence of outlier objects, our proposed technique will still work when the overall planar surface is still the predominant in the image. For example in Fig. 7c, the tree and plant generate the outlier line structures which do not affect our method.

However, it also has some limitation. For instance, the building containing curved shape region will be treated as flat planar surface with respect to the dominant planar orientation. This is shown in Fig. 7c where the window is a slightly curved shape. Throughout many trials we found out this failure case might arise when the X and Y rotation parameter $[\alpha, \beta]$ is greater or less than 1.1 radian.

In terms of quality measurement, since we want to exploit the simple “crop-and-insert” approach during the AR content emplacement in the first video frame, we need to examine how well our system brings the surface to rectangle shape as well as rectifying similarity rotation. In this way two issues arise in [15]. Firstly it requires a manual selection of four line segments by user in order to recover the frontal-parallel view, and therefore it is not a self-rectification approach. Secondly in some cases of line segments selection it will result in different or even a bad frontal-parallel view. One example is shown Fig. 8. The selection of four line segments (Fig. 8a) used in [15] does not produce an acceptable frontal-parallel view

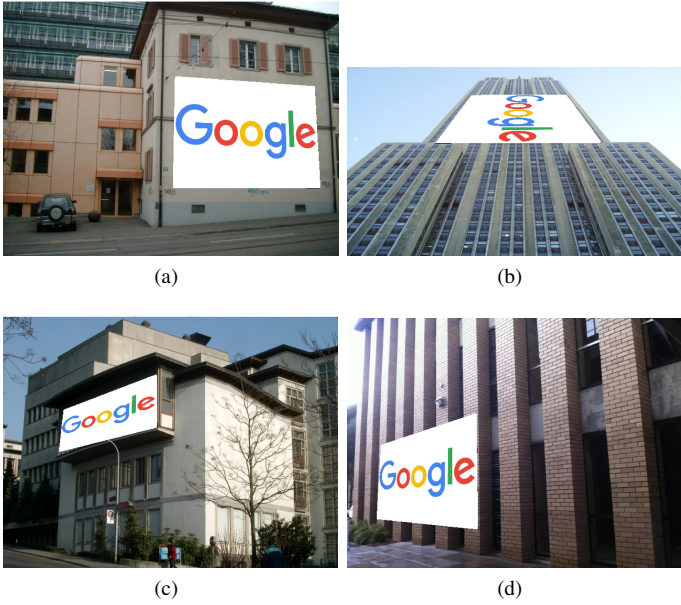


Fig. 7: Experiment results of content emplacement. Images on the left hand side are from ZuBUD database. Images on the right hand side are the typical images of building facade.

(Fig. 8c) with the set of lines we selected, and thus the content augmentation result does not provide a good effect with respect to the actual plane orientation (Fig. 8e). Due to the power cable acting as outliers object, the result of our frontal-parallel view did not perfectly been rectified and thus causes some distortion in the content augmentation. However, our method still achieves better performance than [15].

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a single image camera pose estimation technique for augmented reality application. The principle idea is to exploit the orthogonality of line structure found in man-made environments. The system works by computing a set of dominant planar segments found in the single image and generates the rectified homography to its frontal-parallel view. The inverse relation of this homography is adopted to recover the unknown camera pose relative to the original viewpoint. This proposed method has been tested on a public available database of some real-world man-made building scenes. The result of these augmented scenes was also been verified by emplacing the other image with respect to the corresponding planar surface found in the scene using the proposed technique. In future we are aiming to extend this approach for the further development of AR system that can handle a video footage. We also would like to emplace a historical image on a building scene at the same location.

REFERENCES

- [1] L. Ahyun, L. Jae-Young, S. H. Lee, and J. S. Choi, "Markerless augmented reality system based on planar object tracking," in *Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2011, pp. 1–4.
- [2] G. Simon, A. W. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," in *IEEE and ACM International Symposium on Augmented Reality (ISAR)*, 2000, pp. 120–128.

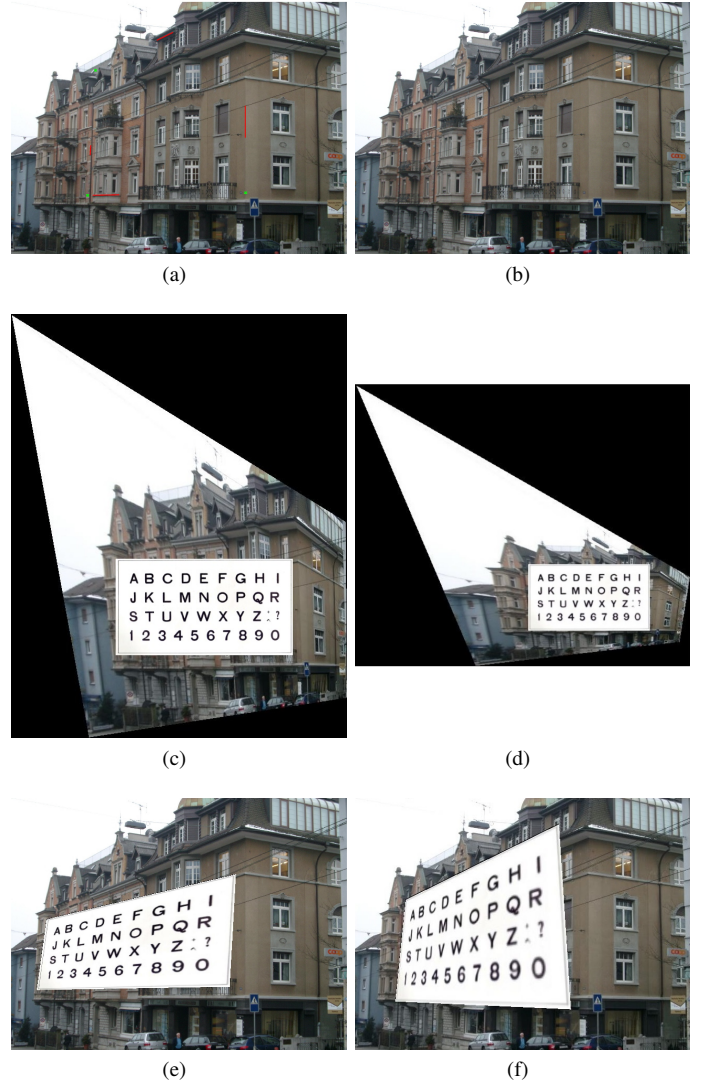


Fig. 8: Experiment results of content emplacement. (a), (c) and (e) are the results from [15], and (b), (d) and (f) are our results. The augmentation result in (e) using method [15] causes distortion on the rooftop.

- [3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [4] J. Jung, S. Lee, H. Lee, H. S. Yang, L. Weruaga, and J. Zemerly, "Real-time sensor-fusion based indoor localization for mobile augmented reality," in *International Conference on Virtual Systems Multimedia (VSMM)*, 2014, pp. 184–191.
- [5] C. Zhuo and L. Xinyu, "Markless tracking based on natural feature for augmented reality," in *International Conference on Educational and Information Technology (ICEIT)*, vol. 2, 2010, pp. 126–129.
- [6] Y. Du, Z. Miao, and Y. Cen, "Markless augmented reality registration algorithm based on orb," in *International Conference on Signal Processing (ICSP)*, 2014, pp. 1236–1240.
- [7] S. Yonemoto, "A video annotation tool using vision-based ar technology," in *International Conference on Cyberworlds (CW)*, 2012, Conference Proceedings, pp. 226–230.
- [8] Y. Xia and W. Zhou, "A tracking and registration method based on orb and klt for augmented reality system," in *Wireless and Optical Communication Conference*, 2013, pp. 344–348.
- [9] C. Yanpeng and J. McDonald, "Viewpoint invariant features from single

- images using 3d geometry,” in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2009, pp. 1–6.
- [10] M. Hamidia, N. Zenati-Henda, H. Belghit, and M. Belhocine, “Markerless tracking using interest window for augmented reality applications,” in *International Conference on Multimedia Computing and Systems (ICMCS)*, 2014, pp. 20–25.
 - [11] J.-M. Morel and G. Yu, “Asift: A new framework for fully affine invariant image comparison,” *SIAM J. Img. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.
 - [12] C. Pirschheim and G. Reitmayr, “Homography-based planar mapping and tracking for mobile phones,” in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 27–36.
 - [13] A. Zaheer, M. Rashid, and S. Khan, “Shape from angle regularity,” in *European Conference on Computer Vision*, 2012, pp. 1–14.
 - [14] C. Rother, “A new approach to vanishing point detection in architectural environments,” *Image and Vision Computing*, vol. 20, no. 910, pp. 647–655, 2002.
 - [15] E. McClean, Y. Cao, and J. McDonald, “Single image augmented reality using planar structures in urban environments,” in *Irish Machine Vision and Image Processing Conference (IMVIP)*, 2011, pp. 1–6.
 - [16] Z. Wei and J. Kosecka, “Extraction, matching and pose recovery based on dominant rectangular structures,” in *IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, 2003, pp. 83–91.
 - [17] D. Liebowitz and A. Zisserman, “Metric rectification for perspective images of planes,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, p. 482.
 - [18] W. Zhang, “Video compass,” in *European Conference on Computer Vision*, 2002, pp. 657–673.
 - [19] F. Antonio, *Faster line segment intersection*. Academic Press Professional, Inc., 1992, pp. 199–202.
 - [20] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
 - [21] T. S. H. Shao and L. V. Gool, “Zubud-zurich buildings database for image based recognition,” Report, 2004.
 - [22] B. Li, K. Peng, X. Ying, and H. Zha, “Simultaneous vanishing point detection and camera calibration from single images,” in *International Symposium on Advances in Visual Computing*, 2010, pp. 151–160.