Wall-Wall boundary detection

Group member : Zhe,Wei
CV:HCI-Practical Course Computer Vision for Human-Computer Interaction

Abstract

Recently, there has been growing interest in developing learning-based methods to detect global structures for 3D scene modeling and understanding. As a student, practice is a good opportunity to learn and understand technological development. It is a key step for 3D scene reconstruction to accurately extract room structure lines. The purpose of this exercise is to extract the wall boundary in panoramic room images. For learning purposes, three different methods have been tried: basic image processing, computer vision image processing and image processing with deep learning . The three are a progressive relationship with each other. A pretrained model [9] is used in deep learning image processing, and in the end the boundary line between the wall is accurately identified.

1. Introduction

Automatic recognition of structure from a collection of line segments is challenging, as not all lines defining the building structure are perfectly detected by low level image processing. To further complicate the problem, extra edges may lie on surfaces of walls or even on objects that are not part of the target structure.

For this reason, most existing approaches rely on mid-level area-based features, such as Geometric Context and Orientation Maps [7], as an intermediate step for layout estimation. Given an image, we determine its informative edge map and subsequently use it to predict the best-fit 3D box for the image. As images are projections of the real world, it is desirable to interpret them only in ways which can be realized in satisfying the real world. Most indoor environments the Manhattan World assumption [2], ie, most planes lie in one of three mutually orthogonal orientations .

Finding the building structure is done in three steps; line segments and vanishing points are found, many plausible building model hypotheses are created, and each hypothesis is tested against an orientation map, which is a map of local belief of region orientations, to find the best matching hypothesis.

Recent methods rely more on deep networks to improve layout estimation. Most of them leverage dense prediction models to classify geometric or semantic label for each pixel. For perspective images, common ways are to predict the boundary probability map, classes of boundaries, classes of layout surface, and corner keypoints heatmaps.

1.1. Image processing

At the beginning basic machine vision was be used to process the image.

Input:Original Panoramic image.

Output:Original image with vertical line FIG.2 .

Figure 1. Edge detection with Canny.

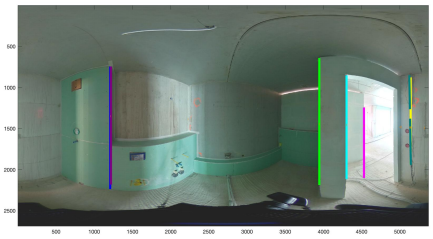

Figure 2. Results of image processing.

The color image was converted to grayscale and smoothing the image using Gaussian filter.Then the magnitude and direction of the image gradient could be calculated.After non-maximum suppressing of gradient amplitude, using Canny operator [1] with dual thresholds for edge detection and connection FIG.1 . Sobel

operator was be used to extract the vertical lines. Due to the limitations of the theory, these methods alone are not enough for higher-cognition detection. For example, if you fold a piece of white paper in half and then open it, the Canny operator can hardly recognize the crease edge, because the color of the paper on both sides of the crease is almost the same.Finally, use Hough transform to do line fitting FIG.2 .

1.2. Computer vision

Through the basic machine vision method, I realize the principle of image processing and the limitations of the basic method. Next, use computer vision depth information [5] and geometric models to break through the limitations of a single processing method.

Input:Original Panoramic image FIG.3.

Output:Original image with vertical line FIG.4.

Figure 3. Input of computer vision method.

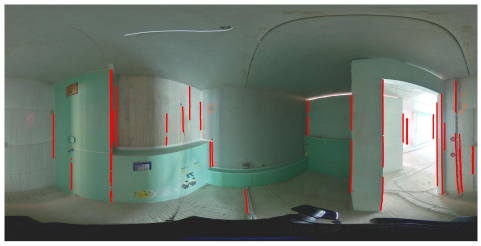

Figure 4. Results of computer vision method.

This paper [8] presents a novel edge detection method by using the local binary pattern features computed on scene deep images.

Input: a panoramic indoor image of 360° in the horizontal direction (180° in the vertical direction is not required) FIG.3.

Output: without manual interaction, the indoor three-dimensional structure (composed of line segments and super pixel patches) is automatically output FIG.4.

The line segmentation algorithm is as follows [8] :

(a)The input is the indoor panorama. You can download it from the Internet or use a panoramic camera to shoot.

(b) Line segments extracted from the panorama. Line segments with different colors of red, green and blue indicate that the line segments belong to different three-dimensional space coordinate axis directions.

Here to clarify the concept of the vanishing points(VP) [4] . Line pixels are projected into the 3D space as spatial rays (Spherical coordinates)[1]:

$$\theta = -(v - h/2) * \pi/h \tag{1}$$

$$\phi = u * 2 * \pi/w \tag{2}$$

$$xyz = \begin{pmatrix} sin(\phi) * cos(\theta) \\ cos(\theta) * cos(\phi) \\ sin(\theta) \end{pmatrix} \tag{3}$$

In panoramas, a straight line in the world is projected as an arc segment on a great circle onto the sphere and thus it appears as a curved line segment in the image. For this reason, we represent each line by the normal vector ni of the 3D projective plane that includes the line itself and the camera center. We adopt the Manhattan World assumption whereby there exist three dominant orthogonal directions. Another particularity of this type of projection is that parallel lines in the world intersect in two antipodal VP whereas in conventional images they do in one single VP. Here, we detect lines and VP by a RANSAC-based algorithm that works directly with panoramas showing entire and unique line segments, avoiding thus duplicate lines coming from different splits and improving the overall efficiency of the method.

RANSAC Algorithm:

– pick randomly two points

– fit line

– check the number of points outside the tolerance band (=number of outlines)

– repeat the process several times with different points

– select the line with the smallest number of outliers

And iterations = 100, we run a Canny edge detector on the panorama and cluster contiguous edge points in edge groups. Each point of the edge group i is projected into the 3D space as a spatial ray r_{ij} , j = 1..Npts. Iteratively, two points of each group are randomly selected (r_{i1},r_{i2}) and thus we get a possible normal direction for the edge group n_i =($r_{i1} \times r_{i2}$). The number of inliers is evaluated, i.e. how many rays fulfill the condition of perpendicularity with the normal under an angular threshold of ±0.5, $|\arccos(n_i \cdot r_{ij}) - 2\pi| \leq \theta$. After a certain number of iterations the process outputs, for each edge group, the model leading to the highest number of inliers giving the n_i that fits the line best.

[1] θ is the latitude of the point on pano.ϕ is the longitude of the point on pano. t(x,y,z)=(sin(ϕ)* cos(θ);cos(θ)*cos(ϕ);sin(θ)), (u,v): 2D points, w and h are the size of the panorama.

1.3. Computer vision with deep learning

Since the first two methods both use the Canny operator for boundary detection, the original problem still exists: the texture interference cannot be eliminated, and the boundary of the same color area cannot be detected. After discussion and research, I decided to use deep learning methods. The pipeline is in FIG.5.In the end, the nice results were obtained. This method is described in detail in the part 2.

2. Computer vision with deep learning

This method is mainly divided into three parts: image preprocessing, recognition of 1D layout [6] , and wall boundary marking. The database used by the pretrained model is Structured3D Dataset [9] .

2.1. Preprocess

All training and test images are pre-processed by the panoramic image alignment algorithm mentioned in [10]. My approach exploits the properties of the aligned panoramas that the wall-wall boundaries are vertical lines under equirectangular projection. Therefore, we can use only one value to indicate the column position of wall-wall boundary instead of two (each for a boundary endpoint).

Input:Original Panoramic image FIG.6.

Output:Aligned original Panoramic image FIG.7.

2.2. Dataset and pretrained method with RNN

Assumptions about the room structures are often made to constrain the solution space so that the predictions of the deep model would not deviate from the common cases too much. However, the ground truth annotations are often obtained via human labor, which is particularly challenging and inefficient for such tasks due to the large number of 3D structure instances (e.g., line segments) and other factors such as viewpoints and occlusions. Given a number of images with annotated layouts for training, state-of- the-art methods are able to achieve good results on the test data. However, acquiring high-quality room-layout annota-tions for panoramic images is labor-demanding. The annotations done by different people might be inconsistent due to ambiguities about the locations of wall boundaries, especially for well-decorated rooms.

Structured3D [9] was developed with the aim of providing largescale photo-realistic images with rich 3D structure annotations for a wide spectrum of structured 3D modeling tasks.

Unlike all the existing methods that use neural networks to perform dense prediction for layout estimation, HorizoNet leverage the property of aligned panorama image to predict the positions of floor-wall and ceiling-wall boundaries, as well as the existence of wall-wall boundary for each column of an equirectangular image. their model only produces three values for each column of an image, and thus the output size of the model is reduced from O(HW) to O(W). The proposed output representation is similar to [3] but they only predict floor-wall boundary for each column of a perspective image using a Dynamic Bayesian Network. In contrast, their work can handle panoramas and recognize floor-wall, ceiling-wall and wall-wall boundaries using a deep neural network. They use RNN where each "time step" is responsible for estimating the result across a few image columns.

2.3. 1D Layout Representation

The size of network output is $3 \times 1 \times 1024$. As illustrated in Fig.8 , two of the three output channels represent the ceiling-wall (y_c) and the floor- wall (y_f) boundary position of each image column, and the other one (y_w) represents the existence of wall-wall boundary (i.e. corner). The values of yc and yf are normalized to $[-\pi/2, \pi/2]$. Since defining y_w as a binary-valued vector with 0/1 labels would make it too sparse to detect (only 4 out of 1024 non-zero values for simple cuboid layout), we set $y_w(\mathrm{i}) = c^{dx}$ where i indicates the ith column, dx is the distance from the ith column to the nearest column where wall-wall boundary exists, and c is a constant [6].

2.4. Results:Wall-Wall boundary

Through deep learning modeling, we infer the coordinates of the corner on the picture and save it in a $.json$ file. Through the coordinates, we can connect the wall corners and get the boundary between wall and wall.

I only added three result pictures in the report Fig.9 More pictures and codes can be found in the program compression package.

3. Conclusion

Through the three parts of trial and practice, we gradually know the advantages and disadvantages of each method. In the process of continuous trial and learning, we have a new understanding of the knowledge of boundary detection. We can also see the interaction between theoretical update and technological progress. relationship. For me, the purpose of this practical class is only to detect the boundary of the wall, and I still need to continue learning to restore the room 3D layout later. New methods continue to optimize the results of detection, but learning and understanding the original theory also helps me under-

Figure 5. CV + Deep learning pipeline overview.

Figure 6. Original Panoramic image.

Figure 7. Aligned original Panoramic image.

Figure 8. 1D Layout.

stand and build my own analytical capabilities, such as geometric and mathematical models. Thank you for the enlightenment that the practice has brought me and my supervisor for the correct guidance.

References

[1] J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6):679–698, 1986.

[2] J. M. Coughlan and A. L. Yuille. Manhattan world: compass direction from a single image by bayesian inference. In Proceedings of the Seventh IEEE International Conference on Computer Vision, volume 2, pages 941–947 vol.2, 1999.

[3] E. Delage, Honglak Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 2418–2428, 2006.

[4] C. Fernandez-Labrador, A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero. Layouts from panoramic images with geometry and deep learning. IEEE Robotics and Automation Letters, 3(4):3153–3160, 2018.

[5] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 1, pages 519–528, 2006.

[6] C. Sun, C. Hsiao, M. Sun, and H. Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1047–1056, 2019.

[7] J. Xu, B. Stenger, T. Kerola, and T. Tung. Pano2cad: Room layout from a single panorama image. In 2017

Figure 9. Wall-Wall boundary.

IEEE Winter Conference on Applications of Computer Vision (WACV), pages 354–362, 2017.

[8] H. Yang and H. Zhang. Efficient 3d room shape recovery from a single panorama. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5422–5430, 2016.

[9] J. Zheng, J. Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. European Conference on Computer Vision (ECCV), 2020.

[10] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2051–2059, 2018.