Geometric Understanding of Indoor Scenes

Moontae Lee
Dept. of Computer Science
Stanford University
moontae@cs.stanford.edu

Abstract

Finding and recognizing objects are basic functions in computer vision which can play fundamental roles in various applications. As indoor scenes have high degree of rigid structures, we study the problem of understanding indoor scenes in terms of geometric clues inherent in each image. Since traditional 2D detectors are highly sensitive to changing of viewpoints and partial occlusion, we argue how 3D approaches can be utilized to improve the performance of object localization and detection. Starting from an assumption that all objects are standing on the floor, we first recover 3D structures to assign a reasonable three dimensional coordinates over the 2D images, and then slide 3D windows having different scales over each spatial area to localize their positions. We next evaluate their visible faces by using Histogram of Oriented Gradients and Self-Similarity Descriptor features, and then learn the classifier to detect sofa, table, and bed. Finally, we report the performance of our classifier and discuss how these efforts can contribute to the problem of holistic scene understanding. For the future works, we will hire richer information considering spatial context and interactions between objects.

1. Introduction

Imagine you are walking to your room in the house at night without using any light sources. All you can see are only small part of wall edges and partial outlines of objects. Under this condition, you probably start to think that 1) walls are running parallel to each other and orthogonal to the floor, 2) objects are standing on the floor and likely to be aligned along walls, 3) the door should be on the wall because of your prior knowledge about the structure of house. As a result, you can easily find your room though you could see almost nothing. The interesting part is that you are able to apply same processes to find objects even in unfamiliar place. This is the power of geometric inference that humans are easily able to do immediately.

While this inference is a natural process for humans, it is not easy for machine to achieve the same information because what machine had is only one single projection image of a real scene. Nevertheless, three assumptions 1) ~ 3) are still reasonable for machine setting because they are physically valid in most indoor contexts. Thus if we can somehow figure out the original 3D structures of each projection image, we are able to infer geometric information which could be largely utilized in finding and recognizing objects.

1.1. Overview

In this paper, we first show that it is available to recover 3D structures from projection images to a certain degree by finding the vanishing points information, and then we will explicitly construct the plausible 3D coordinates for the given image in order to draw the 3D box fitting to room geometry. Since pretty large objects are likely to be aligned along wall directions and other objects along the large ones, once we constructed the plausible 3D coordinates, it means that we can draw reasonable boxes for target objects in many cases.

After 3D reconstruction, we will formally generate a number of 3D windows to slide over the image space. Thus the collection of different 3D sliding windows forms hypothesis space in our setting. To evaluate the target inside the current 3D windows, we will score each visible face by evaluating HOG (Histogram of Oriented Gradients) and SSD (Self-Similarity Descriptor) feature. After scoring each face independently, we will combine them as an unified metric. As HOG and SSD can be efficiently computed on a rectangular region, we here rectify all the visible faces.

However, it is still challenging to find how to improve detection results by hiring spatial contexts similar to what humans normally do through their general knowledge about the indoor structures. To tackle this problem, we will focus on two spatial interactions; interaction with other objects (e.g. relative distance and relative size) and interaction with the room layout (e.g. distance from the bottom centroid to each wall). This is because, for example, a table is more likely to be located in the middle of several sofas or beside beds rather than just being alone aligned along a wall. Therefore we expect that learning a classifier

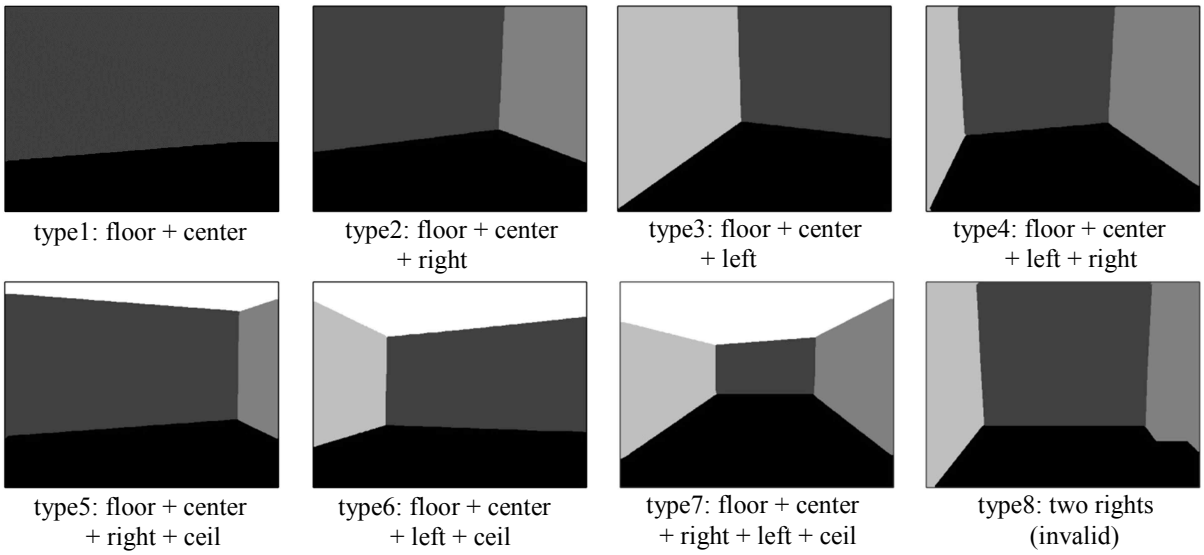

Figure 1: Seven Valid Room Layouts

with hiring these spatial features will cause similar effect with finding some degree of general knowledge like 3), which can be applicable in various other indoor scenes. In other words, our approach is attractive because we can reuse revealed knowledge as a prior for finding plausible hypothesis. It will substantially reduce the size of hypothesis space and contribute to improvement of detection result in the same time.

1.2. Related Work

In our work, finding the plausible 3D geometries inside 2D image is the first task. To accomplish this goal, we are recovering intrinsic and extrinsic camera parameters by using the information of vanishing points. In the early 1980s Harlick [5] studied perspective transformation to analyze 3D location from 2D image. In 1990, as an extension of [5], Wang [6] tried to extract those parameters in terms of vanishing lines formed by a rectangular parallelpiped inside the image. In the early 2000s, Wilczkowiak [7] revisited similar approaches with [6] by using more systematic ways. However, their calibration methods given in [6] and [7] are both based on one particularly shaped object inside the image whereas we estimates vanishing points by collecting all of remarkable straight lines given in the scene. This is the same approach with the recent developments of Wang [2] and Hedau [3].

Thus mainstream of our work is closely connected to two works of Hedau [1] and Wang [2]. Starting from [2]'s room layout detection result (See the above Figure 1), we will accept similar assumptions to [1] in which they modeled real world structures as the set of 'boxes'. They assumed that 1) all objects stand on the floor, 2) each face of objects are parallel to the walls. Due to the first assumption, they could decide the shape of box from fixing one of corners on the floor as a reference point. The second assumption means the orientation of objects are always aligned along the wall direction. Owing to this assumption, they can uniformly rectify each face only through the information of vanishing points. But, realistically, there are many cases where the second assumption is violated whereas the first assumption is largely valid. Therefore, in this paper, we adopt only the first assumption and loose the second assumption by considering rotations on the floor. (It means we are no longer able to rectify visible faces only through vanishing points information)

In addition to accepting previous assumptions, we put a constraint that every wall should be appeared at most once. For instance, type 8 in the above Figure 1 is invalid because of two right walls. Recently, Lee [4] introduced the concept of indoor world model which is the combination of "Manhattan World" and "Single-floor single-ceiling". Here we employ much tighter constraints so as to neatly figure out the spatial interaction with the room layout.

However, in contrast to [1], we are trying to detect more than one object by hiring SSD features as well as HOGs. This is because, under the single target setting, it is difficult to distinguish correct detection from correct localization. Detecting single large object such as bed is suitable only for the previous assumption 2) which is not always true. Furthermore, the bigger target objects are, the harder we can figure out the effectiveness of spatial interactions because the most part of image is filled only by a single object. Therefore we will detect three frequently appeared

indoor objects: sofa, table (or desk), and bed in order that they can reveal the power of general knowledge.

2. 3D Reconstruction

As an initial step, we find vanishing points which implied the information of three orthogonal directions in the world coordinate. This can be done by collecting relatively long lines inside the image, and then find the three dominant intersections, which can be placed outside of the image plane. Since finding vanishing points is not the main purpose of our paper, we will not cover the exact details, which can be easily found in [3].

2.1. Layout Detection

Our paper is starting from Wang [2]'s room layout detection results given in the previous Figure 1. It was done by considering room clutters as latent variables and learning discriminatively through structured SVM. To avoid terminology confusion, we call the room configuration of wall and border as the 'room layout' instead of the 'box layout'. This is because a 'box' represented a 'room' in the paper [2] whereas we defined it as cuboid-like 3D windows to be fitted for target objects.

To neatly sliding 3D windows, the room layout should be one of the seven predefined types given in the Figure 1. (e.g: if there are two right walls like Type 8, then some 3D box near the upper right corner will invade wall areas) Furthermore, distances to each wall will play so fundamental roles as spatial features in future that we should know the exact positions of borders between walls.

Thus we here detected the exact number of global borderlines through both Hough Transform and Shi-Tomasi corner detection, and merge local borderlines into global ones. After getting the proper number of global borderlines, we re-compute every corner by an average intersection of three borders. Finally we recognize characteristics of borderlines (e.g: the border between left and center wall) by checking orthogonal direction around the them in the room layout image. The following figure shows the detected 8 borderlines.

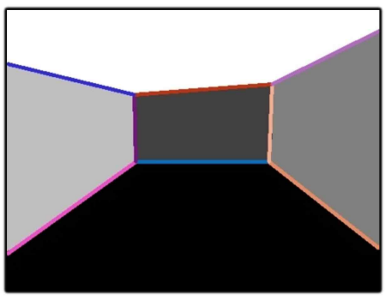

Figure 2 : 8 Detected Borderlines for Type 7 Layout

2.2. Camera Calibration

Now we start to recover the camera parameters for the full 3D reconstruction. Let the p be a 2D homogeneous pixel coordinates and P be the corresponding 3D homogeneous world coordinates. Then projection between two coordinates can be formalized by the following equation.

$$\lambda p = K[R \mid t]P$$

(where [R | t]: *camera extrinsic matrix mapping world coordinate to camera coordinate, K: camera intrinsic matrix mapping camera coordinate to pixel coordinate)*

Since it is possible to arbitrarily fix the origin of world coordinate, we can achieve the natural world coordinate by setting t as a zero vector of no translation. (i.e: world coordinate is centered at the optical center, which is a camera center). If we assumes the zero skew and unit aspect ratio, the projection equation will be given as below.

$$\lambda p = KRP \Rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u0 \\ 0 & f & v0 \\ 0 & 0 & 1 \end{bmatrix} [R_{*1}\ R_{*2}\ R_{*3}] \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

(where f: focal length, the distance between image center and camera center, (u0, v0): image center, the intersection of principal axis and image plane, R_{*i}: i-th column vector of rotation matrix)

From the fact that vanishing points are the projection result corresponding to the limit point of three orthogonal directions in world coordinate, we can get the following relations.

$$\begin{aligned} e_1 = [1\ 0\ 0]^T &\Rightarrow v_1 = KRe_1 \ (i.e\ e_1 = R^{-1}K^{-1}v_1) \\ e_2 = [0\ 1\ 0]^T &\Rightarrow v_2 = KRe_2 \ (i.e\ e_2 = R^{-1}K^{-1}v_2) \\ e_3 = [0\ 0\ 1]^T &\Rightarrow v_3 = KRe_3 \ (i.e\ e_3 = R^{-1}K^{-1}v_3) \end{aligned}$$

(where e_i: i-th Euclidean basis in world coordinate, V_i: i-th vanishing points in pixel coordinate. Since λ can be arbitrarily fixed, we set its value as 1 for convenience)

By an orthogonality of basis directions, we can get the following results.

$$\begin{aligned} 0 = e_1^T e_2 &= (v_1^T K^{-T} R^{-T})(R^{-1}K^{-1}v_2) = v_1^T K^{-T} K^{-1} v_2 \\ 0 = e_2^T e_3 &= (v_2^T K^{-T} R^{-T})(R^{-1}K^{-1}v_3) = v_2^T K^{-T} K^{-1} v_3 \\ 0 = e_1^T e_3 &= (v_1^T K^{-T} R^{-T})(R^{-1}K^{-1}v_3) = v_1^T K^{-T} K^{-1} v_3 \end{aligned}$$

(where R^{-T} and R^{-1} are cancelled each other because every rotation matrix is orthogonal) Since we knew the information of three vanishing points $\{v_1, v_2, v_3\}$, it is available to uniquely determine three intrinsic parameters $\{f, u_0, v_0\}$ as closed form equations through the above three equations.

After recovering the camera intrinsic matrix K, we are

also able to recover the rotation matrix R by the relations we already revealed.

$$Re_1 = R_{*1} \Rightarrow v_1 = KRe_1 = KR_{*1} \Rightarrow R_{*1} = K^{-1}v_1$$
$$Re_2 = R_{*2} \Rightarrow v_2 = KRe_2 = KR_{*2} \Rightarrow R_{*2} = K^{-1}v_2$$
$$Re_3 = R_{*3} \Rightarrow v_3 = KRe_3 = KR_{*3} \Rightarrow R_{*3} = K^{-1}v_3$$

If we merge the above three equations into a single matrix form, we can easily get the rotation matrix R as a closed form equation given in the following.

$$R = [R_{*1}\ R_{*2}\ R_{*3}] = [\mathrm{K}^{-1}\mathrm{v}_1\ \mathrm{K}^{-1}\mathrm{v}_2\ \mathrm{K}^{-1}\mathrm{v}_3] = K^{-1}V$$

(where $V = [\mathrm{v}_1\ \mathrm{v}_2\ \mathrm{v}_3]$, the column-wise collection of three vanishing points vectors)

Up to this point, we recovered all intrinsic and extrinsic parameters by the closed form equations except the translation vector, which was arbitrarily fixed at the camera center. Note that X'Y'-plane for camera coordinate has the same orientation with UV image plane, and world coordinate is centered at the camera center in our natural coordinate of zero translation. See the following Figure 3.

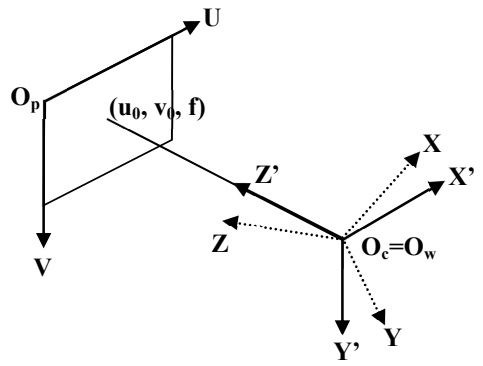

Figure 3 : Three Coordinates Systems

(where UV: pixel coordinate, X'Y'Z: camera coordinate, XYZ: world coordinate, $\{O_p, O_c, O_w\}$: origin of three coordinates respectively, (u_0, v_0, f): the image center at camera coordinate)

Having these calibrated parameters, we can put everything into the unified perspectives. In other words, if we assume the camera height measured from floor is 1, the unit length, world coordinates corresponding to points on the floor in the image plane can be evaluated by the following equation.

$$P = \frac{R^{-1}K^{-1}p}{v_2^T R^{-1}K^{-1}p}$$

(where v_2: the vertical vanishing point, which could be out of image planet) Note that this equation means the inverse transformation from 2D pixel coordinate to 3D world coordinate, and then normalize its height as 1. This is because Y coordinate of any points on the floor should be 1. (See the Figure 3) In after, these results will be utilized in precisely formulating 3D sliding windows with different sizes at different position in the image space.

In case that there are two infinite vanishing points (one horizontal and one vertical), it is impossible to recover the focal length f analytically because it is infinite. To resolve this problem, I manually set the focal length as 2000 and adjust the image center as the third vanishing point.

3. Features

To evaluate each window hypothesis, we need a method to score visible faces inside the given window. In order to score each face uniformly, we rectify all visible faces. This process substantially distinguishes our 3D approaches from the traditional 2D detectors by the fact we can still evaluate other faces even if the front face is occluded by other object.

Since we, in contrast to [1]'s approach, did not assume all objects are precisely aligned along wall directions, we rectify visible faces by using 3D information that we recovered from the previous section, not only through vanishing point directions. We then evaluate 800 dimensional HOG features and 720 dimensional SSD features from each rectified face.

3.1. Image Rectification

Though parallel lines were not parallel anymore in the image plane, they are still parallel in world coordinate. Since all camera parameters that we recovered allowed us to fully reconstruct each image as 3D scene, we are able to find four ideal target points to which original skewed four vertices of each face will be mapped. After then, we are able to compute the homography matrix between pairs of four points. It will provide the rectifying transform matrix.

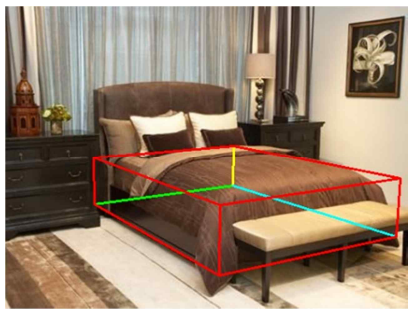

Figure 4 : Three Visible Faces for Rectification

See the above Figure 4. Even if an object is covering the front face of a bed, we can still evaluate the left and upper faces by rectifying them. Figure 5 in the next page is now showing three rectified images and feature extractions.

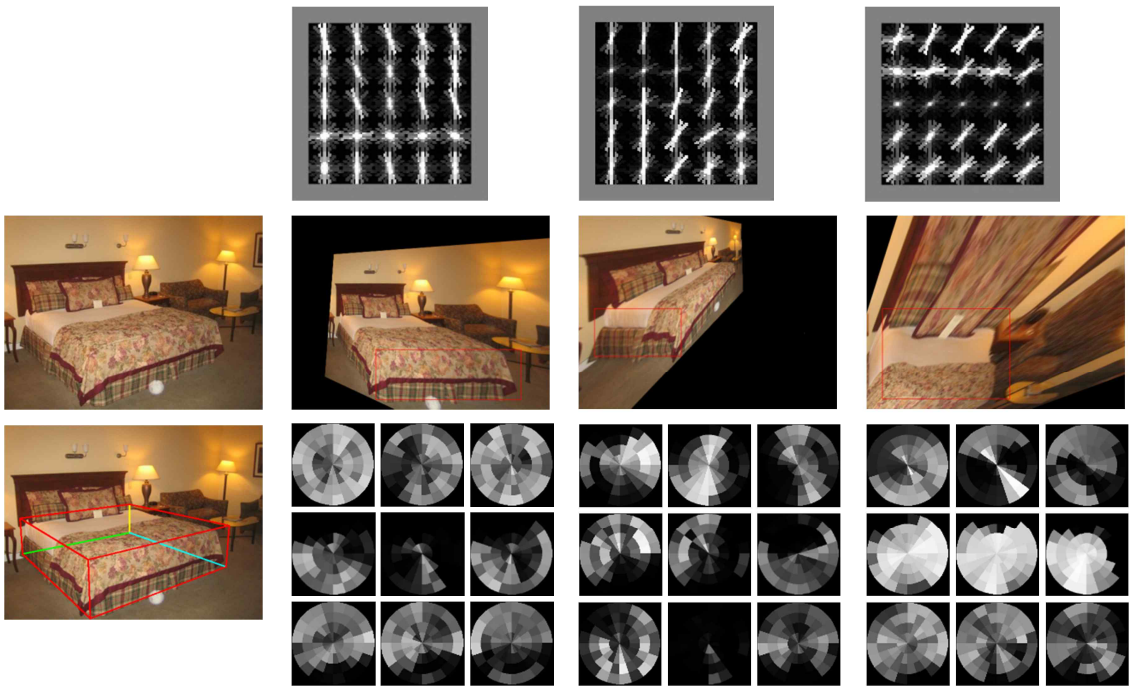

Figure 5 : Image Rectifications and Feature Extractions (1st row : HOG, 3rd row : SSD)

3.2. Feature Extractions

After rectifying visible faces, we resize each rectified fragment to the 87x87 pixels square patch. (If the size of fragment is smaller than the patch, it will be extrapolated to the given size by inter-area interpolation) Then we will first compute the HOG feature by dividing each patch into 5x5 cells, where each cell has 32 orientation bins. Thus the total dimension of HOG features is 5x5x32 = 800. Note that HOG features can be efficiently computed on a rectangular region. This is also the reason why we are rectifying all visible faces.

HOG features are proven as good in the part-based model [8] because it is invariant to the change of lighting and small deformations, and also showed fairly remarkable performance on the previous paper [1]. However, we also introduce SSD features, which are strong local features. This is because many indoor objects are of repeated patterns on their faces. Since we are focusing on both localization and detections of three different objects, introducing SSD features is reasonable for better detection. To compute the SSD feature, we will follow the same path suggested in the paper [9]. Here we computed SSD 9 times as intervals of 20 pixels by sliding 40 pixels small patches horizontally and vertically. Each sliding will evaluate 20 orientations with different four different radii. As a result, we acquire 20x4x9 = 720 dimensional SSD features.

The above Figure 5 illustrates all features that we extracted from each face. As we can observe in the first row, HOG grasp how strong each gradient direction is at the given point in the image, whereas SSD catches the strong orientation having similar shapes. (White area means strong similarity toward that direction)

4. Learning and Analysis

In previous discussion, we divided our 3D sliding box into the set of visible faces. Thus we trained the classifier per each face and combine them to conclude the final decision. We used 1-vs-all SVM with linear kernel using the following number of positive and negative images.

	face	# of positive samples	# of negative samples
bed	0	78	2316
	1	78	2265
	2	69	1582
table	0	150	2244
	1	150	2193
	2	133	1518
sofa	0	103	2291
	1	100	2243
	2	81	1570

Figure 6 : The Number of Positive & Negative Samples

Note that we used 1-vs-all SVM, which means all other images are included as negative samples when we are training a classifier for one object. The following table is the training result.

	face	SV	precision	recall
bed	0	388	1.0000	1.0000
	1	336	1.0000	1.0000
	2	289	1.0000	1.0000
table	0	522	1.0000	1.0000
	1	546	1.0000	0.9995
	2	431	1.0000	1.0000
sofa	0	411	1.0000	1.0000
	1	477	1.0000	0.9995
	2	381	1.0000	1.0000

Figure 7 : The Training Result (per face)

As shown in the above table, the classifier achieves the abnormal level of both precision and recall. It means there are neither false positives nor false negatives. This is mainly because we don't have enough number of positive samples. Based upon a visual confirmation, the current classifier denies almost all hypotheses if they are not tightly fit like manually constructed ground-truth bounding boxes. See the following testing result.

	face	SV	precision	recall
bed	0	388	0.9760	0.9607
	1	336	0.9875	0.9596
	2	289	0.9636	0.9408
table	0	522	0.8992	0.9872
	1	546	0.9516	0.9874
	2	431	0.9186	0.9813
sofa	0	411	0.9804	0.9804
	1	477	0.9674	0.9794
	2	381	0.9593	0.9821

Figure 7 : The Testing Result (per face)

Though the entire precision and recall are slightly decreased compared to the previous training result, they are still too very high. This is because it is very rare to encounter the right bounding box in current hypothesis space, which is the collection of randomly generated bounding box. In other words, the above testing result is almost same with only-denying classifier.

However it does not mean that the 3D sliding structure is meaningless. If a random hypothesis is luckily good enough such like what humans drew, the classifier quite surprisingly well localize the bounding box and detect the object correctly. But, the problem is that most hypotheses are negative sample even though we generated them very carefully by fixing one point on the bottom.

(Note that, since the result is too high, we did not try the other kernel such as polynomial or RBF)

5. Visual Result

Though we could not go further for lack of positive samples, here we illustrate the good detection result. Based on some visual confirmation, we firmly believe if we introduce more positive images, then the performance of classifier beats traditional 2D detectors.

(Note that all the above detection results were achieved by generating more than 1000 hypothesis per each image)

6. Conclusion and Future Works

Using geometric information for object detections are very promising in terms of its tight precision and ability to evaluate not only the front face but also other visible faces. In near future, we should first try more positive samples. At the same time, figuring out how to make good hypothesis is the most urgent issue to utilize this setting. After Then we can introduce spatial features that we originally designed.

References

[1] V. Hedau, D. Hoeim, and D. Forsyth. Thinking Inside the Box [ECCV2010]

[2] H. Wang, S. Gould, D. Koller. Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding [ECCV2010]

[3] V.Hedau, D. Hoeim, and D. Forsyth. Recovering the Spatial Layout of Cluttered Rooms [ICCV2009]

[4] D. Lee, M. Hebert, T. Kanade. Geometric Reasoning for Single Image Structure Recovery

[5] R. Haralick. Using Perspective Transformation in Scene Analysis [Computer Graphics and Image Processing 1980]

[6] L. Wang, W. Tsai. Computing Camera Parameters Using Vanishing-Line Information from a Rectangular Parallelepiped. [Machine Vision and Applications 1990]

[7] M. Wilczkowiak, E. Boyer, P. Sturn. Camera Calibration and 3D Reconstruction from Single Images Using Parallelpipeds

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part Based Models. [PAMI 99, 2009]

[9] E. Shechtman, M. Irani, Matching Local Self-Similarities across Images and Video [CVPR 2007]

Acknowledgement

I started this project as the part of CS294a under the supervision of Professor Daphne Koller. However, unfortunately, I have been struggled most parts of my project by myself because this is the first trial of geometric reconstruction. Since I now established all the basic setting, it will become worth topic to research. Thanks for this quarter and valuable advice.