Stereoscopic 3D Image by Single Image

Po-Shien Wang [B99902006 NTU CSIE], Li Wang [B99902078 NTU CSIE],
Chun-Hao Lai [B99902075 NTU CSIE] and Jing-Yeu Chen [B99902055 NTU CSIE]

I. Motivation

Recently, because of the popularity of the 3D movies, there are many old famous movies being reconstructed as the 3D one, such as The Lion King, Titanic and Star wars. But, actually, the work to change a 2D movie to a 3D one is time-consuming and needs many human resources, so we want to find a way to reduce it and even make it automatically done by the machines. And considering what we have learned in class are the works on only one image, so we come up this topic that how to construct a 3D Image from a single plain image.

II. Introduction

First, we do qualitative research on the image to know the component in the image and classify the image into indoor, outdoor with geometric appearance and outdoor. And thus, we can find out the information of the image such as sky and floor and generate a depth map for the sky is the farthest. However, the depth map is not good enough and very rough. So, second, we find the vanishing point in the image and generate another depth map from the vanishing point according to human eyes smoothly. And finally, by combining the two depth map, we construct the final stereoscopic 3D image.

III. Implementation

III.i. Qualitative depth map generation

A. *Color Segmentation*

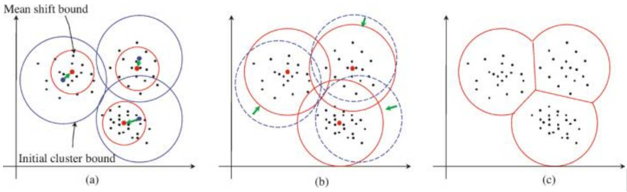
In order to classify the image into indoor, outdoor with geometric appearance, outdoor, the first step is applying the mean shift as the classification algorithm. “Mean shift is a non-parametric feature-space analysis technique”**[1]**. Since this property, Mean shift is often used to cases which hard to evaluate the cluster number like clustering.

Briefly saying, Mean shift treats the points in the feature space as a probability density function. A dense region (cluster) is space that corresponds to local maxima. So the intuitive approach is to perform gradient ascent on the local point until convergence. A simple implementation of this algorithm is the continuously updating the center point to the mean value from all the points in near region until convergence. And all points converge into the same center are regarded as the same group.

A general equation of mean shift is

$$\mathrm{m(x)} = \frac{\Sigma_{\mathrm{x_i \in N(x)}} \mathrm{K(x_i - x) x_i}}{\Sigma_{\mathrm{x_i \in N(x)}} \mathrm{K(x_i - x)}}$$

where $\mathrm{N(x)}$ is the neighborhood of x, and K is the kernel function in Mean shift that estimate the distance between $\mathrm{x_i}$ and x. Typically, Gaussian kernel on the distance to the current estimate is used, where $\mathrm{K}(\mathrm{x_i} - x) = \mathrm{e}^{-c||x_i - x||^2}$.



[2]

Complete image in VIII. Appendix

To build a color-based image segmentation, a clustering method that could identify the chromatically homogeneous regions is needed. Mean Shift algorithm is selected since it can group pixel together without knowing the K. For later classification to indoor/outdoor, mean shift with under segmentation is chosen. By the under segmentation mean shift algorithm, only the predominant color will be extract as the final group, besides, the edge of predominant color will also form the predominant edge. In addition, under segmentation mean shift also reduces the wide color range in the image into a small range of color which is about 10 different colors.

B. *Semantic region detection*

After the color segmentation, what we need to do is to generate the qualitative depth map and classify the image so that we can get a better fusion result. In the paper we find 6 regions to classify each pixel, which are presented in the following table.

Regions	**Labels**
Sky	s
Farthest Mountain	m
Far Mountain	m
Near Mountain	m
Land	l
Other	x

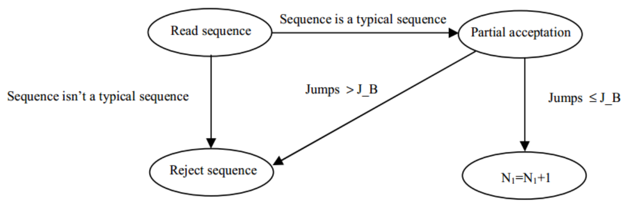
[6]

The task is pretty tricky since it’s obviously not robust to classify a pixel based on only its value (RGB or HIS or any other color space representation). Thus we use a machine learning way (svm) to classify the pixels. This method is adaptable since you can re-train the data by adding some new labeled pixels to get a better result.

The next step is consistency verification, which is to rule out some false-classified regions. We scan the image column by column first to get some sequences of regions (i.e. if we scan the column “112233”, and 1 is classified as sky, 2, 3 are classified as far/near mountain, then we get the region sequence “sm”), and then check if any of region’s vertical size is bigger than a fixed threshold, if so, a false region is detected so we re-classify it to its neighboring region. The assumption is the image we get is taken vertically. This method has a problem that the image after processing will have some vertically stripes on it caused by the column (vertical) scanning. This method may be improved by something related to connected-component labeling, which we leave it to the future work.

Finally we can classify the image. After all these complicated pre-processing, the image data is more consistent and most of the noised points are eliminated. Now we can classify the image to one of the three kinds: outdoor, outdoor with geometric appearance, and indoor. The basic idea is as the following:

Using the new region sequences, the main steps of the algorithm are:



[6]

Complete image in VIII. Appendix

1. Sequences and jumps detection for each sample column: a jump is the number of regions encountered in the examined column.
2. Each sequence is compared to the set of typical sequences. If the sequence is recognized and the jumps number is smaller than a threshold J_B, then the value N1 is increased, where N1 represents the number of accepted sequences. If the sequence isn't a typical landscape sequence or if the jumps number is bigger than J_B then the sequence is rejected.
3. Final classification. The image is classified as Outdoor if the value of N1 is bigger than R1·N, where N is the number of analyzed sequences and R1 is a threshold in [0,1]. Otherwise if the number of sequences with the first region Sky is bigger than R2·N, where R2 is another threshold in [0,1] the image is classified as Outdoor with geometric appearance else it is classified as Indoor.

Note that in this algorithm we don't take all sequences but randomly choose some sample columns to get the examined sequences.

The most important part of this algorithm is: how can we tell if a sequence is a "typical sequence"? The paper gives some examples for typical sequences like "sm", "sl", and etc. But it's not very effective to apply in this algorithm since that we can't name all typical sequence, so I came out with another method: calculating the ratio of "other" regions in a sequence to see if the ratio is bigger than a fixed threshold (in my test, 0.4 is good in the classification), if so, then consider the sequence a non-typical sequence (rejected).

Other thresholds choosing in this algorithm like J_B, R1, R2 are taken from the paper (J_B = 10, R1 = 0.8, R2 = 0.2). The threshold choosing is also a critical part but we already have good accuracy with these fixed thresholds, so we leave it to the future work, too.

After the processing, we get a classified grey-level image, which is the qualitative depth map.

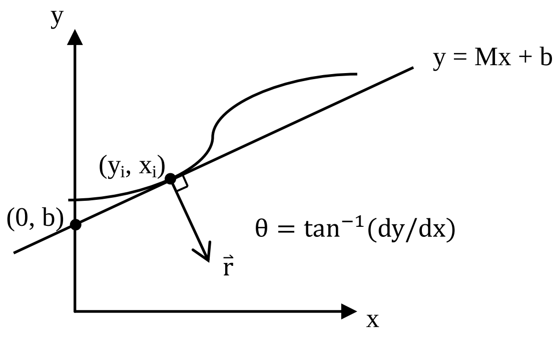
III.ii. Vanishing lines detection

A. Edge Detection

Use Sobel mask to do edge detection.

$$dx = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, dy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

B. Find the line through each edge point (yi, xi)

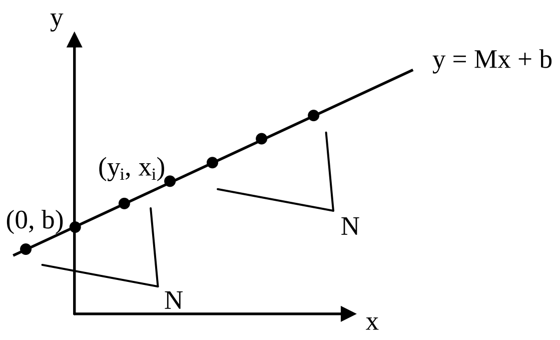


Use the model y = Mx + b

And thus, we can use (M, b) to represent a line. Besides, we know that the direction of the vector $\vec{r}$ will be perpendicular to the slope of the edge, so we can compute M and b by the following equations.

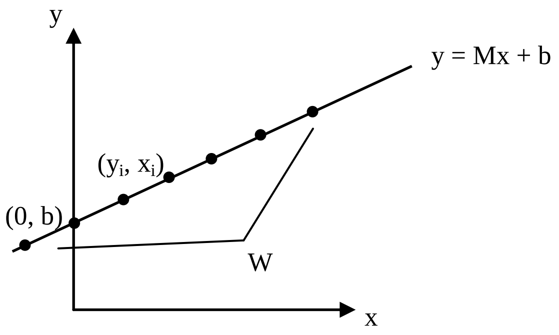
M = $-dx/dy$, b = (y – Mx)

C. Find the straight lines in the image



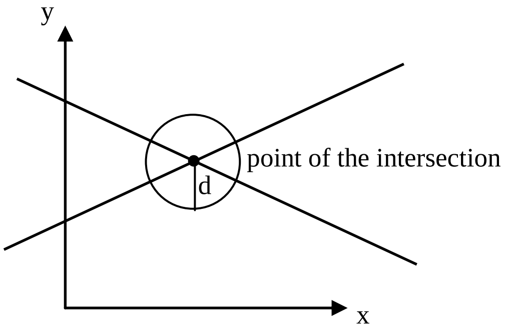
After finding the line through each edge point, we need to know if the line is a straight in the image. We walk along the line through the edge point to see if the N front and N behind points have the same slope M. And if there are N points having the same slope M, we will use the line to find the vanishing point.

D. Weight each straight line



As we know, we will find many edge points through the same straight line, so we give the number W of edge points thought the line as the weight of the line.

E. *Find the vanishing point*



Use the straight lines with weight W > K to find the vanishing point.

We compute the point of the intersection of each two lines, and compute the lines near the point in the circle with radius d. Finally, the point with most lines near it is the vanishing point we find and the lines near the vanishing point are the vanishing lines.

III.iii. Gradient Depth Map Generation

After the position of the vanishing point and the vanishing lines are analyzed, we need to generate the depth gradient, which will be referred when generating the final depth map. The first step of the depth gradient generation is to find the gradient plane, that is, to find the ceiling, the floor, the left wall, and the right wall. The second step is the depth gradient assignment, which generates the depth effect of the four planes.

A. *Gradient Plane Generation*

The position of the vanishing point may be the following five cases: Left Case, Right Case, Up Case, Down Case, and Inside Case, which is shown in Figure 1.

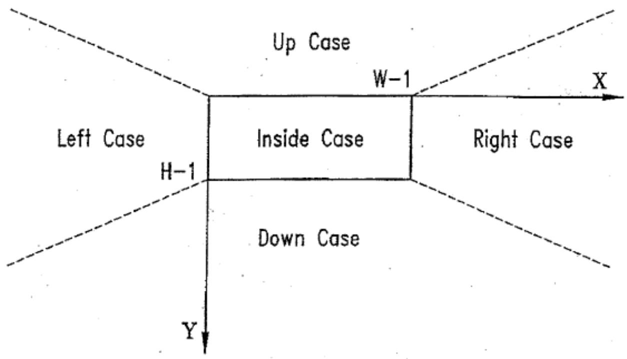


Figure 1 – Five cases[4]

To generate the four planes, we need to classify all the vanishing lines into four classes. The classification is based on two lines, the horizontal and vertical lines that cross the vanishing point. The coordinate system then is divided into four sections by the lines. Vanishing lines located in the corresponding section will be label, shown in Figure 2. For each class, we need to choose one line from all the lines that belong to that class. That is, the result will be four lines, and each line is from a class. Using a set of heuristics, we will decide which lines will be chosen. To simplify the explanation, the following part will focus on the Left Case, shown in Figure 3, which is more general.

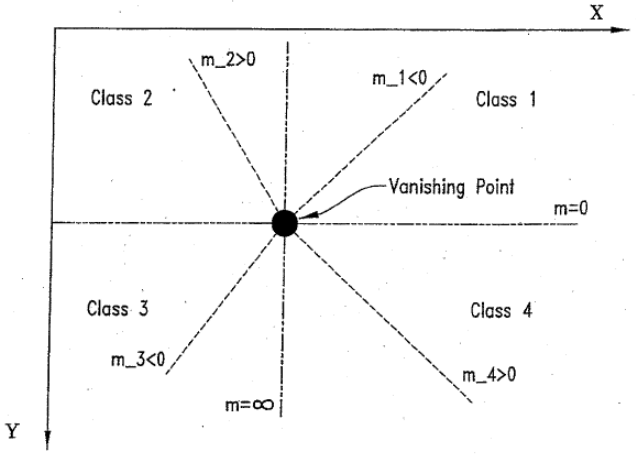


Figure 2 – Four classes[4]

Obviously, all the vanishing lines are only classified as class 1 and 4 in the Left Case. The line chosen for class 1 is assigned as the line that has the smallest slope. The line chosen in class 4 depends on all the lines in both class 1 and 4. If all the slopes of the lines are smaller than a threshold 0.2, then the chosen line is assigned as the line that has the biggest slope; if all the slopes of the lines are bigger than 0.2, then the chosen line is assigned as the line which has the smallest slope; if some slopes are greater and some are less than 0.2, then the chosen line is assigned as the line that has the slope which is closest to 0.2. Finally, two lines are found for the Left Case. Methods for other cases can be found in **[4]**.

The area between the lines from class 1 and 4 becomes the right wall; area above the lines from class 1 and 2 becomes the ceiling; area between the lines from class 2 and 3 becomes the left wall; and the rest becomes the floor. In the Left Case, we can only find ceiling, right wall, and floor.

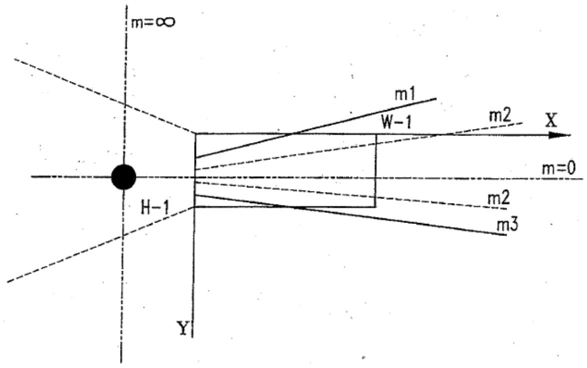


Figure 3 – Left Case[4]

B. *Depth Gradient Assignment*

Let’s analyze the position of vanishing point again. If the vanishing point is on the left of the rightist side of the image, then we apply the following piece-wise linear function, which is the depth level assignment, shown in Figure 4. For the right wall portion, the pixels which are closer to the vanishing point along the x-direction will be blacker. Since humans are more sensitive to the depth variation of the close objects than farther objects, the slope of the depth level increases as it come closer to the vanishing point. Similarly, we can apply this depth level assignment on the left wall by just changing the direction. For the ceiling and floor, the same method is applied by noticing that we consider the y-direction this time.

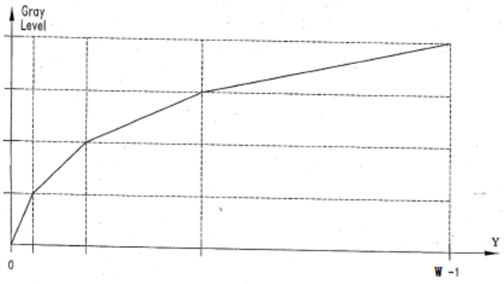


Figure 4 – Depth level assignment[4]

III.iv. Combine the depth maps

In this step, the qualitative depth map from color segmentation and geometric depth map are fused together based on the indoor/outdoor with geometric/outdoor classification. To simplify the equation, we let qualitative depth map as M1, geometric depth map as M2, and the final depth Map as M. The fusion principle is as below **[3]**:

1. Indoor case:

$$\mathrm{M}(\mathrm{x},\mathrm{y}) = \mathrm{M2}(\mathrm{x},\mathrm{y})\forall(x,y)$$

2. Outdoor with geometric case:

$$\mathrm{M}(\mathrm{x},\mathrm{y}) = \mathrm{M1}(\mathrm{x},\mathrm{y})\forall(\mathrm{x},\mathrm{y}) \in \mathrm{Sky}$$
$$\mathrm{M}(\mathrm{x},\mathrm{y}) = \mathrm{M2}(\mathrm{x},\mathrm{y})\forall(\mathrm{x},\mathrm{y}) \notin \mathrm{Sky}$$

3. Outdoor:

$$\mathrm{M}(\mathrm{x},\mathrm{y}) = \mathrm{M2}(\mathrm{x},\mathrm{y})\forall(\mathrm{x},\mathrm{y}) \in \text{Land and } \forall(\mathrm{x},\mathrm{y}) \in \mathrm{Other}$$
$$\mathrm{M}(\mathrm{x},\mathrm{y}) = \mathrm{M1}(\mathrm{x},\mathrm{y})\forall(\mathrm{x},\mathrm{y}) \notin \text{Land and } \forall(\mathrm{x},\mathrm{y}) \notin \mathrm{Other}$$

III.v. Stereoscopic Pair Image Generation

Human brains are brilliant in analyze the left eye view and the right eye view to reconstruct the stereoscopic world. So the final step is to generate the left and the right view, and fuse the two images into one by keeping different colors.

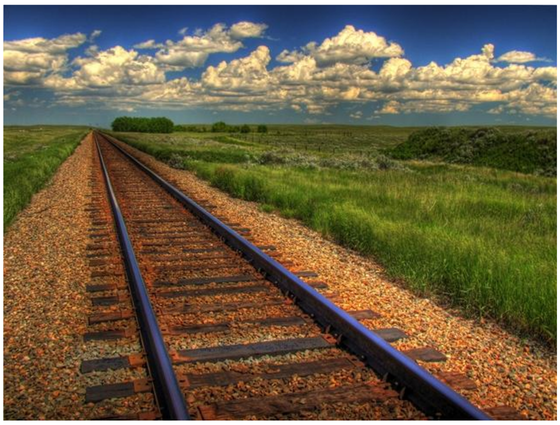
The depth map and the original image determine the shifting amount of each pixel to build the left and right views. For the objects which are farther to the observer, a greater shift is applied, the objects closer to the observer applies a less shift. Written in formula will be like equation (1). This way, we produce a parallax for left and right view.

$$\mathrm{parallax} = \mathrm{M} \times (1 - \frac{depth\_value}{255}) \qquad (1)$$

Different from **[3]**, several values of max parallax M are tested by us. An optimal value 20 is determined when the image is shown in laptop and 14 in projector.

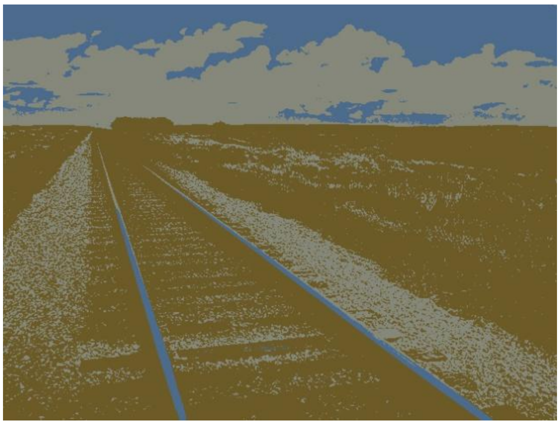
IV. Experimental Results

IV.i. Original images

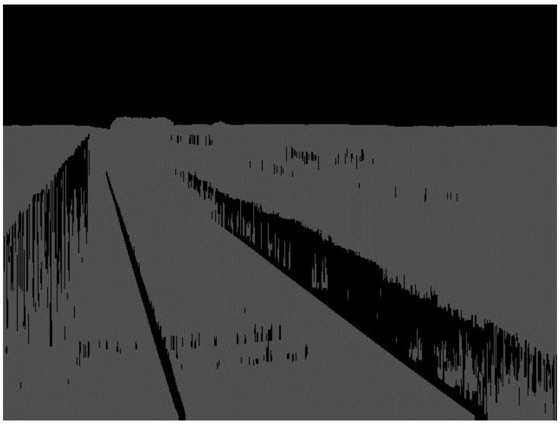


IV.ii. Qualitative depth map generation

A. Color Segmentation

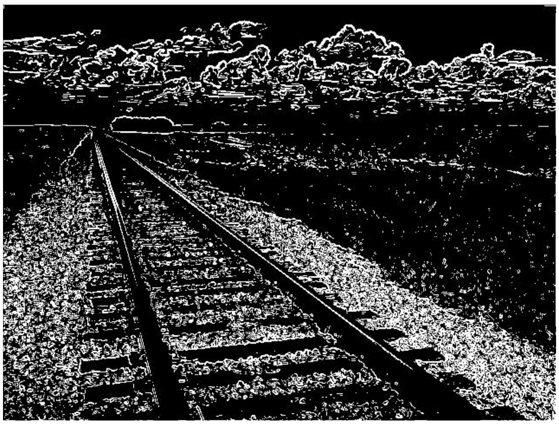


B. Qualitative depth map (After Semantic Region Detection)

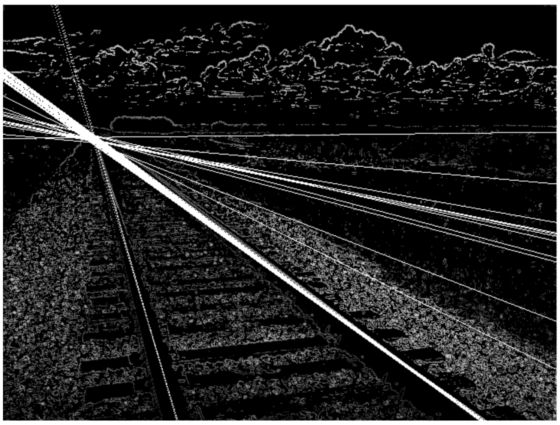


IV.iii. Vanishing lines detection

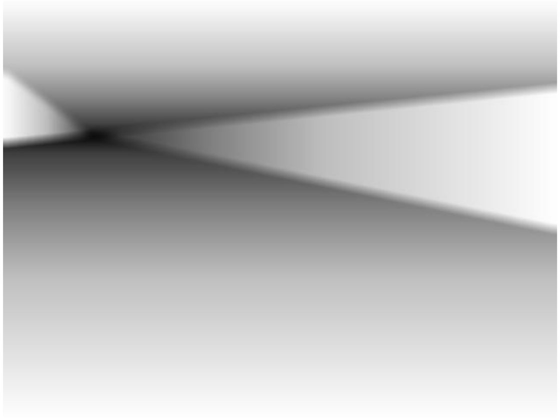
A. Sobel edge detection



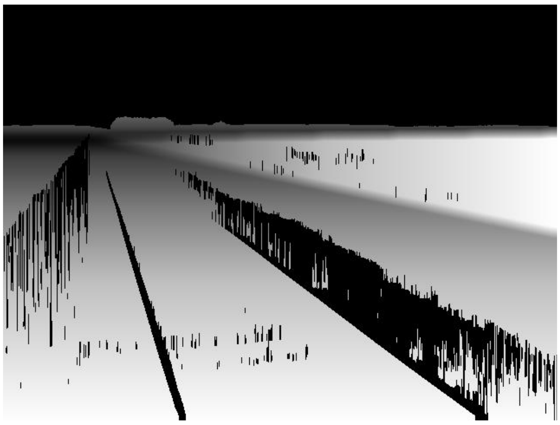
B. Vanishing lines



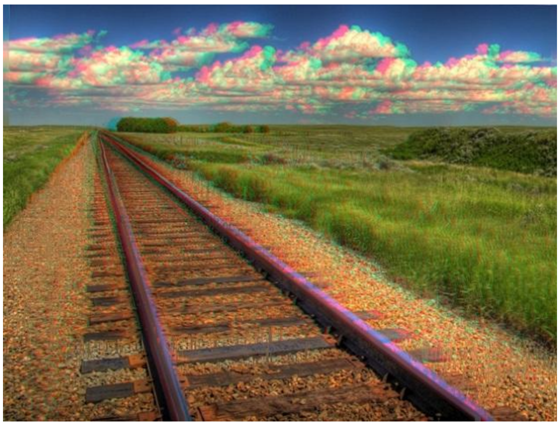
IV.iv. Gradient Depth Map Generation



IV.v. Combine the depth maps



IV.vi. Stereoscopic Pair Image Generation



V. Conclusion

After doing many experiments on many images, we find that the way in our research is not very robust. And thus, the final depth may be also not suitable for human eyes, which is not the same as the actual depth in the reality. And the reasons are the following:

V.i. Qualitative depth map generation

A. Color Segmentation

Even though the Mean shift is much robust to cluster the each pixel, when the picture is too complicate and has many components, the segmentation results will not be good. For example, the sky in our results is colored into two parts because of the cloud. And some parts of the floor are colored the same as the cloud. So we do consistency verification before to generate the final qualitative depth map to eliminate the effect of false color segmentation, which is very effective but there are still something missed.

B. Semantic region detection

Because of the size of our training data is small, the training results are not very great. Many of our outcomes are classified by the human eyes. So, we think that we should use more information such as features not just only the color to train our machine.

How to choose the threshold to classify the images is not clear. The threshold we use now may be just suitable for out testing data. And when working on a new image, something out of control may happen.

V.ii. Vanishing lines detection

It is obvious that there are many parameters to set to find the vanishing lines.

- The bound of (M, b) to see the lines as the same line. [default 0.5]
- The N to detect the straight lines. [default 15]
- The weight K to decide which straight line is important. [default 5]
- The d to find straight line near the point of the intersection. [default 5]

And thus, in some cases, the algorithm works not well. And actually, the algorithm works well only for the image containing components with lines, such as buildings, rail, buses and etc. When the image is very complicate and contains only curves or many lines with many different intersections such as a chess board, it is hard to find the vanishing point.

V.iii. Gradient Depth Map Generation

A. Gradient Plane Generation

The method of using heuristics to detect the gradient planes still fails to find the right planes in some cases. This may due to the lack of vanishing lines acquired. For example, if an Inside Case has no vanishing line in class 1, then we will end up guessing the line for class 1. The method still falls behind from robustness. Combine the result of region detection with vanishing point may improve the result.

B. Depth Gradient Assignment

Gradient depth map may sometimes ends up having obvious boarder between different planes. It is still an issue about letting the objects that have the same distance to the observer assigned to the same depth value.

V.iv. Stereoscopic Pair Image Generation

Viewers may stand in difference distance from the screen. So the max parallax value M must be tuned for every single circumstance. Therefore, generation of stereoscopic pair images still involves a little bit manual work when the tuning of M is necessary.

VI. References

[1] http://en.wikipedia.org/wiki/Mean-shift

[2] http://opticalengineering.spiedigitallibrary.org/data/Journals/OPTICE/23430/057205_1_5.png

[3] S. Battiato, A. Capra, S. Curti, and M. La Cascia, "3D stereoscopic image pairs by depth-map generation", Second International Symposium on 3D Data Processing, Visualization and Transmission, pp. 124-131, 2004.

[4] Method of obtaining a depth map from a digital image, 2005.

[5] D. Comaniciu, P. Meer, “Robust Analysis of Feature Spaces: Color Image Segmentation”, In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 750-755, June 1997.

[6] S. Battiato, S. Curti, M. La Cascia, E. Scordato, M. Tortora, “Depth Map Generation By Image Classification”, SPIE IS&T/SPIE's 16th Annual Symposium on Electronic Imaging 2004.

VII. Division of labor

[1] Po-Shien Wang [B99902006 NTU CSIE]

- Color Segmentation
- Combine the depth maps
- Write the report
- Combine all the codes to execute right and together.

[2] Li Wang [B99902078 NTU CSIE]

- Semantic region detection
- Write the report

[3] Chun-Hao Lai [B99902075 NTU CSIE]

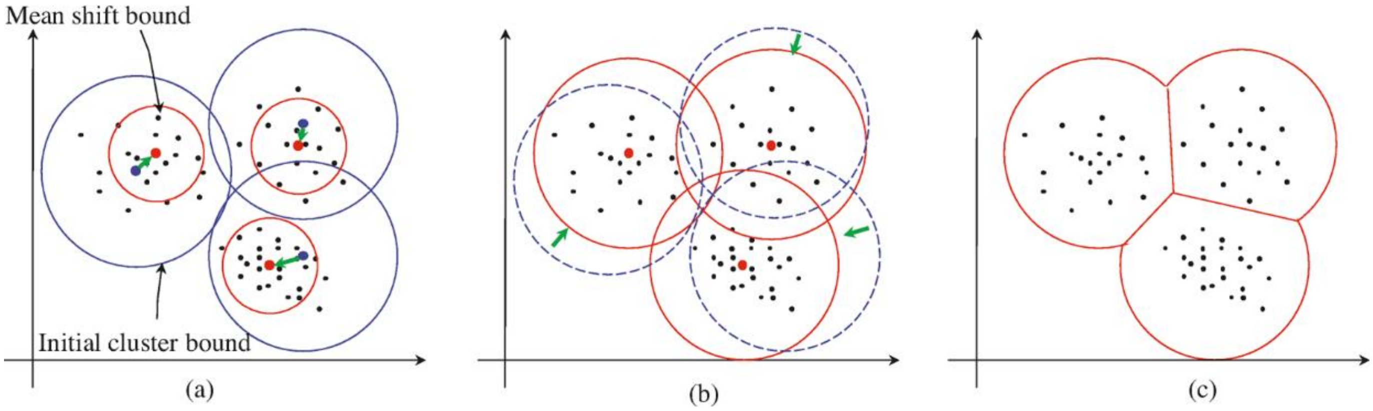
- Vanishing lines detection
- Write the report
- Combine the reports

[4] Jing-Yeu Chen [B99902055 NTU CSIE]

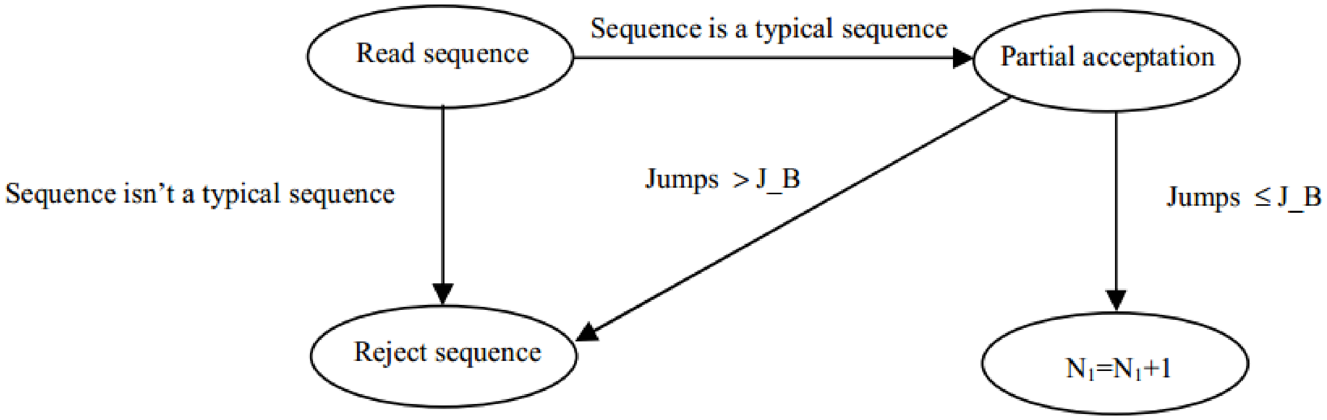
- Gradient Depth Map Generation
- Stereoscopic Pair Image Generation
- Write the report

VIII. Appendix

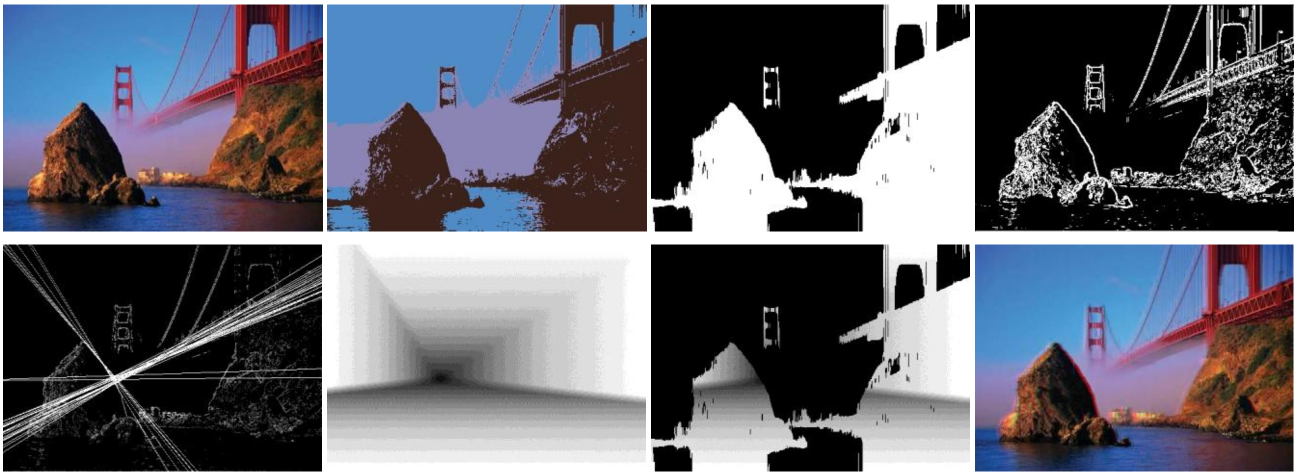
[1]



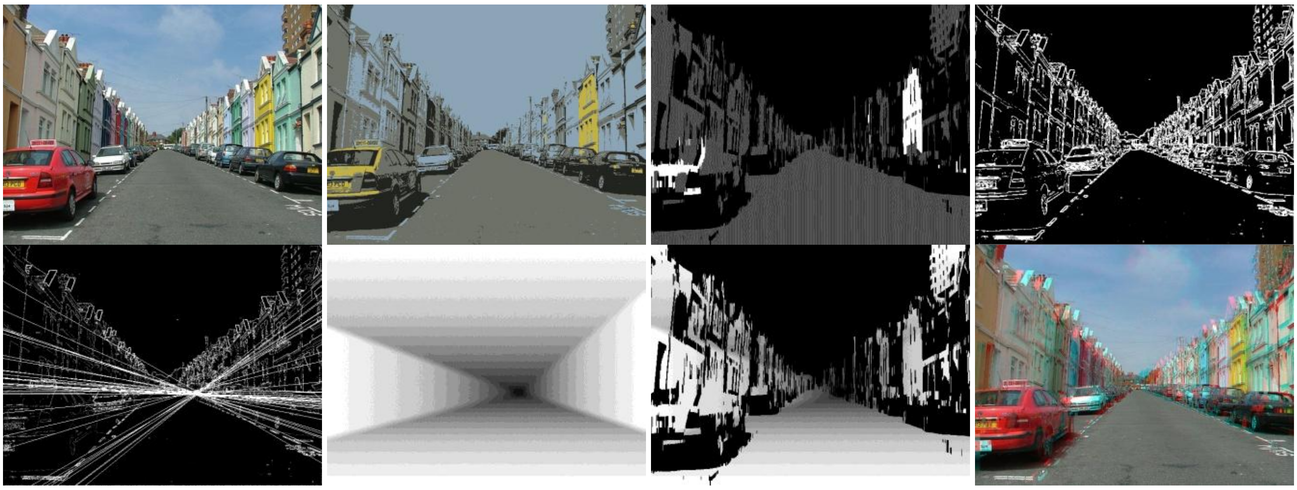
[2]



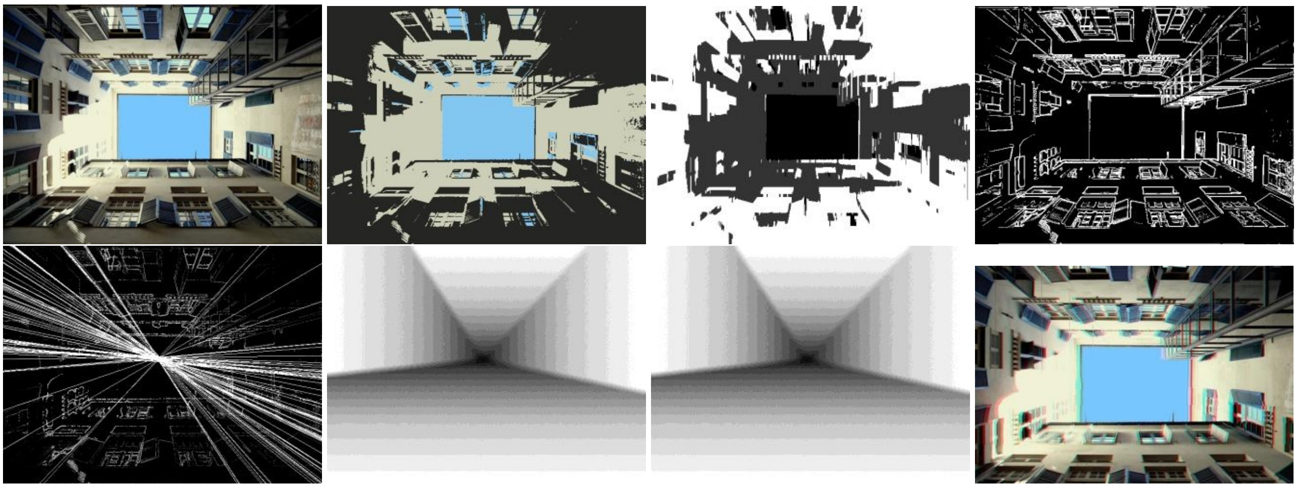
IX. Gallery



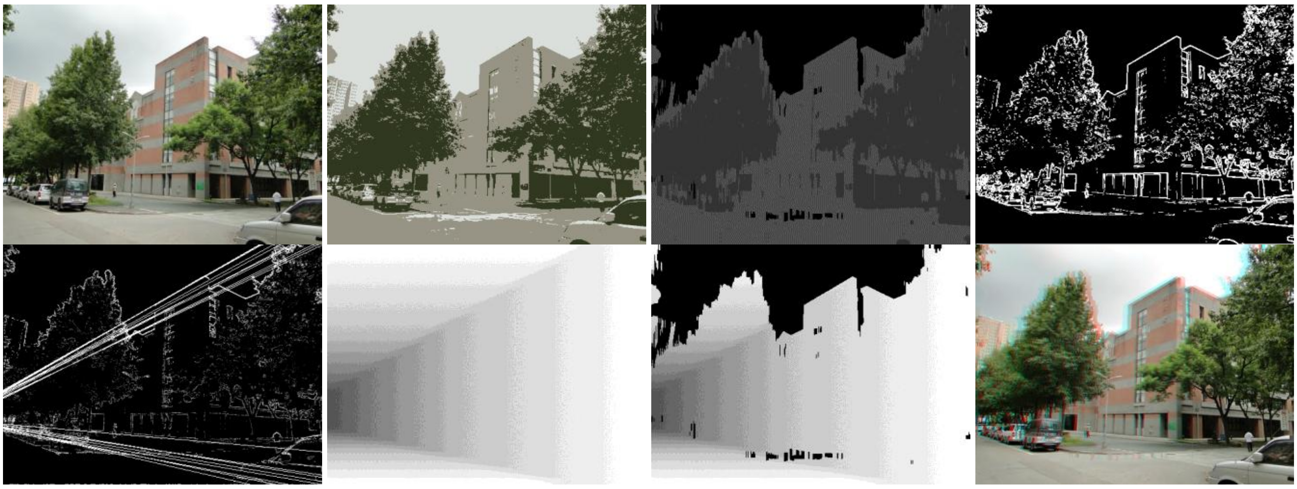
[1]



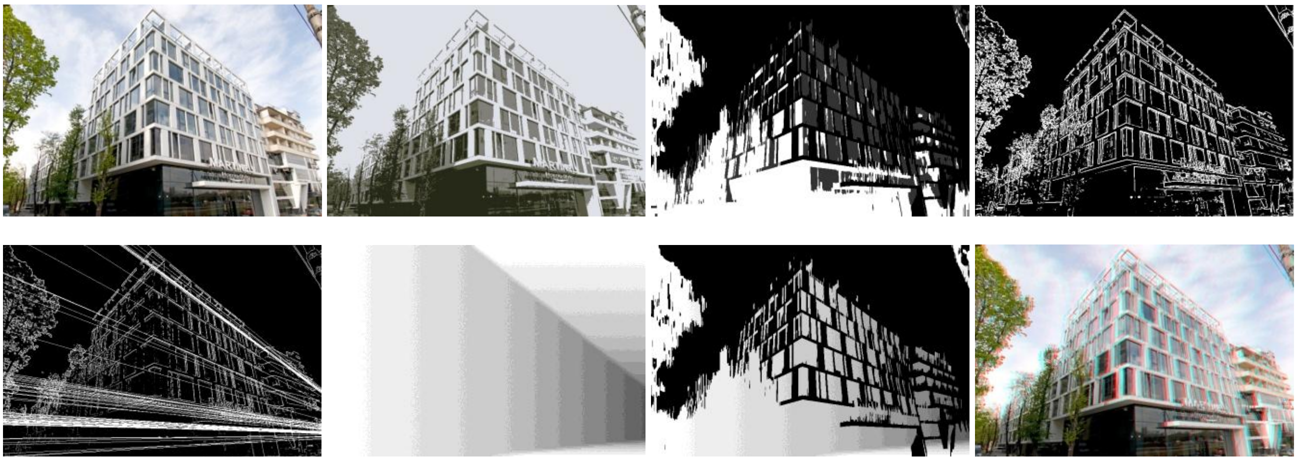
[2]



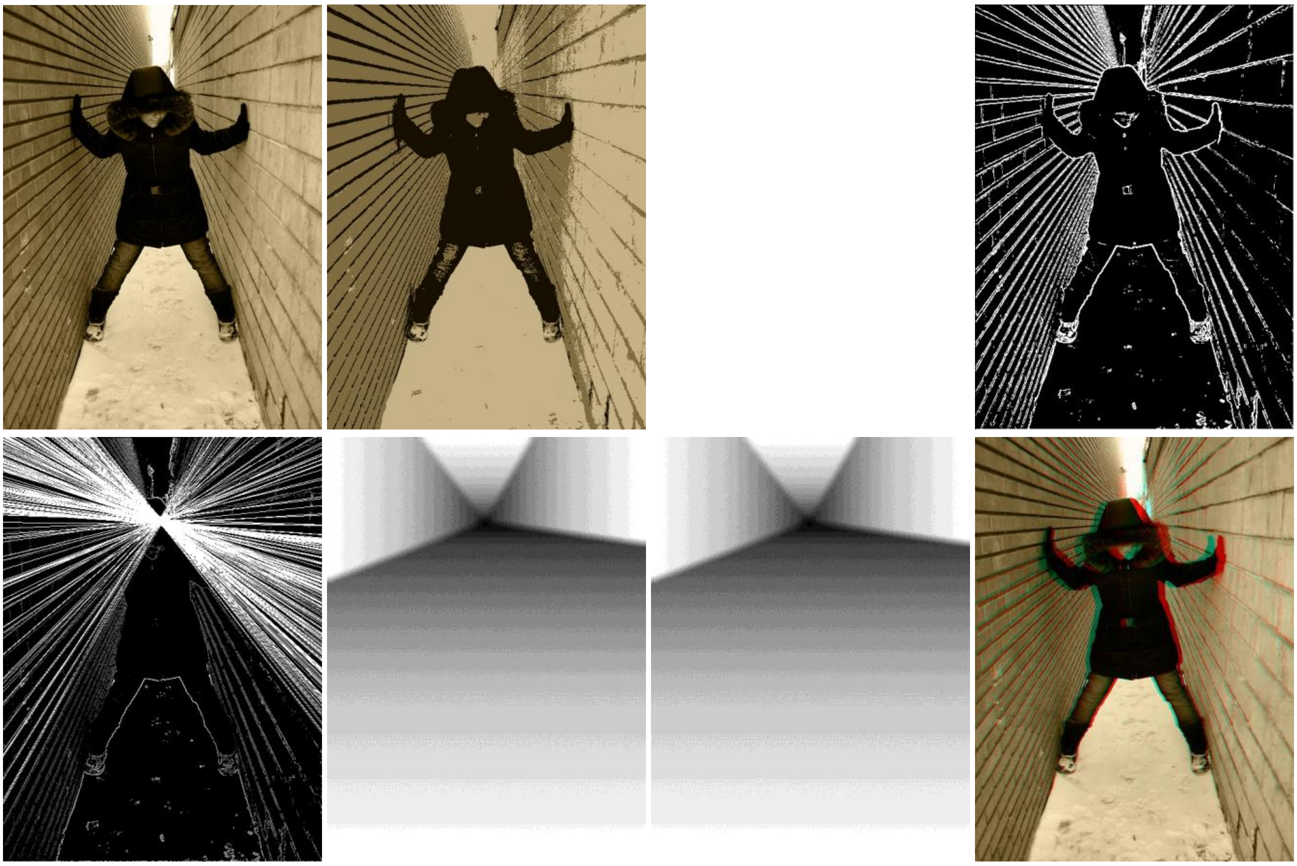
[3]



[4]

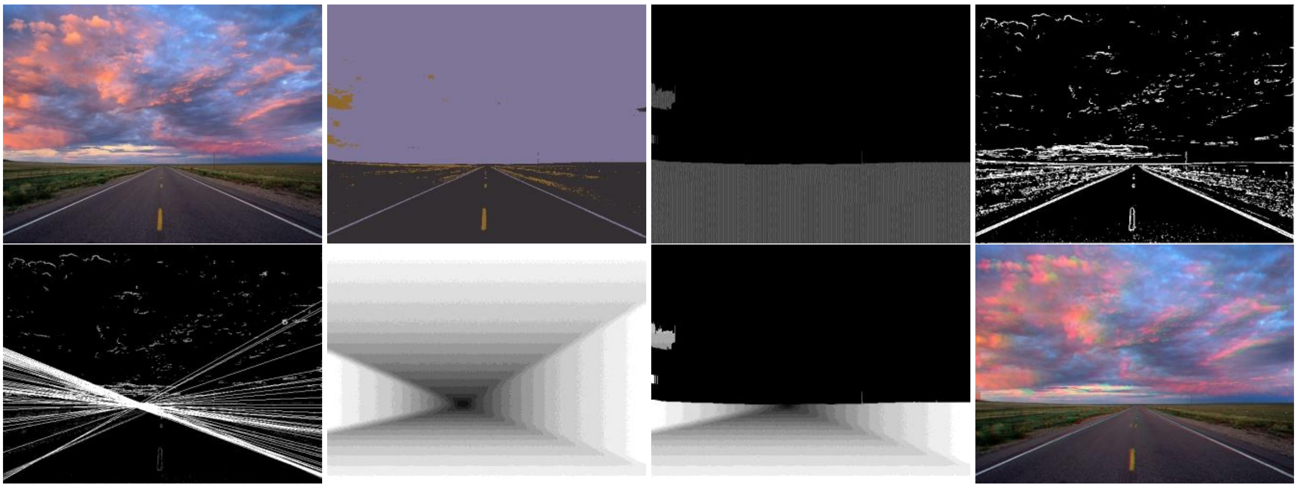


[5]

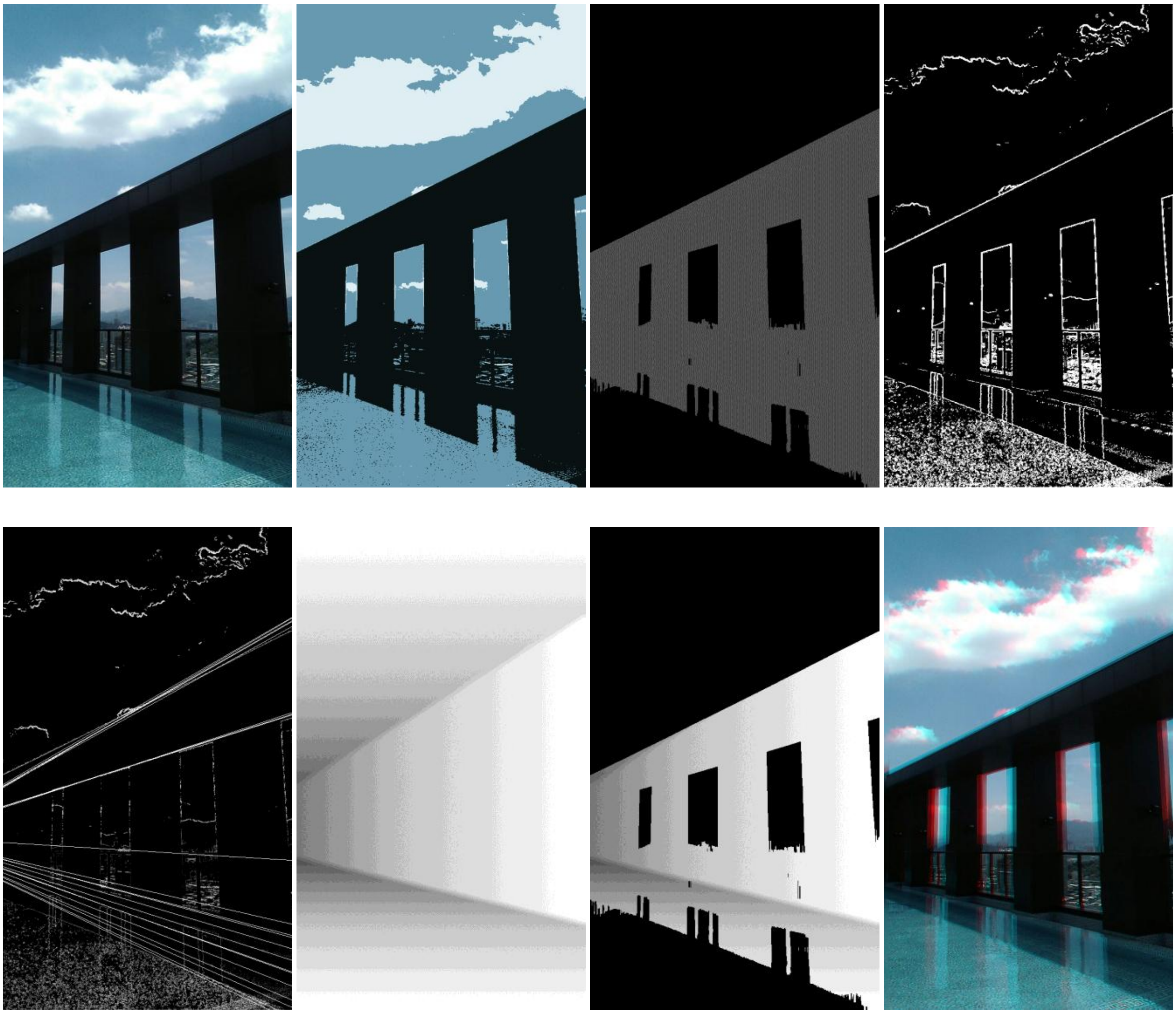


[6]

[7]



[8]



[9]