Assignment 2-Stereo Vision Algorithms

Lei Liu, 9669373

May 5th, 2016

1. Assignment aims

The main objective of this assignment is to build a 3D structure of an industrial scene from its 2D images from two views. The given resource was a pair of stereo images (left rectified image and right rectified image respectively) of the same industrial scene. From these two images, the correspondences should be matched firstly, and then calculate the scene coordinate (X, Y, Z) for a 3D image (Galata, 2016).

The key processes for achieving this goal are below:

1) Load a pair of stereo images in grey level.
2) Find interesting points for both images by an edge detector.
3) Match those interesting points as correspondences.
4) Calculate their disparities for each correspondence.
5) Compute their X, Y, Z coordinate.
6) Display views.

2. Experiments and Result:

The experiments followed the workflow above. In order to build a 3D structure of a scene, at least two relative images are needed for calculating the position of pixels in XYZ coordinates. In the first step, the two source images, see figure 1, were read into MATLAB workspace.

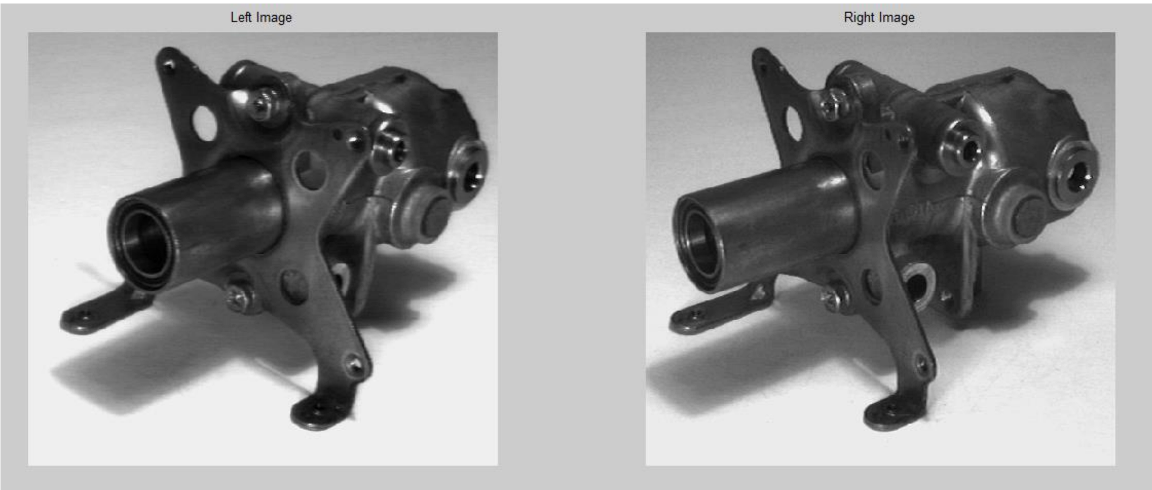

Figure 1: The source images from two view, left image and right image.

2.1 Feature Extraction

All kinds of features from these two images can be employed in the matching process, such as intensity-value, edge elements, and corners. In this experiment, the edge features were employed. The reason is that edge features can represent the structure of an image, they can be found accurately and easily, and the computation would reduce (Galata, 2016). I compared two different edge detectors, Canny edge detector and Sobel edge detector. The Canny method calculates the gradient of an image by using the derivative of Gaussian filter, and locates the edge points that have the maximum local gradient. This method is less sensitive to noise and could detect the true weak edges, since it applies the hysteresis thresholding which detects both strong and weak edges (Canny, 1986). The Sobel method computes an approximation of the gradient of the image intensity and locate the high-frequency regions that correspond to edges (Sobel, 2014). Figure 2 compared the results of their edge detection.

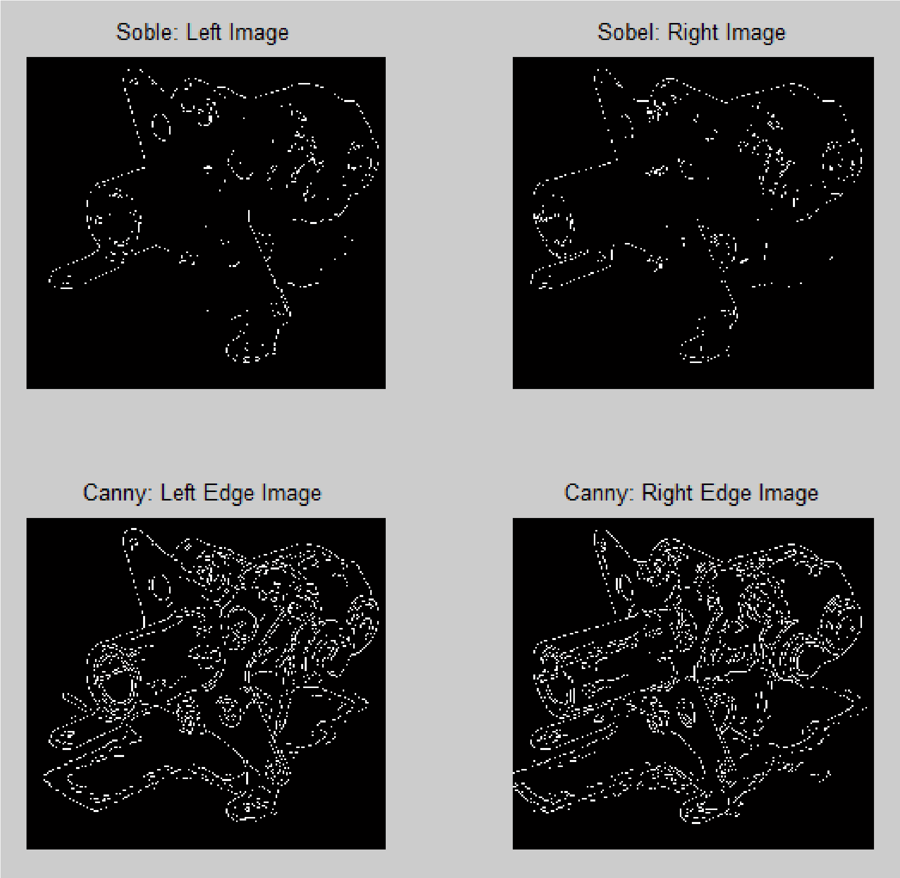

Figure 2: The results of edge detection. The above two images were edges detected by Sobel method, the other two images were detected by Canny method.

Comparing the edge detection results between Sobel method and Canny method, it is obvious that Canny method produced the better result. Almost all the detail of texture of original image can be seen, even the shadow of that object in the bottom of the image. However, the shadow does not represent the real feature of the object in the image which should be considered as noise, therefore, it would affect the 3D structure of the object. Thus, in this case, the Sobel method was employed for locating the interesting points.

2.2 Matching Correspondences

After finding the interesting points of both left and right images, the next step was focused on matching the correspondence between left image and right image. Since these two images were rectified, which means their epi-polar lines were already aligned, we can, therefore, match the correspondences by looking at pixels in one line rather than searching globally. The epi-polar line restricts the corresponding feature in right image lies on the same line of the left image. This constraint reduces the searching space from two-dimensions to one-dimension, and is reliable for matching corresponding features. The approach to matching the process is that for each feature point in the left image, compute the distances with every feature point in the corresponding epi-polar line of right image, and then select the one with minimum Euclidean distance. The reason is that every pair of the corresponding points should have the similar location in left and right images. The pseudo-code of this process follow:

```
for each row of the image
   find the feature points in the left image;
   find the feature points in the right image;
   for each feature point in the left image
      compute the disparities with every point in the right image;
   end
   select the disparity that has the minimum absolute value
end
```

With these minimum distances, the XYZ coordinate can be calculated by the follow equation:

$$\begin{cases} Z \propto \dfrac{1.0}{K + disparity} \\ X = Z * \dfrac{x_l}{f} - \dfrac{b}{2} \\ Y = Z * \dfrac{y_l}{f} \end{cases}$$

where the disparity is the minimum distance scaled by the physical size of pixels (0.011 mm). The f is the focal length of camera (17.0mm), and b is the value of baseline. Here I assumed that the baseline is 1, since this is a constant value that will not affect the relative position of X coordinate but only their scale. For the value of K, it was set to be 1000 firstly, and its effect will be discussed later. By applying the process above, the 3D view of the industrial scene was shown below:

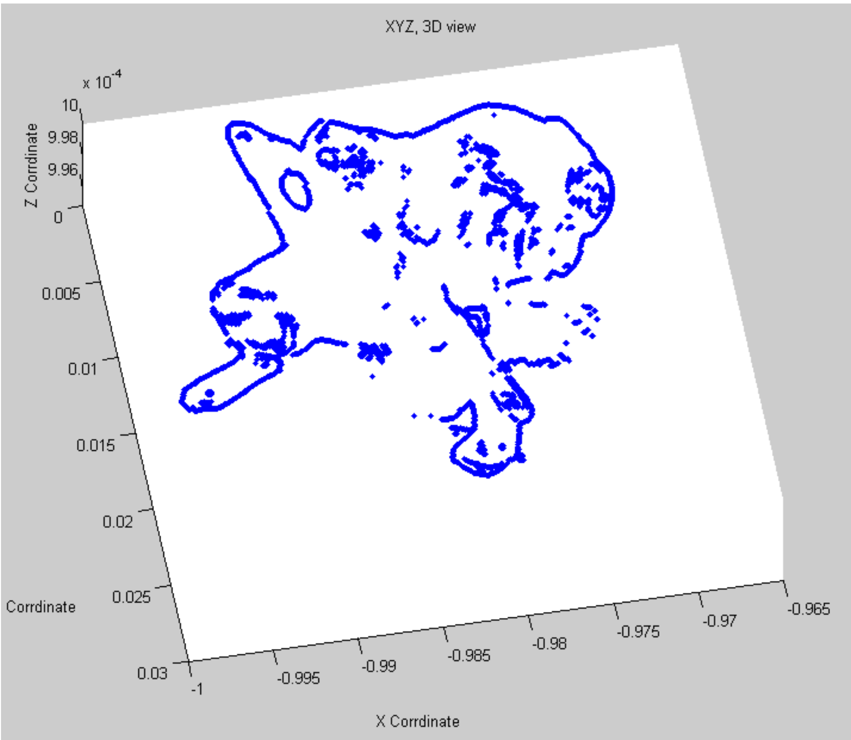

Figure 3: 3D plot of the reconstructed industrial scene.

From figure 3, the main edges of the industrial scene were displayed clearly in 3D view. However, the bottom part of the view was not clear. There were some isolated feature points and some edges were broken as well. The reason could be the mismatching of the corresponding features or the process of feature extraction where many isolated points could also be found in the bottom of edge images (figure 2). The above view (X,Z coordinates)of and side view (Y,Z coordinates) were shown in figure 4. The major number of points overlaid together, therefore, the structure could barely be observed.

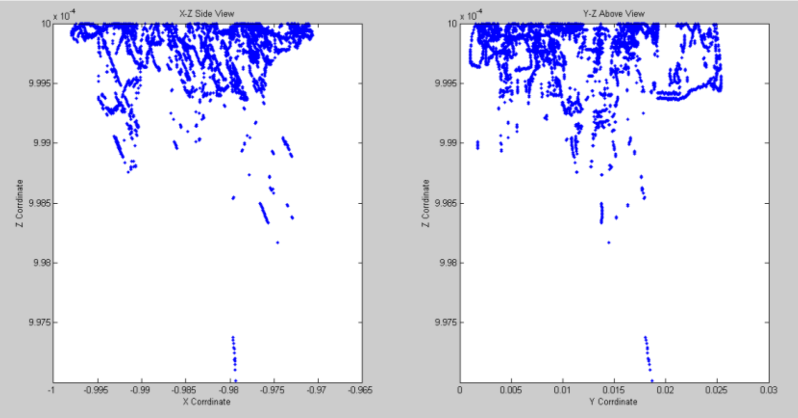

Figure 4: Above View and Side View from the reconstructed 3D structure.

2.3 Adjusting K value

In order to find a proper value of k, different k values were applied to the previous processes. Figure 5 shown the results of 3D view with k value from 0 to 10000.

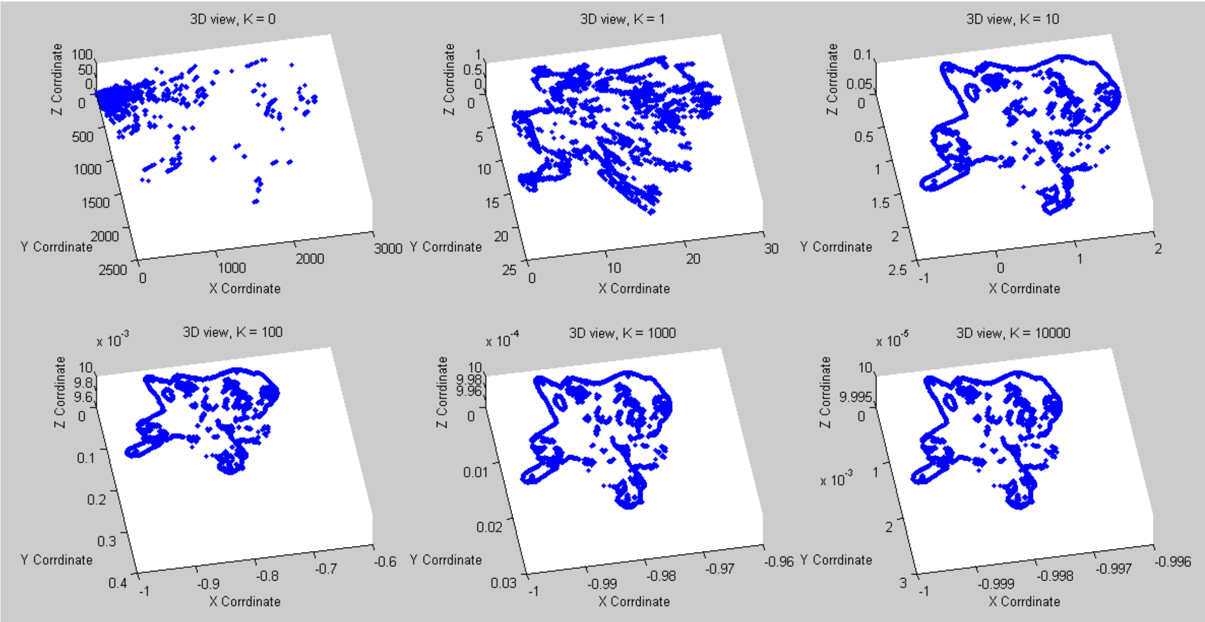

Figure 5: 3D view with different K values (0, 1, 10, 100, 1000, 10000 from left to right and top to down)

Without k value, all the points were prone to be gathered together. When the k value was increasing, the structure of the scene became much clearer. As the Z coordinate is proportional to 1.0/ (K + disparity), the K value controls the scale of Z coordinate.

For the calculation of Z coordinate, an extra overall scaling factor is introduced to the above equation. Combining with the formal equation of Z coordinate, the following equation was formed: $Z = \frac{1.0}{(K+disparit)} * \alpha = \frac{bf}{disparity}$, where b stands for baseline value, α is the scaling factor. Then, the α is calculated by $\alpha = bf\ (\frac{K}{disparity} + 1)$.