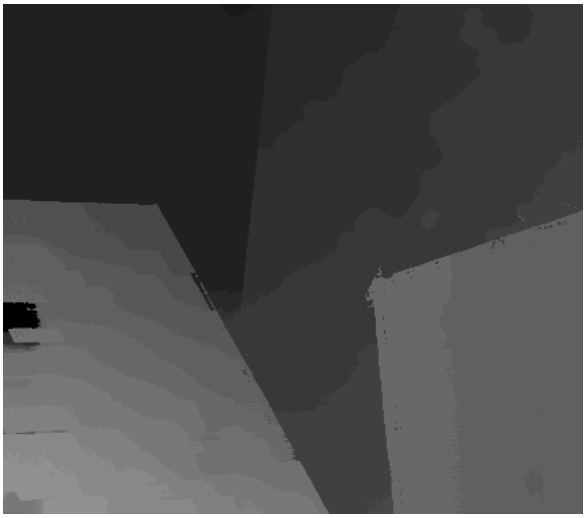
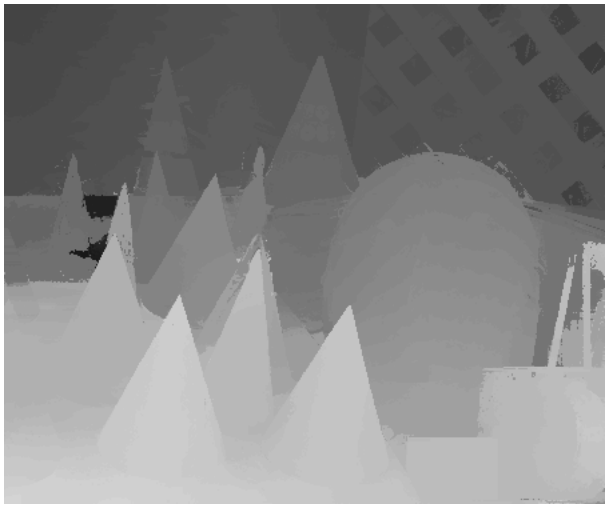
Computer Vision HW4 Report

Student ID: B07303024
Name: 李品樺

Visualize the disparity map of 4 testing images.

Tsukuba	Venus
	
Teddy	**Cones**
	

Report the bad pixel ratio of 2 testing images with given ground truth (Tsukuba/Teddy).

	bad pixel ratio
Tsukuba	4.21%
Teddy	13.89%

Describe your algorithm in terms of 4-step pipeline.

- **Step 1: Cost Computation**

  I choose squared difference (SD) to compute the matching cost. Using the following shift_Image() function and the help of np.roll() function to compute the difference of the two images without using many for-loops.

```python
def shift_image(X, dx, dy):
    X = np.roll(X, dy, axis=0)
    X = np.roll(X, dx, axis=1)
    if dy>0:
        X[:dy, :] = 0
    elif dy<0:
        X[dy:, :] = 0
    if dx>0:
        X[:, :dx] = 0
    elif dx<0:
        X[:, dx:] = 0
    return X
```

- **Step 2: Cost Aggregation**

  In order to refine the cost according to nearby costs, I use xip.jointBilateralFilter() function and take the original colored image as guidance. Note that choices of parameters of joint bilateral filter, such as sigmaColor, sigmaSpace, will affect the filtered result.

- **Step3: Disparity Optimization**

  I utilize np.argmin() function to perform “winner take all” on the disparity images.

- **Step4: Disparity Refinement**

  I implement left-right consistency check by checking each pixel value in two for loops. Then use hole filling to correct the disparity map. Finally, a weighted median filtering is used on the left image to generate the result. These procedures aim to enhance the disparity image.