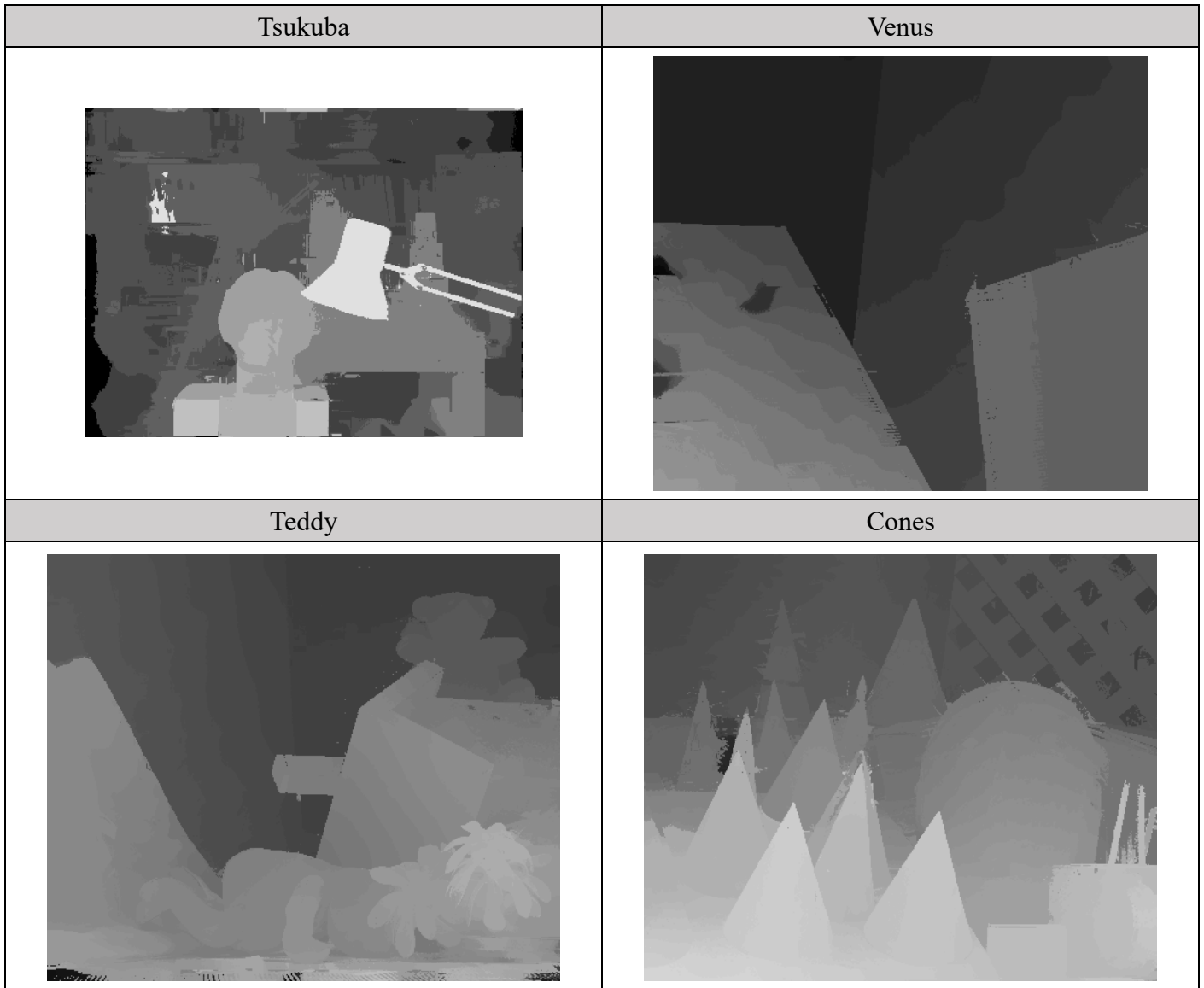


# Computer Vision HW4 Report

Student ID: R12945040

Name: 郭思言

Visualize the disparity map of 4 testing images.



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

	bad pixel ratio
Tsukuba	4.04%
Teddy	9.57%

```
~/Desktop/CV/hw/hw4/hw4_student ..... ✓ cvhw4
python main.py --image Tsukuba
[Bad Pixel Ratio] 4.04%
duration: 4.445 sec

~/De/CV/hw/hw4/hw4_student ..... ✓ took 5s cvhw4
python main.py --image Venus
duration: 6.341 sec

~/De/CV/hw/hw4/hw4_student ..... ✓ took 7s cvhw4
python main.py --image Teddy
[Bad Pixel Ratio] 9.57%
duration: 9.127 sec

~/De/CV/hw/hw4/hw4_student ..... ✓ took 9s cvhw4
python main.py --image Cones
duration: 8.488 sec
```

**Describe your algorithm in terms of 4-step pipeline.**

**My implementation of stereo matching in a 4-step pipeline:**

**1. Cost Computation:**

- i. Convert the input images to a local binary pattern representation (Census transform). This involves comparing each pixel in a window to the center pixel and encoding the result as a binary pattern.
- ii. Compute the Hamming distance (census cost) between corresponding windows in the left and right images to obtain the matching cost. This is done for both left-to-right and right-to-left directions to facilitate later consistency checks.

**2. Cost Aggregation:**

- i. Apply a joint bilateral filter to the computed cost volumes. This step refines the cost values by considering nearby costs and the intensity similarity, which helps to preserve edges and smooth homogeneous regions.

**3. Disparity Optimization:**

- i. Determine the disparity at each pixel using the "winner-take-all" approach (for each pixel, select the disparity with the minimum cost from the refined cost volumes, resulting in initial disparity maps `winner\_L` and `winner\_R`), where the disparity corresponding to the minimum cost is selected. This is done separately for both left-to-right and right-to-left cost volumes.

**4. Disparity Refinement:**

- i. Perform a left-right consistency check to identify and mark inconsistent disparities as invalid: Compare the left-to-right disparity map with the right-to-left disparity map to detect inconsistencies. Mark inconsistent disparities as invalid (-1).
- ii. Fill holes (invalid disparities) by propagating the closest valid disparities from the left and right neighbors: For each invalid disparity, find the closest valid disparities from the left and right neighbors and use the smaller one to fill the hole.
- iii. Apply a weighted median filter to further refine the disparity map, enhancing the accuracy and smoothness of the final output.