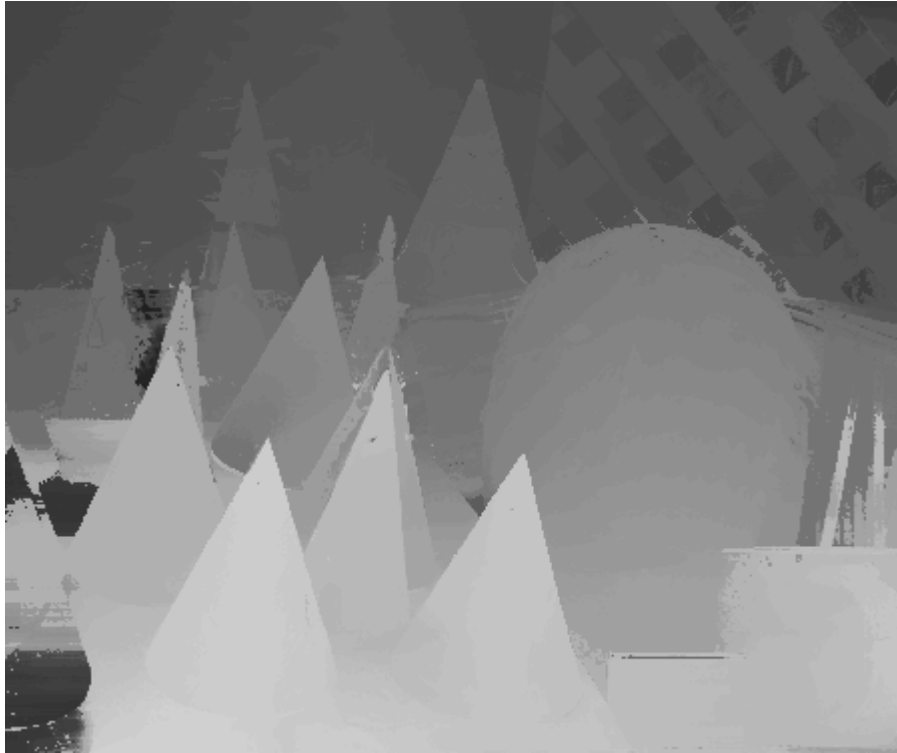


# Computer Vision, Spring 2023 HW4

B11705009 An-Che, Liang

## Part 1: Visualization

### Cones



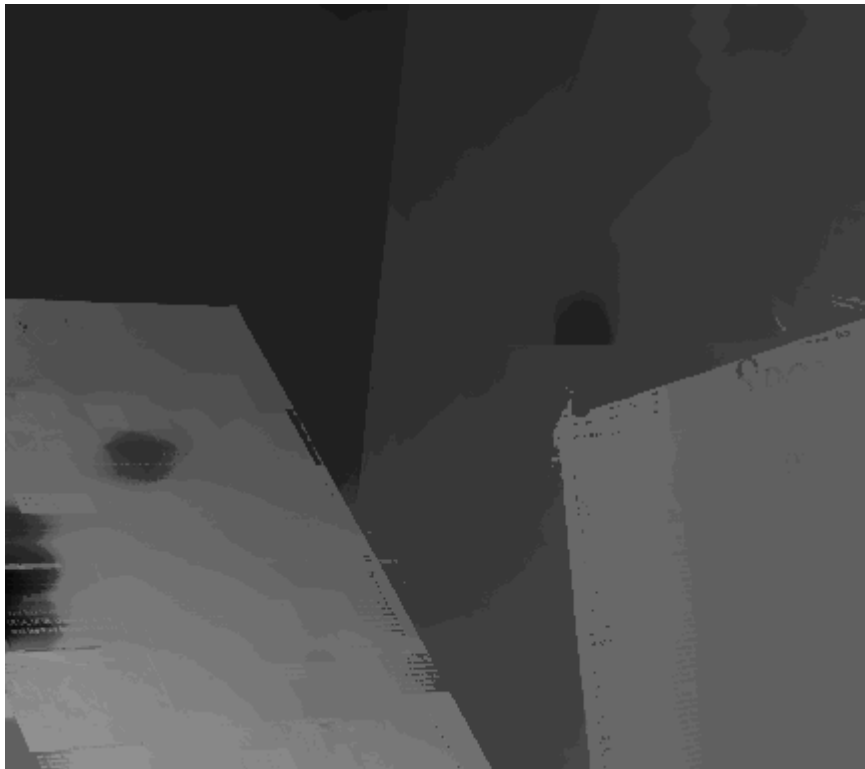
### Teddy



Tsukuba



Venus



## Part 2: Bad Pixel Ratio

1. Tsukuba: 4.20%
2. Teddy: 15.69%

## Part 3: Algorithm

### Compute matching cost

I used Absolute difference to calculate the matching cost of two pixels.

### Cost Aggregation

I used `xip.jointBilateralFilter` to smooth the noise of matching cost with the original image as guidance, which is different from the assignment information slide, but yield optimum results.

## Disparity Optimization

I use `np.argmin()` to get the disparity with minimum matching cost at each pixel.

## Disparity Refinement

Use like the assignment information slide, I first use left-right consistency check to mark points with inconsistent disparity, then find the vertically closest point with valid disparity, compare the left result and right result, then choose the smaller one to fill the hole. Finally use `xip.weightedMedianFilter` with regard of the original image to enhance the edges.