

**Assignment 4:**  
**Stereo Matching**

Computer Vision  
National Taiwan University

Fall 2019

TA: 李曼妤 (leemanyu@media.ee.ntu.edu.tw)



# Part 2: Disparity Estimation

- Compute disparity maps of the **left image** for the four standard test pairs from [Middlebury v2](#)
- Evaluation metric: bad pixel ratio (error threshold = 1)

*Tsukuba*



Max disp = 15  
Scale factor = 16

*Venus*



Max disp = 20  
Scale factor = 8

*Teddy*



Max disp = 60  
Scale factor = 4

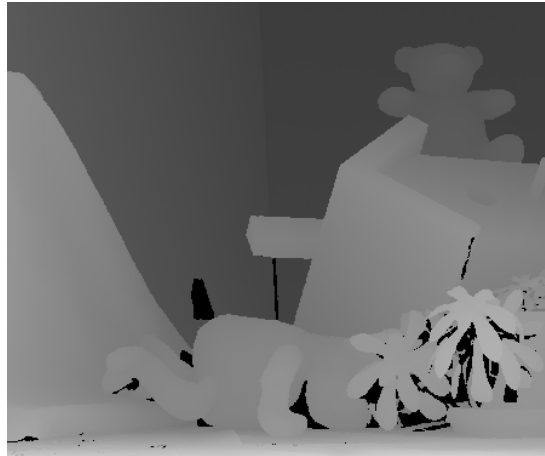
*Cones*



Max disp = 60  
Scale factor = 4

# Part 2: Description

- Max disp : the maximum possible disparity
- Scale factor : for visualization purpose



- Bad pixel ratio , error\_threshold
  - GT\_valid : pixels in GT that aren't zero
  - Bad\_pixel :  $| \text{GT} - \text{output} | > \text{error\_threshold}$  in GT\_valid
  - Bad pixel ratio = # of Bad\_pixel / # of GT\_valid

# Part 2: Description

- Typical pipeline

```
def computeDisp(Il, Ir, max_disp):
    h, w, ch = Il.shape
    labels = np.zeros((h, w), dtype=np.float32)
    Il = Il.astype(np.float32)
    Ir = Ir.astype(np.float32)
    # >>> Cost computation
    # TODO: Compute matching cost from Il and Ir

    # >>> Cost aggregation
    # TODO: Refine cost by aggregate nearby costs

    # >>> Disparity optimization
    # TODO: Find optimal disparity based on estimated cost. Usually winner-take-all.

    # >>> Disparity refinement
    # TODO: Do whatever to enhance the disparity map

    return labels.astype(np.uint8)
```

- You can also use other methods !
  - Except for deep learning

# Part 2: Regulations

- Implement your code in [computeDisp.py](#)
- Evaluate using [eval\\_stereo.py](#)
  - We will run ***python3 eval\_stereo.py*** to test your function for grading
  - **Do not edit this file !**
  - If you only pass the score using your own `eval_stereo.py` will not get any points
- Do not use deep matching costs
- If runs longer than 30 minutes will only get 70 % score for part 2
- Materials [[link](#)]

# Part 2: Regulations

- packages:
  - You can not use library that directly help compute disparity.
  - You can use any library to help generate sub-results (ex. median filters, bilateral filters, box filters,...).
  - If you are not sure whether some libraries are permitted, you need to ask TA (FB Q&A or e-mails).

# Grading (Total 15 %)

- Part2: 10%

**Table.** Score vs. bad pixel ratio

Score	Tsukuba	Venus	Teddy	Cones
10	< 8	< 5	< 18	< 15
8	>= 8	>= 5	>= 18	>= 15
3	>= 9	>= 7	>= 24	>= 20
0	>= 10	>= 10	>= 30	>= 25

- Ranking according to your avg. score among the class
  - (Bonus) If top 10(+1%)
- Report : 1%(Part1) + 4% = 5%



# Report

- Your student ID and name
- Part 1
  - Write down your proof.
- Part 2
  - Explain your algorithm in terms of the standard 4-step pipeline. (cost computation, cost aggregation, disp. optimization, disp. refinement)
  - Show your output disparity maps in the report.
  - Show your bad pixel ratio of each disparity maps in the report.
  - Your reference papers or websites.

# Submission

- Code: \*.py (Python 3.5+)
- A PDF report (StudentID.pdf, ex. R07654321.pdf)
- Put all above files in a directory named StudentID and compress to a zip file named StudentID.zip
  - Other compression format (ex: rar, tar.gz....) are **NOT** allowed
  - e.g. R07654321.zip
- Your grade will be multiplied by 0.8 if you have any format error.
- Submit to CEIBA
- Deadline: 12/24 11:00 pm