ICG Final Project Demo & Presentation

B04504116 鄭人豪
B04901069 林志皓

Obejctive

Reconsstruct 2.5D image from 2 plane image

RGB*2 → **RGB-D**

Left

Right

Pipeline

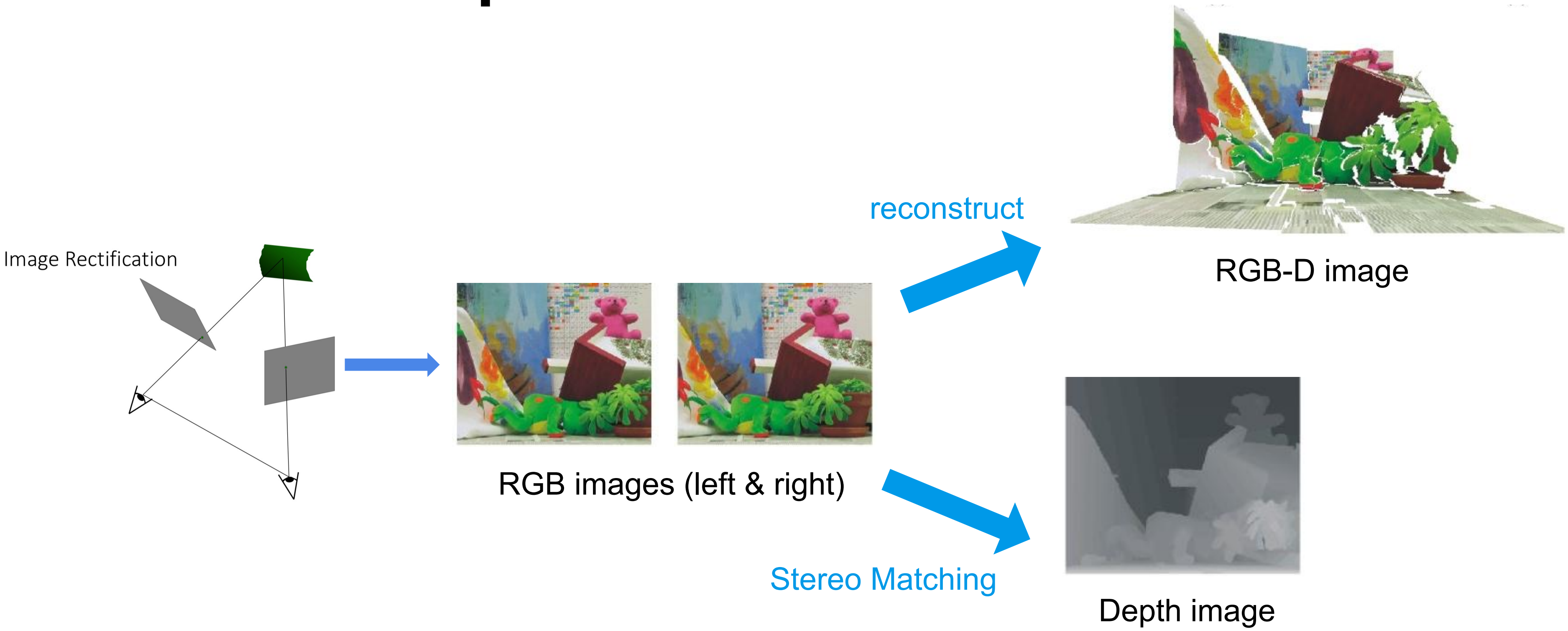
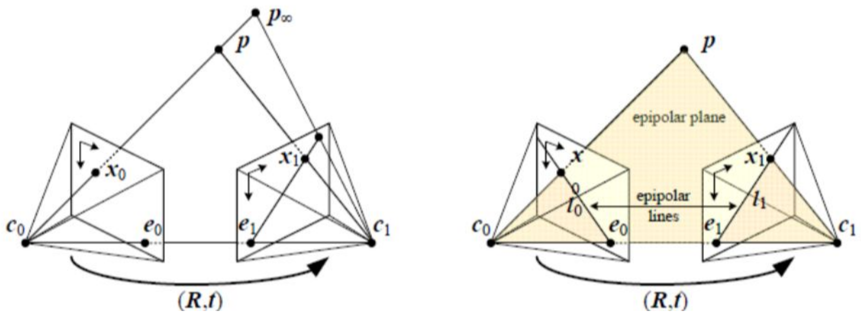


Image Rectification
reconstruct
RGB-D image
RGB images (left & right)
Stereo Matching
Depth image

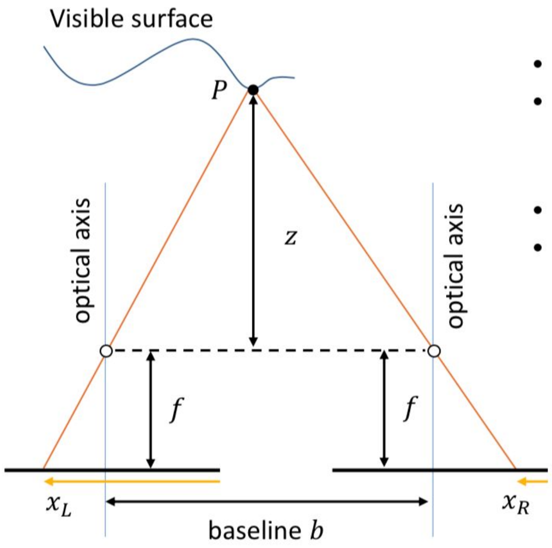
Introduction - Stereo Matching

- For pixel x_0 in one image, where is the corresponding point x_1 in another image?
  - Stereo: two or more input views
- Based on the epipolar geometry, corresponding points lie on the epipolar lines
  - A matching problem



Introduction - Stereo Matching

Depth from Disparity



- Disparity $d = x_L - x_R$
- It can be derived that
$$d = \frac{f \cdot b}{z}$$
- Disparity = 0 for distant points
- Larger disparity for closer points

Pipeline - Stereo Matching

- Cost computation
- Cost (support) aggregation
- Disparity optimization
- Disparity refinement

Results

```
Tsukuba
* Elapsed time (cost computation): 2.240678 sec.
* Elapsed time (cost aggregation): 381.830109 sec.
* Elapsed time (cost computation): 2.282026 sec.
* Elapsed time (cost aggregation): 391.524820 sec.
* Elapsed time (disparity optimization): 0.015572 sec.
* Elapsed time (disparity refinement): 1.173584 sec.
Venus
* Elapsed time (cost computation): 4.027865 sec.
* Elapsed time (cost aggregation): 717.365762 sec.
* Elapsed time (cost computation): 3.965656 sec.
* Elapsed time (cost aggregation): 734.562450 sec.
* Elapsed time (disparity optimization): 0.028704 sec.
* Elapsed time (disparity refinement): 1.574592 sec.
Teddy
* Elapsed time (cost computation): 9.067297 sec.
* Elapsed time (cost aggregation): 1488.150797 sec.
* Elapsed time (cost computation): 8.334749 sec.
* Elapsed time (cost aggregation): 1452.823378 sec.
* Elapsed time (disparity optimization): 0.121340 sec.
* Elapsed time (disparity refinement): 2.727065 sec.
Cones
* Elapsed time (cost computation): 8.328705 sec.
* Elapsed time (cost aggregation): 1016.414072 sec.
* Elapsed time (cost computation): 8.331107 sec.
* Elapsed time (cost aggregation): 1004.053331 sec.
* Elapsed time (disparity optimization): 0.109782 sec.
* Elapsed time (disparity refinement): 2.274131 sec.
```

Drawbacks of traditional method

1.Extremely slow

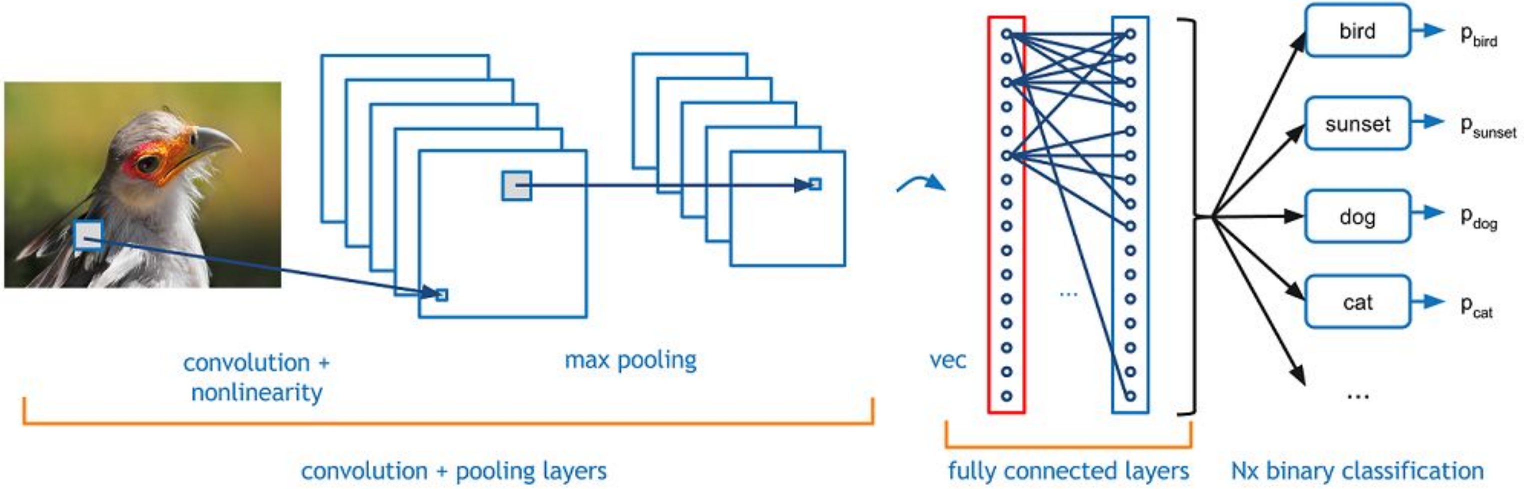
2.Accuracy with limitation

3.Sensitive to light, color consistency

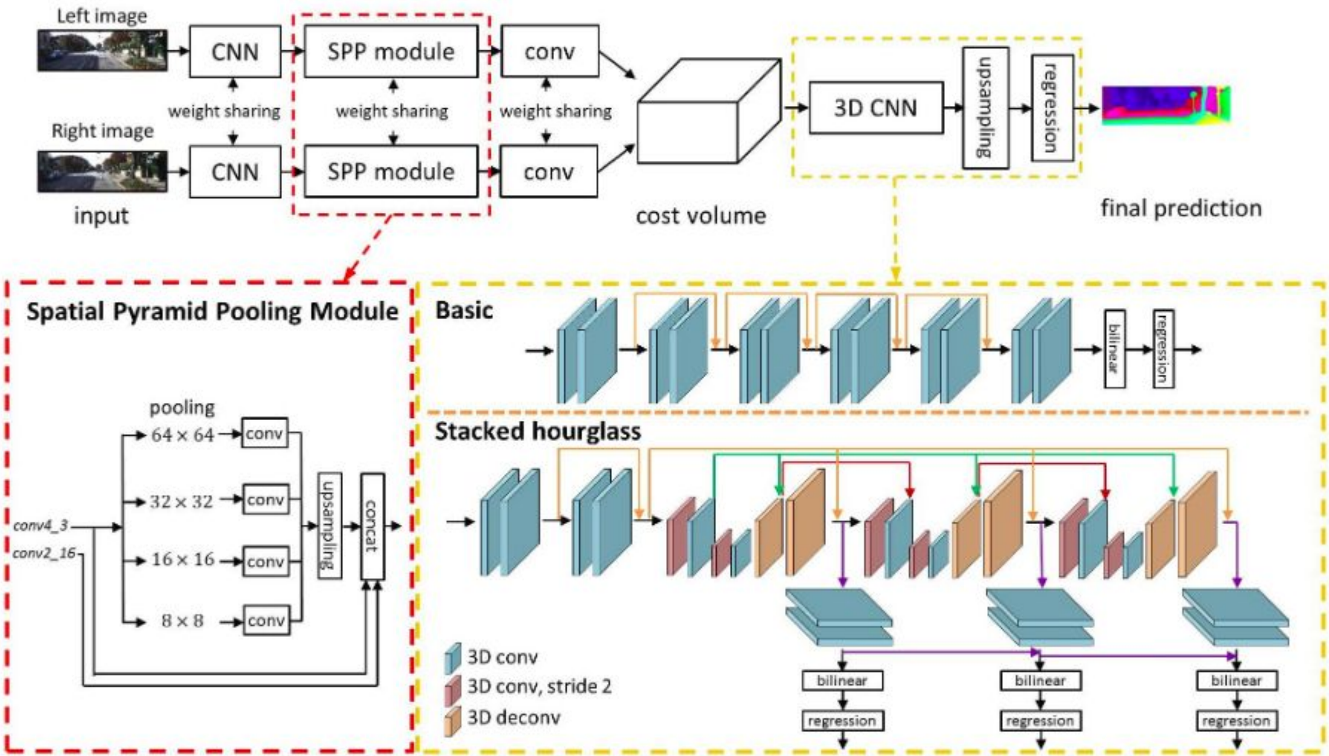
Deep learning

Advantage: fast, can handle harsh case

based on Convolutional Neurual Network



PSMNet 2018 state-of-the-art



KITTI Dataset

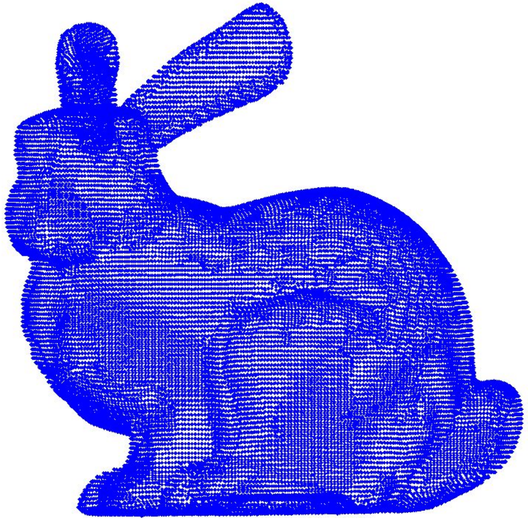
left image

right image

3D Reconstruction

match RGB and depth infomation for each pixel

and generate point clouds data



Implementation

using python to process RGB & depth infomation

generate vertex coordinations and colors

render with **python / WebGL**

Demo Time !

```python
def normalize_vertices(vertices, width, height):
    vertices = np.array(vertices).reshape(-1, 3)

    depth = vertices[:, 2]
    scale = 5
    vertices[:, 0] = (vertices[:, 0] - width//2) * scale/ width
    vertices[:, 1] = (vertices[:, 1] -height//2) * scale/ height
    vertices[:, 2] = (depth - depth.mean()) * scale/(depth.max()- depth.min())
    vertices = vertices.reshape(-1, )
    return list(vertices)

def normalize_colors(colors):
    colors = np.array(colors)
    colors = (colors / 400).round(5)
    return list(colors)

def create():
    rgb_img = cv.imread(args.rgb)
    height, width, _ = rgb_img.shape
    if args.depth.endswith('pfm'):
        depth = readPFM(args.depth)
    elif args.depth.endswith('npy'):
        depth = np.load(args.depth)
```