

# Depth Estimate from Stereo cameras

---

Tham khảo:

- Depth estimation from stereo cameras (pdf)

## Disparity estimation

- Nội dung của stereo matching là dựa trên việc tìm sự tương thích giữa hai ảnh đầu vào. Trong bài này, việc khớp gwei hai điểm được xem xét bằng việc kiểm tra nxn pixel láng giềng N quanh cả 2 điểm đấy.
- Matching block được đặt cho mỗi pixel trong một hình ảnh.
- Sự khác nhau trong vị trí của các điểm trong các mặt phẳng ảnh là *disparity* của điểm đó.
- Matching block size là một thông số quan trọng ảnh hưởng tới kết quả của việc ước tính disparity. Block nhỏ hơn có thể tìm kiếm chi tiết tốt hơn, nhưng dễ có lỗi, trong khi block lớn hơn thì robust hơn nhưng lại bỏ qua các chi tiết nhỏ.

## Depth from disparity

- Disparity không giống depth. Bối quan hệ giữa disparity và depth dựa trên cấu hình camera
- Công thức:  $\text{depth} = \text{baseline} * \text{focal} / \text{disparity}$

Ví dụ:

```
*To obtain depth, you need to convert disparity using the following
formula:
depth = baseline * focal / disparity
For KITTI the baseline is 0.54m and the focal ~721 pixels.
The relative disparity outputted by the model has to be scaled by 1242
which is the original image size.
The final formula is:
depth = 0.54 * 721 / (1242 * disp)*
```

## Phương pháp

1. Dùng bộ dữ liệu stereo (ApolloDataset hoặc CityscapesDataset), chạy qua PSMNet để ra disparity map.
2. Từ disparity map, kết hợp với thông số camera để tính ra depth map
3. Depth map + object detected -> Dense point cloud hay gì???

- Mạng đã chạy được
- Dữ liệu camera:
  - Citicape có bộ dữ liệu camera:

```
{
  "extrinsic": {
    "baseline": 0.209313,
    "pitch": 0.038,
    "roll": 0.0,
    "x": 1.7,
    "y": 0.1,
    "yaw": -0.0195,
    "z": 1.22
  },
  "intrinsic": {
    "fx": 2262.52,
    "fy": 2265.3017905988554,
    "u0": 1096.98,
    "v0": 513.137
  }
}
```

Trong đó: *baseline* (m), *fx*, *fy* (pixel).

*fx*, *fy* là gì???

## Các bộ dữ liệu:

### 1. ApolloScapesDataset:

- [Link download](#)
- [git repository](#)

Dữ liệu:

- Ảnh stereo
- Ảnh disparity
- Ảnh depth

==> Sử dụng tập Stereo\_train\_001 (4.4GB) có đầy đủ thông tin trên để tính depth, nhưng bộ dữ liệu này không có ảnh depth được tính sẵn.

### 2. Cityscapes

- [Link download](#)
- [git repository](#)

Dữ liệu:

- Có dữ liệu để train nhận dạng cột đèn giao thông
- Có disparity
- Có thông số camera
- Không có ảnh depth có sẵn

## Các bước dự kiến

1. Chạy lại file submission.py chạy mạng PSMNet để ra disparity map.
2. Dùng bộ Stereo\_train\_001 để tính depth từ disparity.
3. Kết nối với object detection
4. Nối thông từ chạy ảnh stereo ra depth.

## Triển khai:

intrinsic.txt

```
K = [2301.3147, 0, 1489.8536; 0, 2301.3147, 479.1750; 0, 0, 1]
```

==> focal = 2301.3147 (pixel)

## Khó khăn hiện tại

- Khó xác định baseline và focal của hệ thống stereo
- Chưa ghép được hệ thống

## Một số opensource

### PMSNet

- Thử chạy file submission.py với dữ liệu đã train sẵn \*.tar
- Hi vọng có thể sử dụng mô hình đã train, chế biến lại code submission.py để dùng luôn.

10/7/2019:

- Code submission.py đang chạy trên cuda, gặp một số lỗi khi chạy trên cpu

--> Code này cần CUDA

### DenseDepth

12/7/2019: DenseDepth từ [link này](#).

Tập trung vào Depth môi trường đường phố ngoài trời. Sử dụng model đã train sẵn: [KITTI](#)

Chạy `python test.py` với model kitti.h5 và ảnh từ KITTI dataset (phải resize về 348x1248)

Thắc mắc:

- Input:
  - Model train sẵn
  - Ảnh
- Output:
  - Chiều sâu???

Vì sao chỉ 1 ảnh lại có thể ra đc chiều sâu???

Cài cuda và sử dụng trên venv

1. Cài cuda

<https://medium.com/@kapilvarshney/how-to-setup-ubuntu-16-04-with-cuda-gpu-and-other-requirements-for-deep-learning-f547db75f227>

2. Tạo môi trường venv để chạy lại code submission.py

3. Cài các gói cần thiết: `pip install -r requirements.txt`

- Kiểm chứng kết quả đo???