3D Object Detection using Geometry Method

•••

Team10:	310611077 郭江穎
	310505018 林書恆
	311555029 鍾維廷

Outline

1. Goal and Motivation
2. Method
    a. Kitti dataset
    b. Utilizing ground-truth 2D bboxs
    c. Using geometry method to derive disparity map from stereo images
    d. Estimate Depth Map with Deep Learning Technique
        i. High Quality Monocular Depth Estimation via Transfer Learning
    e. Method to formulate 3D bboxs
3. Results
    a. Compare performances of 3D bboxs whose disparity map generated respectively by geometry method and deep learning technique
4. Conclusion

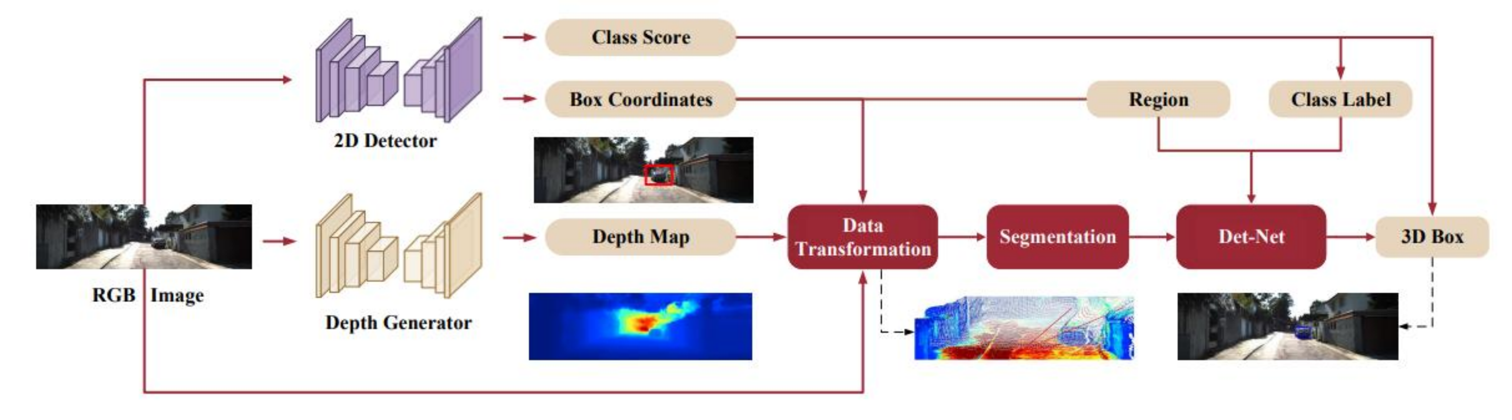
5. References

Goal and Motivation

- Camera-based object detection is a fundamental and crucial technology for autonomous driving.
- Typically, deep learning is employed to accomplish this task, which has been tackled by several works, such as YOLO, Fast R-CNN, etc.
- However, these methods are all 2D detection techniques and do not provide depth information, which is essential for driving tasks such as path planning or collision avoidance.

Goal and Motivation

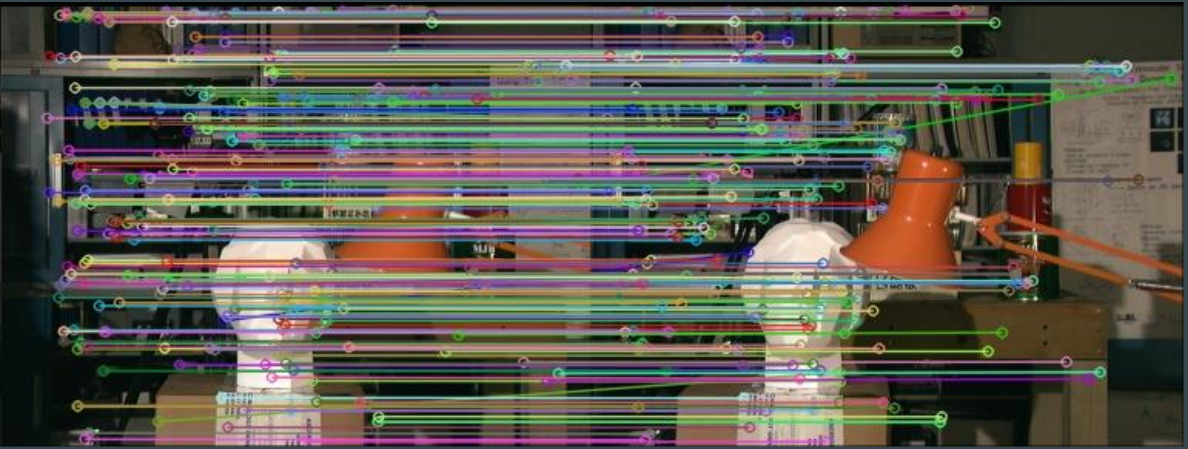
- Due to the ill-posed nature of inferring depth from cameras, the recent approach is to directly employ deep neural networks for depth regression. Subsequently, using the obtained depth and intrinsic matrix, 3D bounding boxes can be computed.

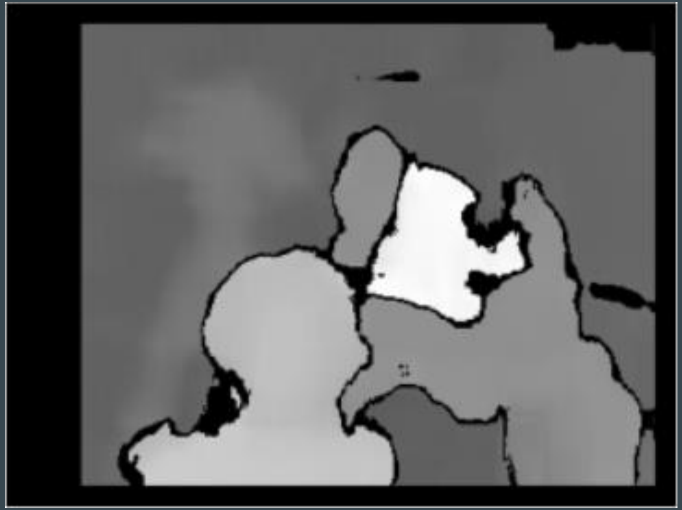


Goal and Motivation

On this topic, we intend to utilize mathematical algorithms for stereo matching and combine them with **SIFT keypoints** to compute Epipolar lines for depth reconstruction.

Subsequently, we will input the obtained results into a 2D object detection model based on the YOLO backbone, to calculate the 3D bounding box for each objects.





Method

Kitti dataset

To achieve our goals, we have chosen to utilize the **KITTI dataset** which provides a realistic autonomous driving scenario with stereo camera data, including both left and right eye views.

Utilizing ground-truth 2D bboxs

- If we directly use Yolo to predict 2D bboxs ,we still lack the informations of the orientation of objects. Therefore, we choose to utilize ground-truth 2D bboxs and the orientation provided by the Kitti datasets, rather than using any deep learning model.

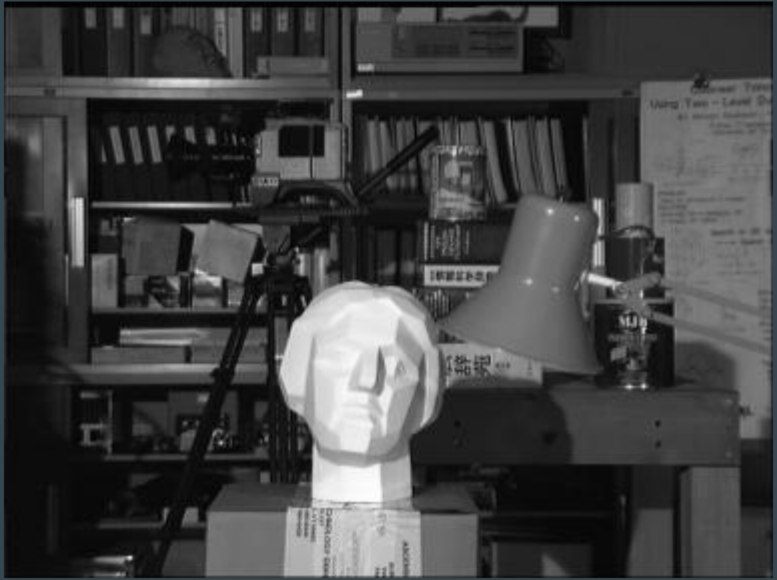
Using geometry method to derive disparity map from stereo images

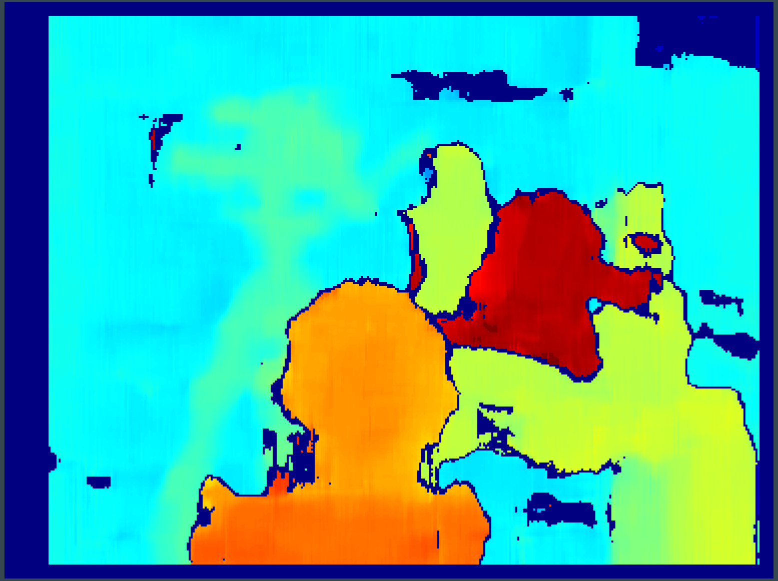
- Projection of an image point back into the scene results in an outgoing ray.
- 3D to 2D:
  - $u = f_x \frac{x_c}{z_c} + o_x$
  - $v = f_y \frac{y_c}{z_c} + o_y$
- 2D to 3D:
  - $x = \frac{z}{f_x}(u - o_x)$
  - $y = \frac{z}{f_y}(v - o_y)$

Using geometry method to derive disparity map from stereo images

- Using two camera and find the corresponding point
  - Left camera (ul,vl)
  - Right camera (ur,vr)
- With two outgoing ray equations from both left and right camera, where these two ray intersect is the physical point lies.
-
- Totally we have 4 equations and internal parameters fx,fy,b,ox,oy are known, thus we can find (x,y,z)in the scene.
- We derive the depth.
- Z=$\frac{b f_x}{(u_l - u_r)}$
- Where $(u_l - u_r)$ is called Disparity.

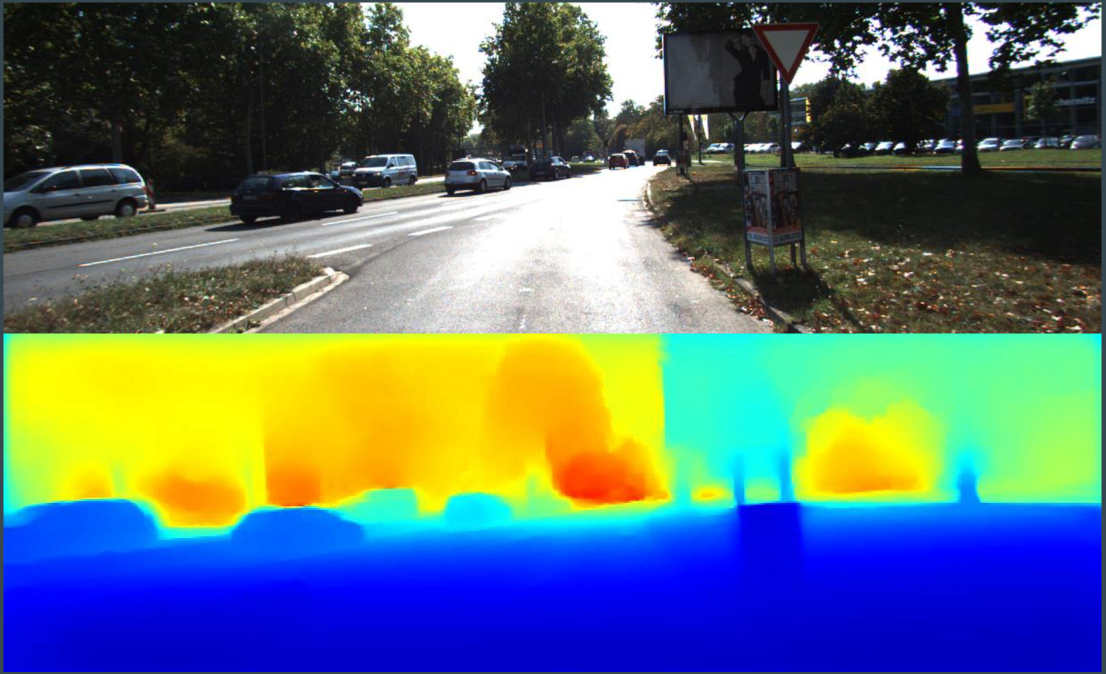
Using geometry method to derive disparity map from stereo images



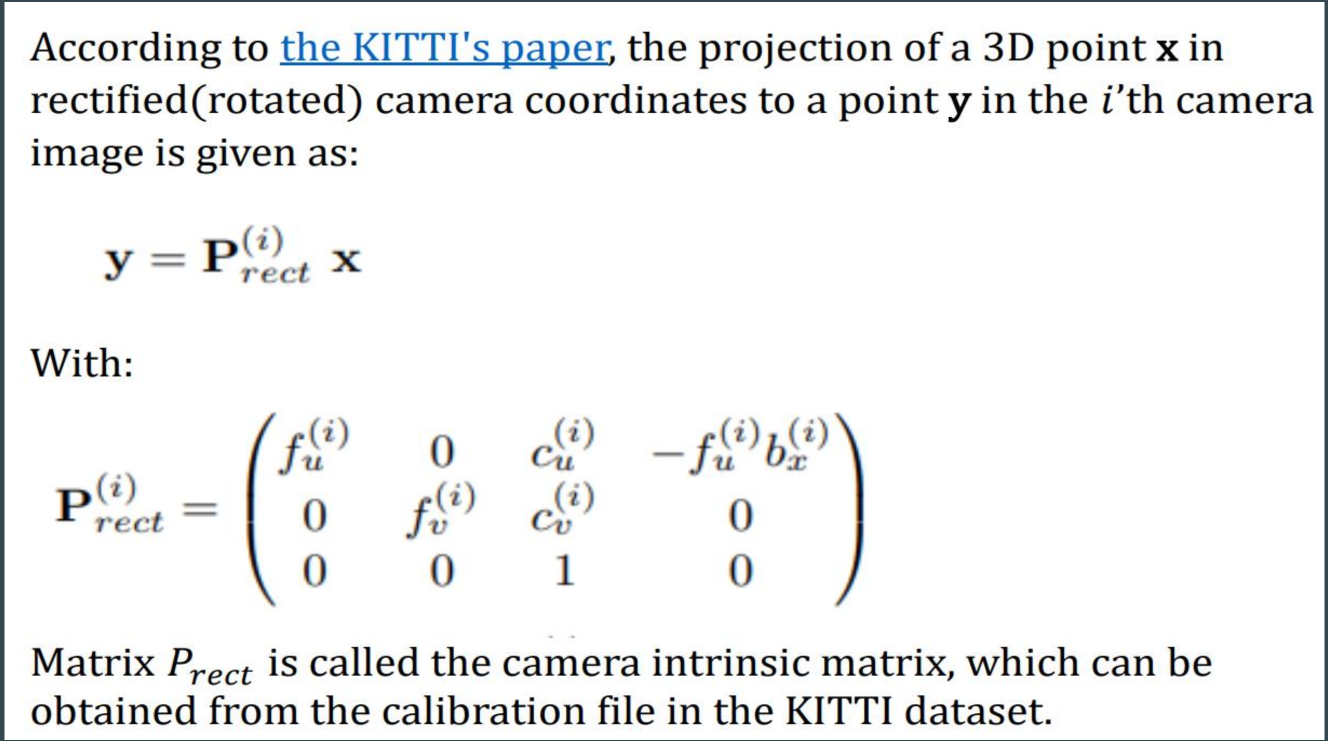


Estimate Depth Map with Deep Learning Technique

- We use the model of “High Quality Monocular via Transfer Learning” to estimate depth map.



Method to formulate 3D bboxs



According to the KITTI's paper, the projection of a 3D point **x** in rectified(rotated) camera coordinates to a point **y** in the i'th camera image is given as:

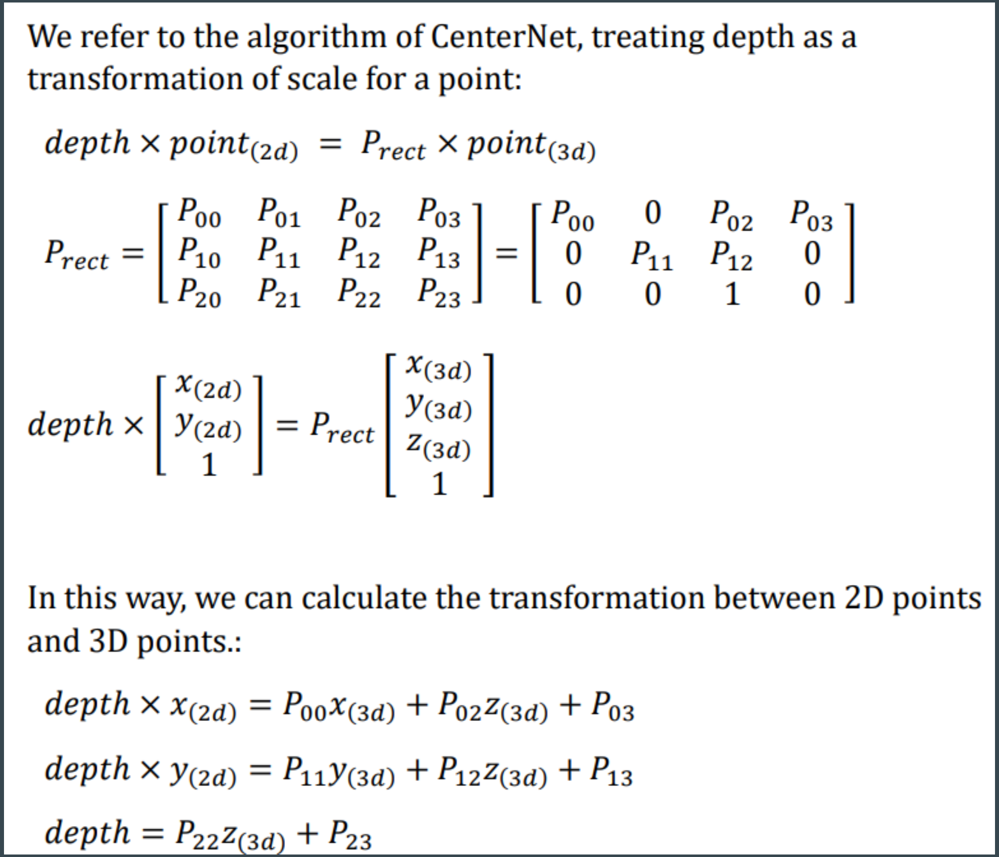
$$\mathbf{y} = \mathbf{P}_{rect}^{(i)}\ \mathbf{x}$$

With:

$$\mathbf{P}_{rect}^{(i)} = \begin{pmatrix} f_u^{(i)} & 0 & c_u^{(i)} & -f_u^{(i)} b_x^{(i)} \\ 0 & f_v^{(i)} & c_v^{(i)} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Matrix P_{rect} is called the camera intrinsic matrix, which can be obtained from the calibration file in the KITTI dataset.

Method to formulate 3D bboxs



We refer to the algorithm of CenterNet, treating depth as a transformation of scale for a point:

$$depth \times point_{(2d)} \;=\; P_{rect} \times point_{(3d)}$$

$$P_{rect} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \end{bmatrix} = \begin{bmatrix} P_{00} & 0 & P_{02} & P_{03} \\ 0 & P_{11} & P_{12} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$depth \times \begin{bmatrix} x_{(2d)} \\ y_{(2d)} \\ 1 \end{bmatrix} = P_{rect} \begin{bmatrix} x_{(3d)} \\ y_{(3d)} \\ z_{(3d)} \\ 1 \end{bmatrix}$$

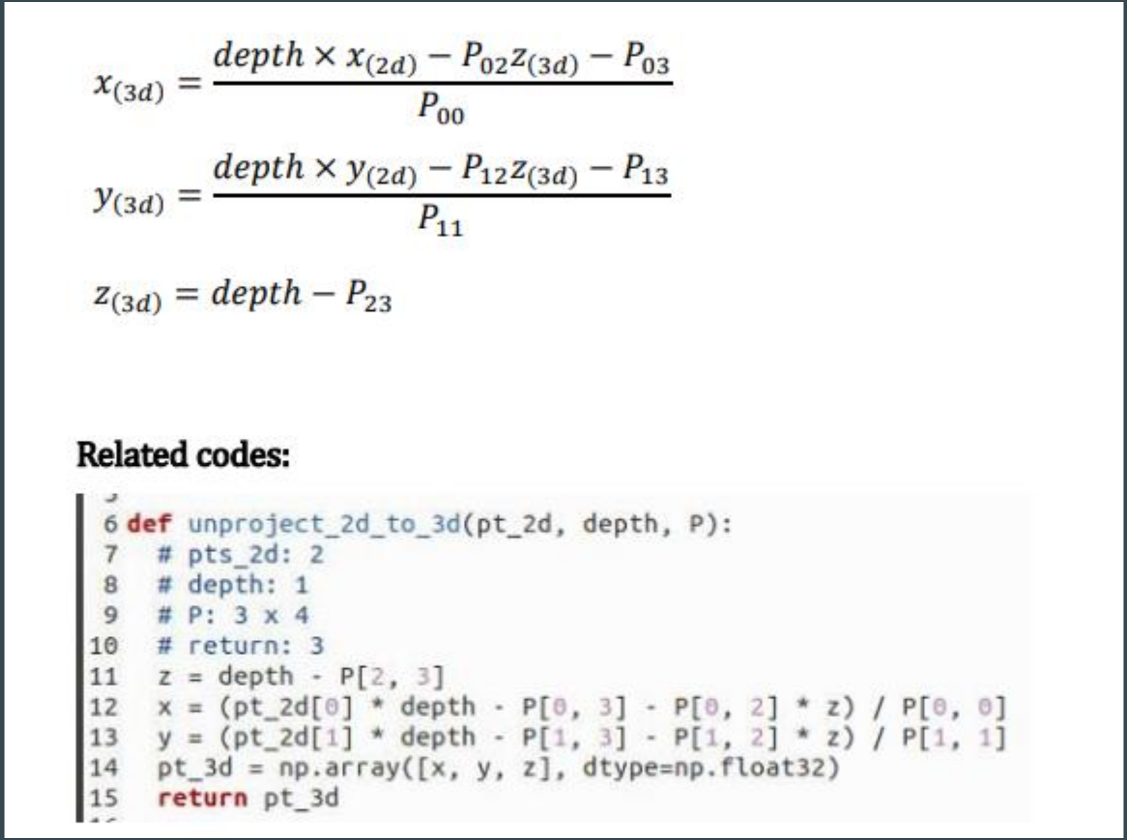
In this way, we can calculate the transformation between 2D points and 3D points.:

$$depth \times x_{(2d)} = P_{00}x_{(3d)} + P_{02}z_{(3d)} + P_{03}$$

$$depth \times y_{(2d)} = P_{11}y_{(3d)} + P_{12}z_{(3d)} + P_{13}$$

$$depth = P_{22}z_{(3d)} + P_{23}$$

Method to formulate 3D bboxs



$$x_{(3d)} = \frac{depth \times x_{(2d)} - P_{02} z_{(3d)} - P_{03}}{P_{00}}$$

$$y_{(3d)} = \frac{depth \times y_{(2d)} - P_{12} z_{(3d)} - P_{13}}{P_{11}}$$

$$z_{(3d)} = depth - P_{23}$$

Related codes:

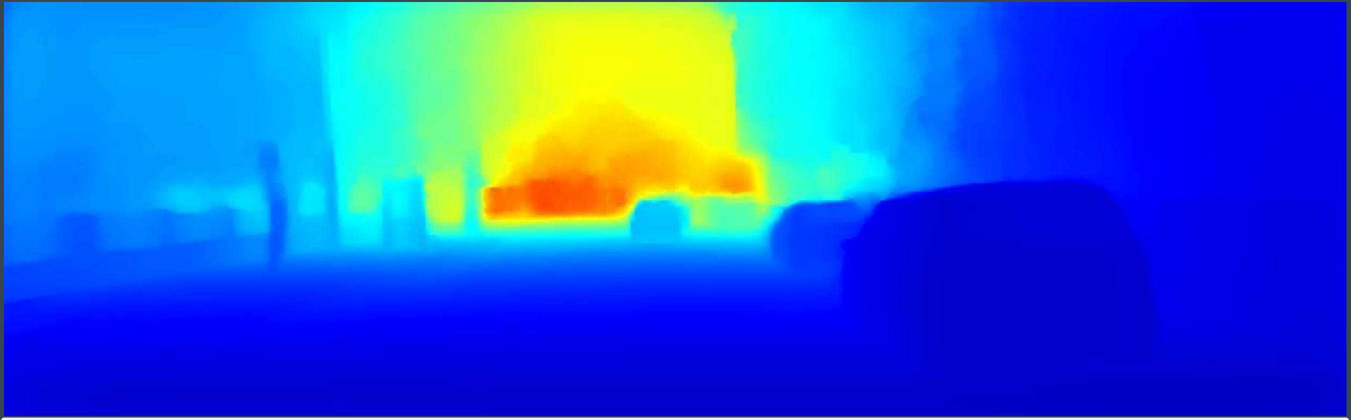
```
def unproject_2d_to_3d(pt_2d, depth, P):
  # pts_2d: 2
  # depth: 1
  # P: 3 x 4
  # return: 3
  z = depth - P[2, 3]
  x = (pt_2d[0] * depth - P[0, 3] - P[0, 2] * z) / P[0, 0]
  y = (pt_2d[1] * depth - P[1, 3] - P[1, 2] * z) / P[1, 1]
  pt_3d = np.array([x, y, z], dtype=np.float32)
  return pt_3d
```

Results

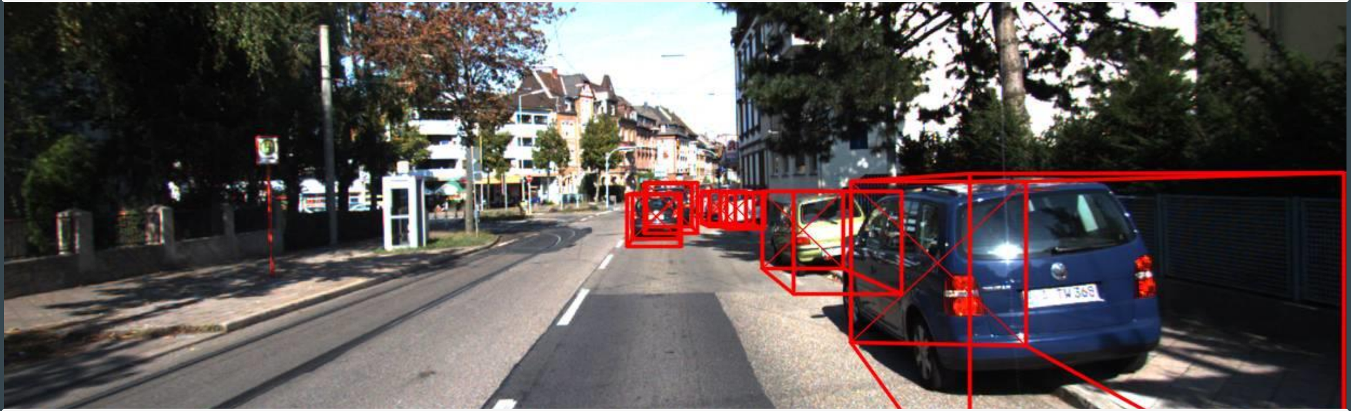
•••

Compare performances of 3D bboxs whose disparity map generated respectively by geometry method and deep learning technique

Image 1 based on Deep learning

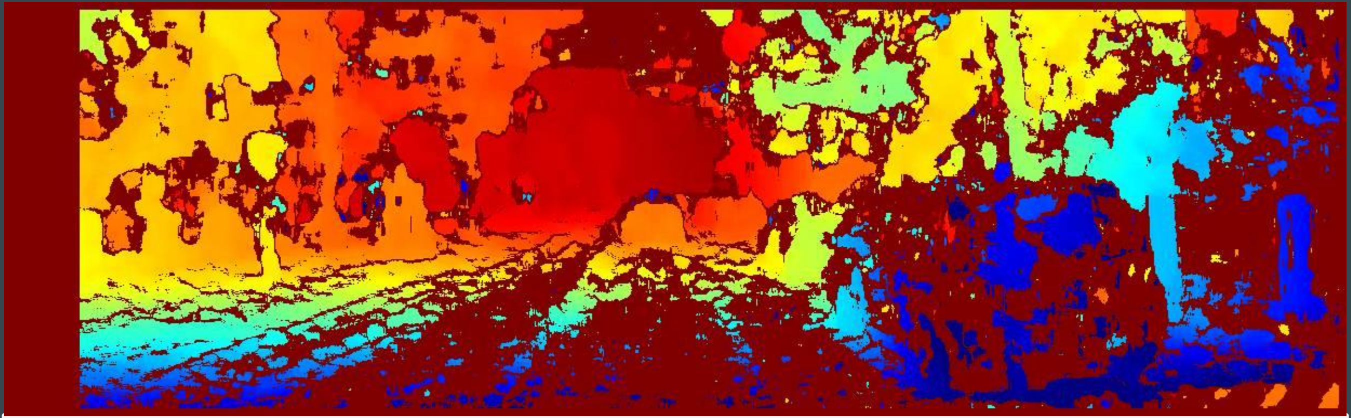


Disparity map

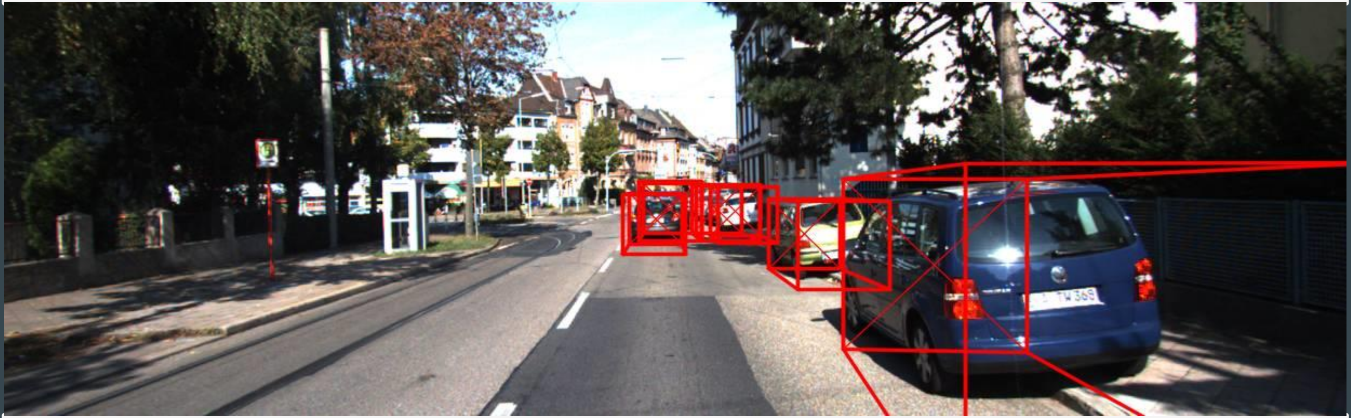


3D bboxs

Image 1 based on Geometry method

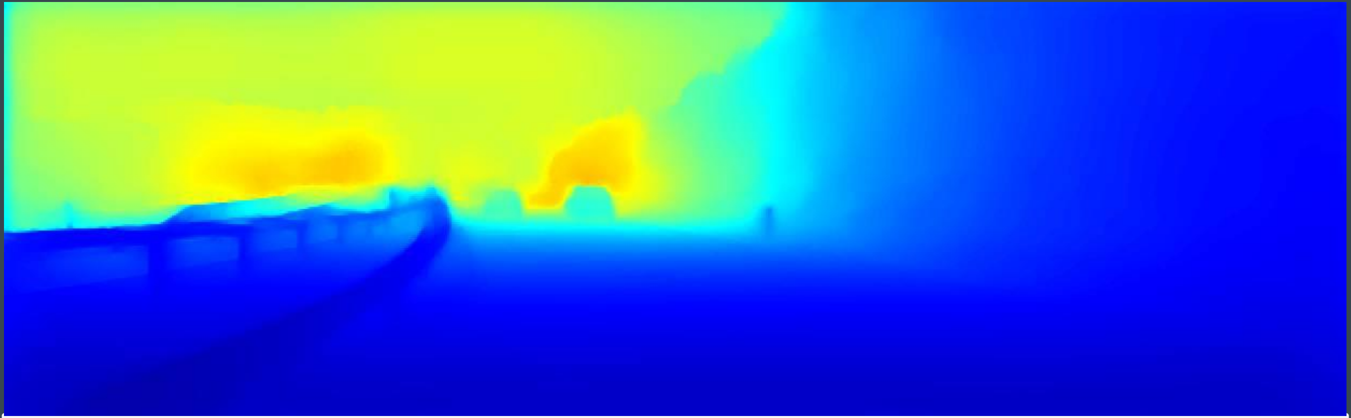


Disparity map

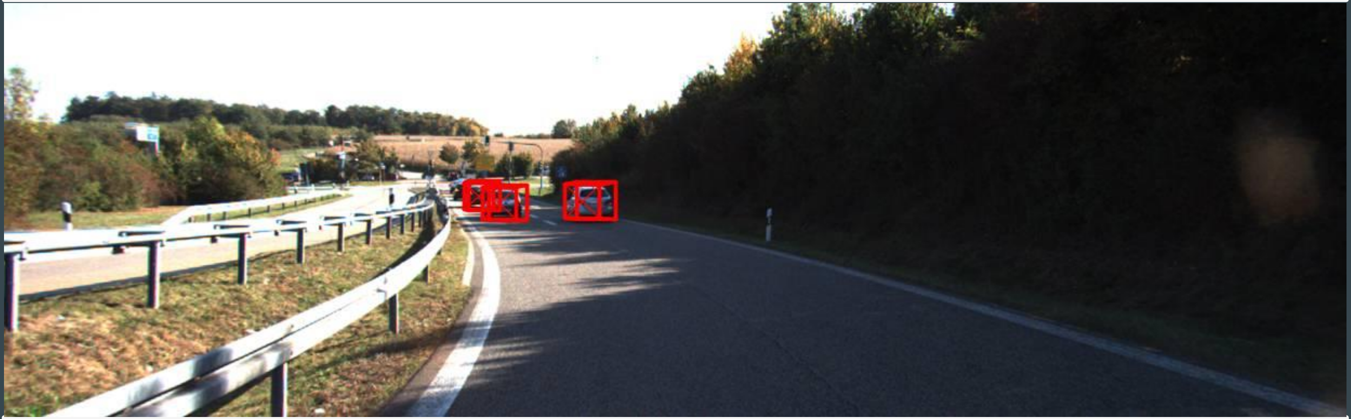


3D bboxs

Image 2 based on Deep learning

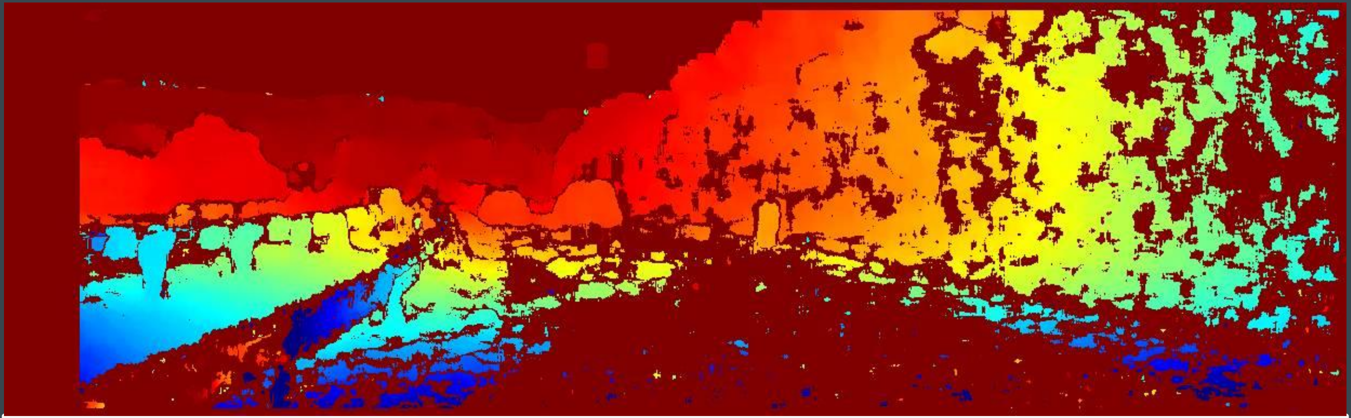


Disparity map

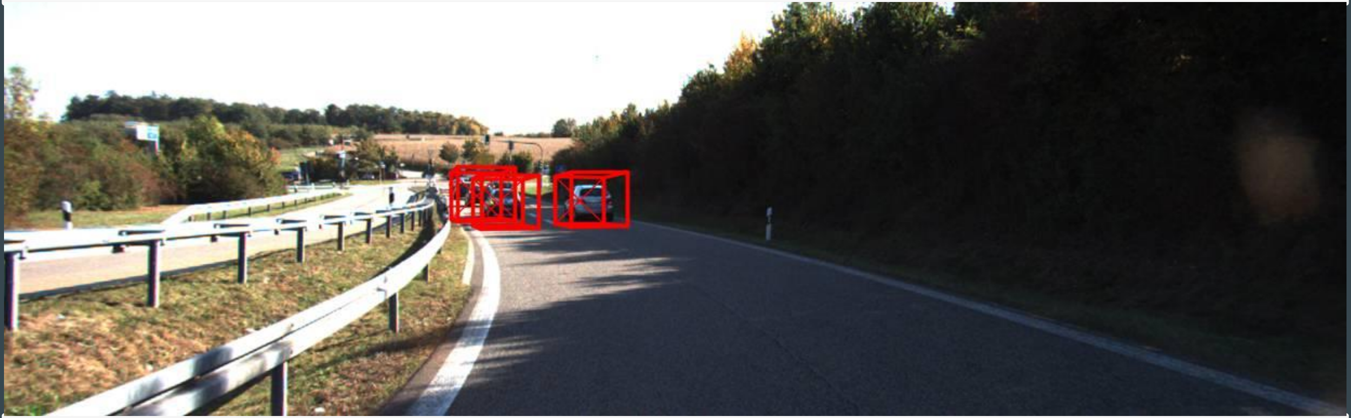


3D bboxs

Image 2 based on Geometry method

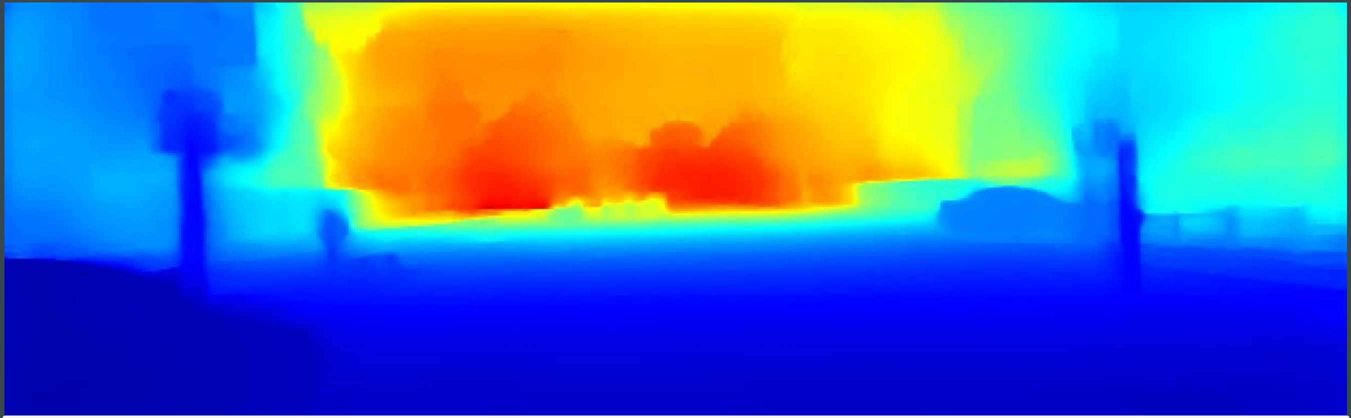


Disparity map

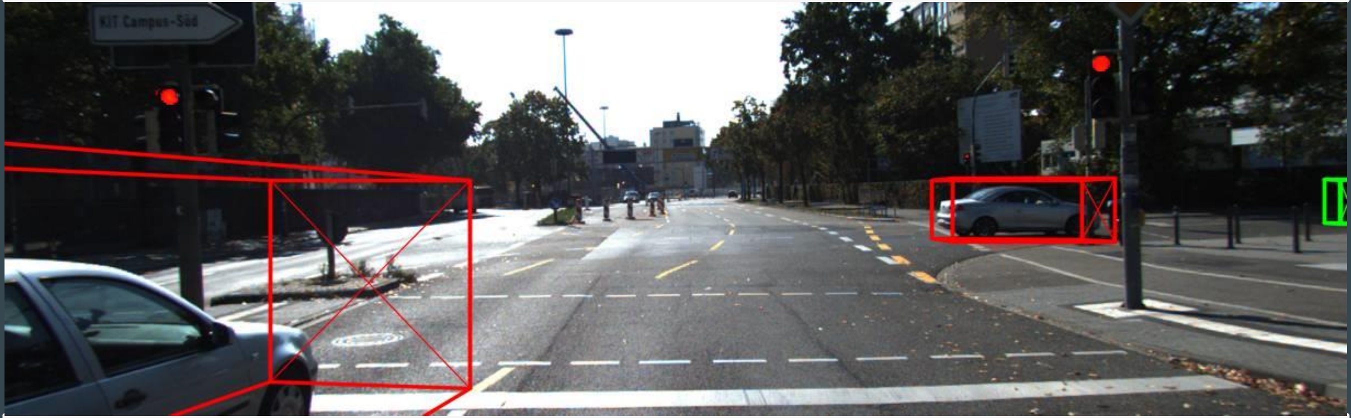


3D bboxs

Image 3 based on Deep learning

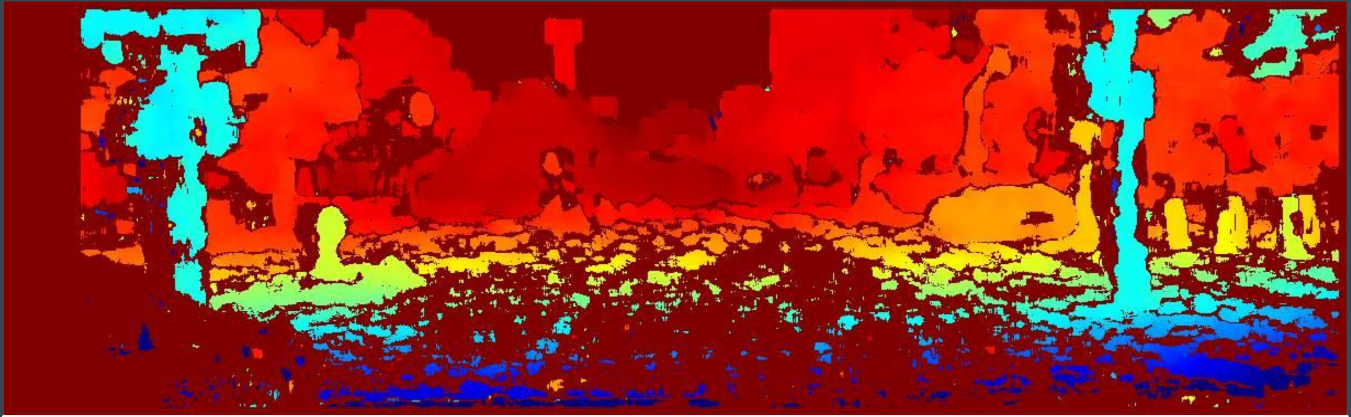


Disparity map

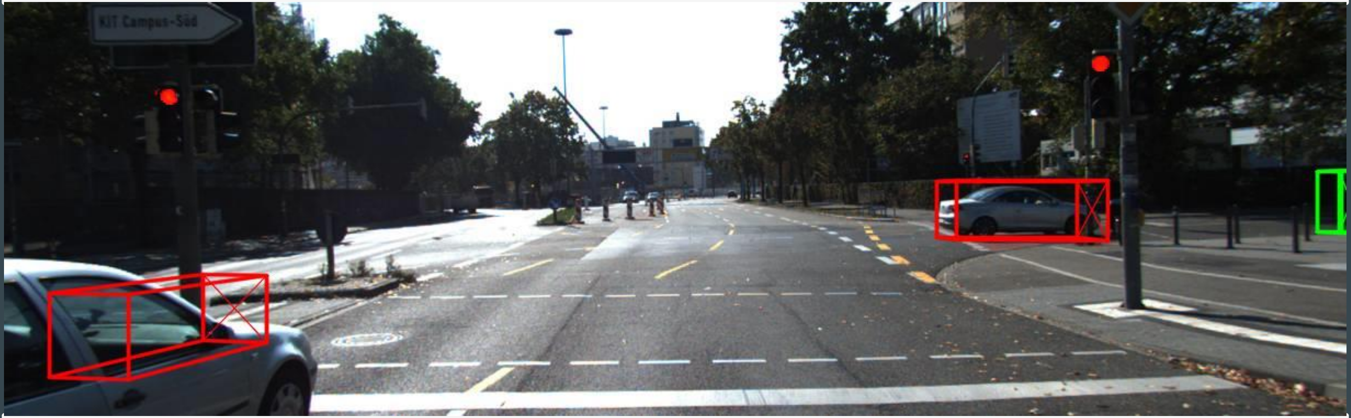


3D bboxs

Image 3 based on Geometry method

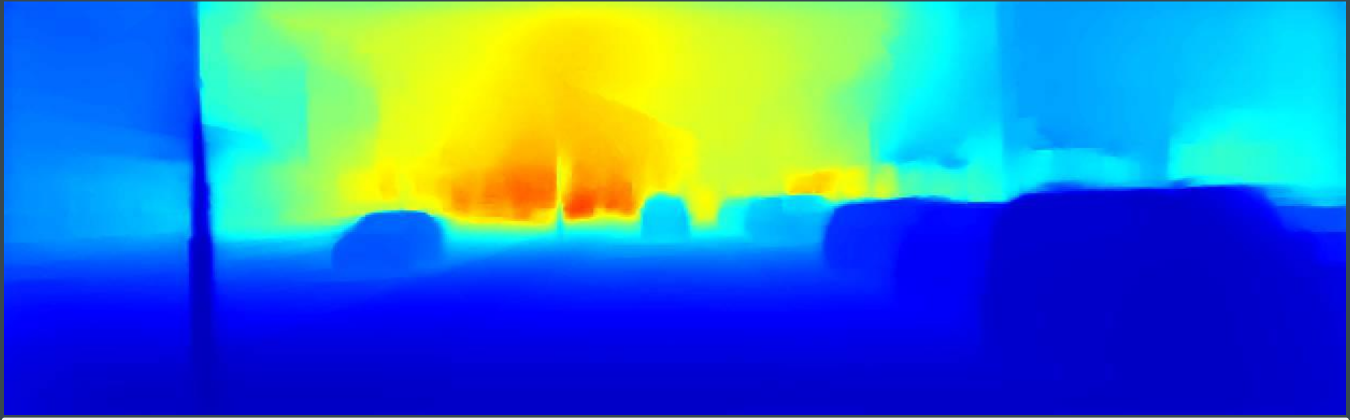


Disparity map

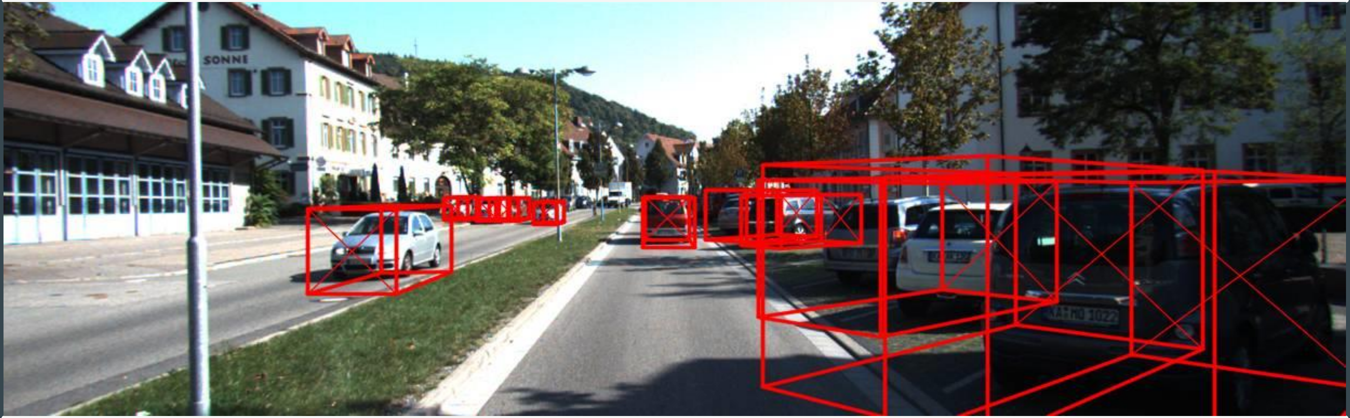


3D bboxs

Image 4 based on Deep learning

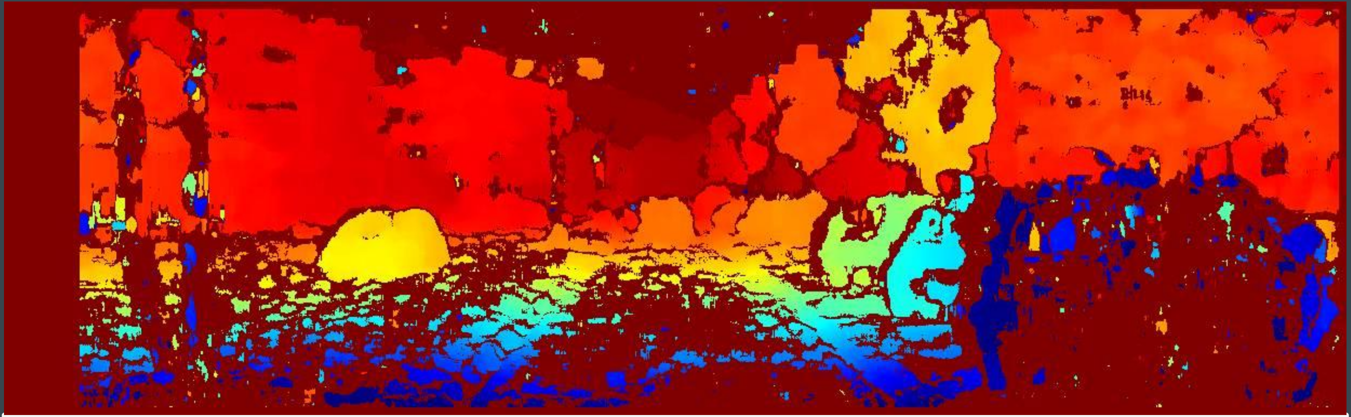


Disparity map

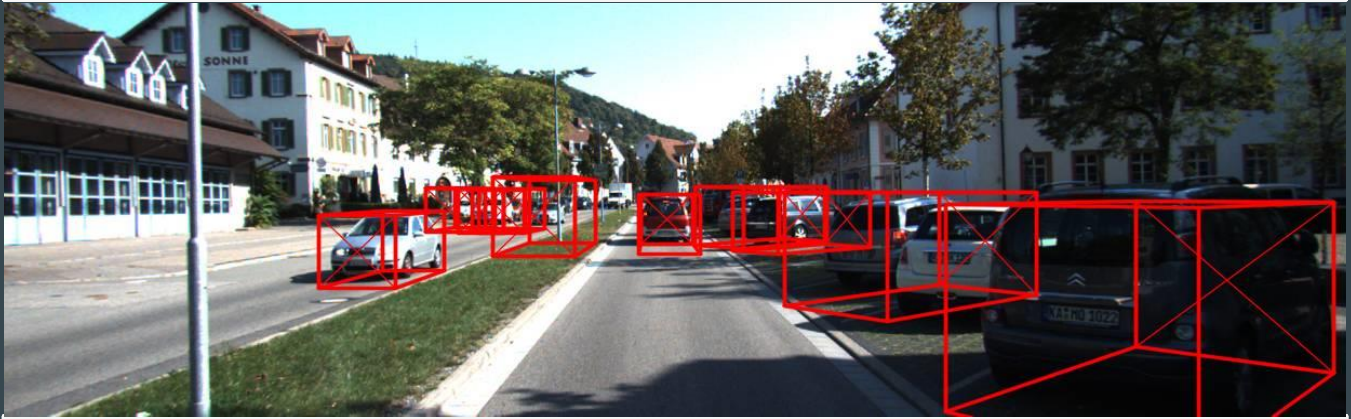


3D bboxs

Image 4 based on Geometry method



Disparity map



3D bboxs

Conclusion

Thank you for your listening

References

1. Alhashim, Ibraheem, and Peter Wonka. "High quality monocular depth estimation via transfer learning." arXiv preprint arXiv:1812.11941 (2018).
2. Geiger, Andreas, et al. "Vision meets robotics: The kitti dataset." The International Journal of Robotics Research 32.11 (2013): 1231-1237.