Exploration des algorithmes de deep-learning et Robotique pour la correspondance entre la vision et un espace 3D

YE LIU

June 2018

1 Introduction

The object of the project is to build a robot with 3d vision only by camera, with the algorithms of machine learning and robotic. Our object of the first step is to build a robot, which can move around by itself and avoid the obstacles only by the vision from one camera. Then it should be able to make the cartography of its surrounding. Finally, the robot can interact with the environment, complete some specially designed tasks, for an example, go find a bottle of Coca-cola and bring it to us.

To achieve our objective, a rigorous cartographic process is essential. At the beginning, we have decided to use only one camera, which will allow our system to be easily planted to other platform. And as the machine learning developing rapidly, we can have a satisfying result by monocular camera [7]. Our process of SLAM (Simultaneous localization and mapping) is from a former work CNN-SLAM[7] of *Tateno, Keisuke, et al.* The main process is shown in Figure 1.

First the system will choose key frame image as the robot functioning. These key frame images will be the input for depth prediction and semantic segmentation using deep learning algorithms. Every other input frame images will be used to refine the depth prediction and to realize a stable accuracy camera pose estimation. By combing the camera pose, depth prediction, segmentation and the original image, we will have a representation of the 3D world.

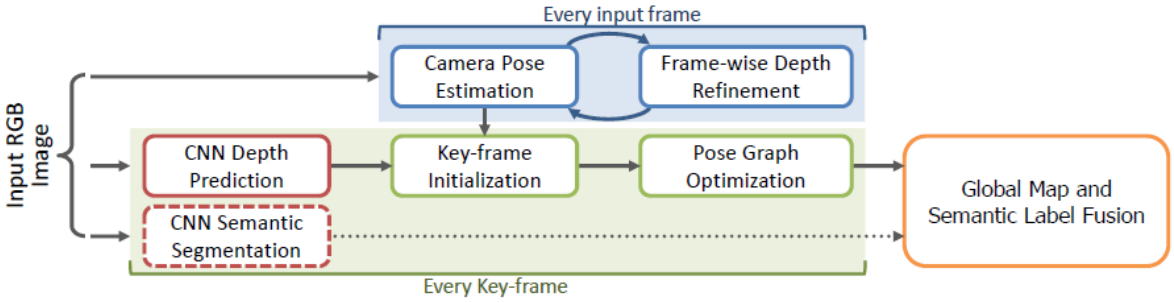

Figure 1: CNN_SLAM overview [7]

What is more, the real time implementation problems should be solved by multi-thread algorithms and algorithm optimization. Further, we decided to use bounding box other than the segmentation of pixel level, which will be more reliable, and easier to generate. Then we should develop an algorithm to predict the action area of the objects, which will allow the robot to interact with the surrounding.

1.1 Summary

Our first objective is to build a robot for sample indoor tasks with cartography. At the present, we have developed our own depth prediction and segmentation algorithm. However these preliminary results are not satisfying. So we move to the pretrained algorithms, in order to move to the next step and to have

a little demo first. We have obtain our preliminary camera pose estimation without refinement, which will be improved now. And we have our representation of bounding box of polyhedral. Now, we got our first robot kit, AlphaBot2 Pi. We are working with it to refine the depth and camera pose estimation, also we are working with robot mobility.

2 Depth prediction

To have the depth information from a single image, we decided to use a deep learning neural network to predict the depth map. First we would use a pretrained base network to encode the image, such as ResNet or VGG [6]. Then we should use up-sampling or a transposed convolutional network to decode the decoded image. The same principle should also be used for semantic segmentation.

2.1 Our work

We invested lots of former works in this field. Then we decided to use Keras as the front-end, which provides us the pretrained base networks. And we tested with up-sampling and transpose convolution separately. More precisely, we used the resnet50 as our basic encoder. Also we used the FPN(feature pyramid network [4]), we took 3 layers in the resnet50, which are the 40, 46 and 22. Then we combined these 3 layers with 3 transposed convolution layers as our decoder, Figure 2.

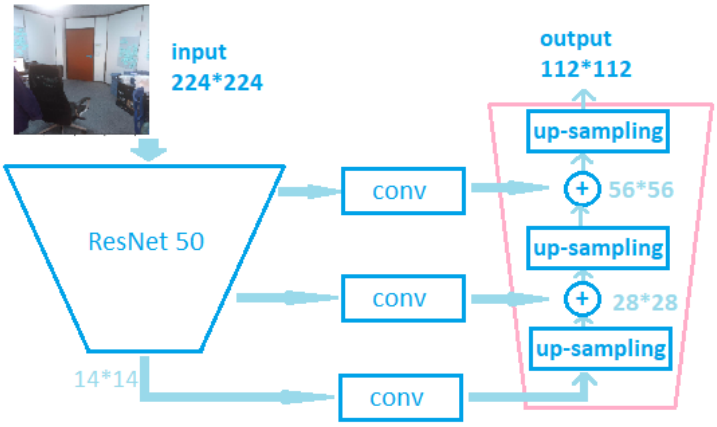

Figure 2: Our depth prediction architecture

We took NYU Depth Data-set V2 as our training set. To have a fast evaluation to our algorithm, we used about 1500 images, which is a small part of NYU Depth Data-set. We used our GPU (GeForce GTX 1080 Ti) and took 2 week for testing different neural networks. Here is the result with the image we took from our office in Figure 3.

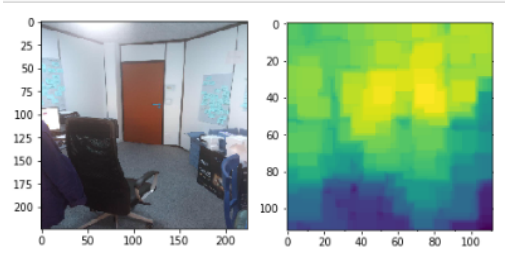

Figure 3: Depth prediction of our office

2.2 FCRN depth prediction

To save time and to quickly move to the next step, we decided to used the result from FCRN (Fully convolutional residual network [2]), which is better than our current result.

3 Semantic segmentation

3.1 Former works

Semantic segmentation is to understand the image at pixel level. There are traditional ways to solve it, such as graphic segmentation, selective search. Recently, lots of machine learning methods have appear. As FCN (fully convolutional network) in 2014, which use deconvoluional layers to up sample, and introduced skip connections to improve the coarseness of up-sampling [5]. Then, in 2015 dilated convolution is proposed, which is designed for dense predictions [8]. Then three versions of DeepLab are proposed [1], which are using dilated convolution, ASPP (atrous spatial pyramid pooling), and fully connected conditional Markov field. However, the dilated convolution layers are computationally expensive, and take too much memory. So another architecture is made by Lin, Guosheng, et al, the RefineNet [3]. These methods have their advantages and drawbacks, we should elaborate carefully to choose the best architecture for our project.

3.2 Our work

For our project, we would like to design an algorithm to make a segmentation for more objects, such as pens, bottles, or laptops. And it should be aimed at the environment in the office. Which means that maybe we should take our own dataset, or we should switch to an other method.

So we decide to use the deeplab V3 [1] for now, to complete a demo robot. In the later part of this project. We will come back to train our own semantic segmentation network, which will be specially designed for our problem.

4 Camera calibration

To realize our treatment to the images, we should first come to the calibration of the camera. This is to estimate the parameters of a lens and image sensor of an image or video camera, which will be used to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene.

There are mainly two parts in camera calibration. Firstly, to project the 3D world to the camera plan, secondly transform the image of the real world to pixel level.

$$Z\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}f_x&0&0\\0&f_y&0\\0&0&1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix},\begin{bmatrix}x_{pixel}\\y_{pixel}\\1\end{bmatrix}=\begin{bmatrix}\delta_x&0&c_x\\0&\delta_y&c_y\\0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

which can be rewrite as follow, where κ is the camera calibration matrix:

$$Z\begin{bmatrix}x_{pixel}\\y_{pixel}\\1\end{bmatrix}=\begin{bmatrix}\delta_x&0&c_x\\0&\delta_y&c_y\\0&0&1\end{bmatrix}\begin{bmatrix}f_x&0&0\\0&f_y&0\\0&0&1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix}=\begin{bmatrix}f_x\delta_x&0&c_x\\0&f_y\delta_y&c_y\\0&0&1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix}=\kappa\begin{bmatrix}X\\Y\\Z\end{bmatrix}$$

The distortion is represented as follows:

$$x_{distorted}=x(1+k_1r^2+k_2r^4+k_3r^6)$$

$$x_{distorted}=x+(2p_1xy+p_2(r^2+2x^2))$$

We used the calibration toolbox offered by opencv to calibrate our camera.

5 Camera pose estimation

With the calibration matrix and the undistortion process, we only lack the camera pose matrix P to get the relationship between the images and the real world, as follow:

$$z'\begin{bmatrix}x_{pixel}\\y_{pixel}\\1\end{bmatrix}=\begin{bmatrix}f_x\delta_x&0&c_x\\0&f_y\delta_y&c_y\\0&0&1\end{bmatrix}[R|t]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}=\kappa P\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

5.1 Feature points

We used the SIFT and SURF feature to describe the points of the images. And we suppose that the two images are not taken far away from each other, so we add the coordinates of the feature points as part of the description.

Then we match the point pairs by minimize the distance between the descriptions. Then we are going to use these matching pairs to calculate the transformation matrix between two camera poses. By combining the result with the depth prediction, we can also have the corresponding 3d points, which may be less precise.

5.2 Estimation by PnP

As we have the depth map, we can also use the perspective N point algorithm to calculate the transformation by the relationship between 2d points $\vec{x}$, and 3d corresponding points $\vec{X}$.

We set the camera pose matrix as : $P = \kappa \cdot \begin{bmatrix} R & t \end{bmatrix}$. so that:

$$z' \begin{bmatrix} \vec{x} \\ 1 \end{bmatrix} = P \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \vec{x} \\ 1 \end{bmatrix} \times (P \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix}) = 0$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times (\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \hat{X}) = \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} \hat{X}^T & 0_{1\times4} & 0_{1\times4} \\ 0_{1\times4} & \hat{X}^T & 0_{1\times4} \\ 0_{1\times4} & 0_{1\times4} & \hat{X}^T \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} = 0$$

Then we can solve P by single value decomposition.

5.3 Estimation from 3D to 3D points corresponding

Firstly, we used the algorithm to calculate an optimal 3D affine transformation between two 3D point. We using the function $opencv.estimateAffine3D$. However, we found that the result it calculated is not a rigid body transformation matrix. So as our corresponding points are not perfect, the result after multiply the matrix will end up with great distortion.

To fix this problem, we redirected our solution to rigid body. We went to find the optimal rotation and translation between corresponding 3d points. Firstly, we calculate the centroid of the two point clouds:

$$centroid_A = \frac{1}{N}\sum_{i=1}^{N} P_A^i, centroid_B = \frac{1}{N}\sum_{i=1}^{N} P_B^i$$

Then we used Singular Value Decomposition (SVD) to get the rotation matrix:

$$H = \sum_{i=1}^{N} (P_A^i - centroid_A)(P_B^i - centroid_B)^T$$

$$\begin{bmatrix} U & S & V \end{bmatrix} = SVD(H), R = VU^T$$

Finally, translation matrix is calculated:

$$t = -R \times centroid_A + centroid_B$$

5.4 RANSAC

For our least square problem the outliers may have a great influence on the result, which demand us to detect the outliers. So we decided to use the Random sample consensus algorithm, which is a learning technique to estimate parameters of a model by random sampling of observed data.

5.5 Summary

We have tested all of these result. Estimations by fundamental matrix and by PnP are not very accuracy. The algorithm by minimization of the re-projection error is not complete. And the rigid body 3d transformation with RANSAC shows the best result. So we use the result calculate by section 5.3 with RANSAC as our temporary transformation matrix result.

6 Bounding box 3D, represent real world

To achieve our first object, we need the robot to know the contour of different object. As our objects are not depended on the detail of the object architecture, and it will take a lot of effort to get a detailed 3D model (as depth prediction is not accuracy for details). We have decided to pay less attention to the detail, instead we use a bounding box to represent the object.

We are sure about the contour of the mask image and so as to the original image, so that we can extract the bounding box by calculating the maximum and the minimum value of the coordinates of the mask. However, we have not a very stable nor accuracy depth prediction for details, so we haven't a reliable depth for the bounding box. So we take a larger depth to make sure that the whole object is in the box, Figure 4.

Figure 4: Process of getting bounding box from segmentation mask and depth prediction

The following pictures (Figure 5) show our first demo of the representation of the world, and the video in `https://www.youtube.com/channel/UCr7SlXAxC6705D-miGIcxaA`. We detected and located the objects, made bounding boxes of them, with all the points information saved in other place. Then detect the floor, shown in the image. Further, we will develop algorithm to detect the walls. And make better representation with images taken from different angles.

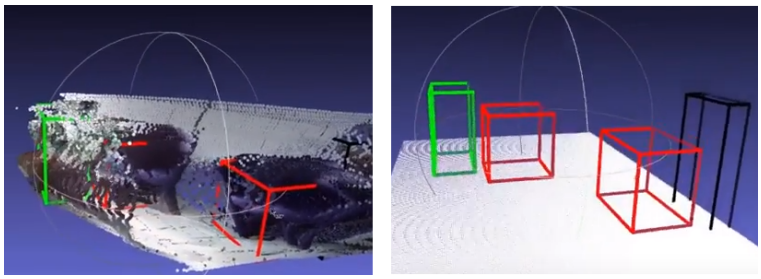

Figure 5: Demo of our desired representation of the world, (left) the 3d Meshlab point cloud with bounding box; (right) the bounding box and the floor.

7 robot mobility

Our first implementation is based on AlphaBot2 Pi kit, Figure 6. We used the HTTP protocol to communicate with the robot, and using the GPIO ports of the Raspberry Pi to control the robot. And we achieved our first objective, which is to program the robot to move and avoid the obstacles automatically.

Firstly, the robot will take the images around it, and predict the depth map by neural network. With these information, robot will build a 3D model of its surrounding. And the robot will find a random direction to go basing on the model.

Figure 6: AlphaBot2 Pi Kit

8 Problems we met

8.1 luminous problem

The luminosity is a problem for both the depth prediction and the segmentation, for example, the shadow on the ground. We are thinks about using histogram to reduce these influence, and we will make data augmentation based on the luminosity.

8.2 depth prediction

The prediction of the depth works bad for the contour of objects, the algorithm will make a plane at the place of the contour between two distant objects. And the depth predict bad as well, when the robot is close to the object.

8.3 depth and pose refinement

As the depth has many problems, we have to find a robust way to refine the depth prediction, and at the same time the estimation of the pose. Also we will introduce Kalman filter to get a reliable estimation of the position of the robot.

9 Conclusion

In the first half of the internship, we have made the depth prediction neural network, we evaluated different semantic segmentation algorithms. Then we studied the camera calibration problems and camera pose estimations. Finally, based on these result, w'e have succeed to make a 3D model of the surrounding of a robot, which allows the robot to move safely.

The next step will be studying and build a segmentation algorithm for our problem, and build a complete 3D model based on bounding boxes and triangles. And find a robust way to estimate camera pose. These will further allow the robot to do more proposed works.

References

[1] et al. Chen, Liang-Chieh. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv*, 2016.

[2] et al. Laina, Iro. Deeper depth prediction with fully convolutional residual networks. *3D Vision (3DV), 2016 Fourth International Conference on. IEEE*, 2016.

[3] et al. Lin, Guosheng. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv*, 2016.

[4] et al. Lin, Tsung-Yi. Feature pyramid networks for object detection. *CVPR*, 2017.

[5] Evan Shelhamer Long, Jonathan and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE conference on computer vision and pattern recognition*, 2015.

[6] Diogo Almeida Targ, Sasha and Kevin Lyman. Resnet in resnet: generalizing residual architectures. *arXiv*, 2016.

[7] et al. Tateno, Keisuke. Cnn-slam: Real-time dense monocular slam with learned depth prediction. *arXiv preprint arXiv:1704.03489*, 2017.

[8] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv*, 2015.