A Technical Report on Deep Visual Homing through Omnidirectional Camera

Lei Tai,[1*] Ming Liu,[2]

[1]MBE, City University of Hong Kong
[2]ECE, Hong Kong University of Science and Technology.

[*]E-mail: lei.tai@my.cityu.edu.hk

1 Introduction

Visual homing [*1–3*] is a basic visual navigation ability for a mobile robot. It uses one pair of images for reference and current pose to navigate the mobile robot. There are several advantages of visual homing compared with traditional vision methods with metrics map: low computational and memory cost, low sensitivity to error accumulation, and lightweight planning. A challenge of the visual homing problem is the estimation of the homing vector, which is the direction from the current position of the robot to the target position. It was once calculated through visual servoing methods [*3*].

Deep learning related methods are leading the advantages of artificial intelligence areas. Convolutional neural networks (*CNN*) are widely used in vision tasks like image classification, object detection, and semantics segmentation. The success of deep learning is mainly attributed to the effective feature representation ability of the hierarchical model.

Considering the fact that traditional visual homing methods still cost a lot of effort of the model computation. We tried to design a model-free method through building a *CNN* model

which takes the images pair as input and take the homing vector as output directly.

2 Technical Contents

2.1 Method

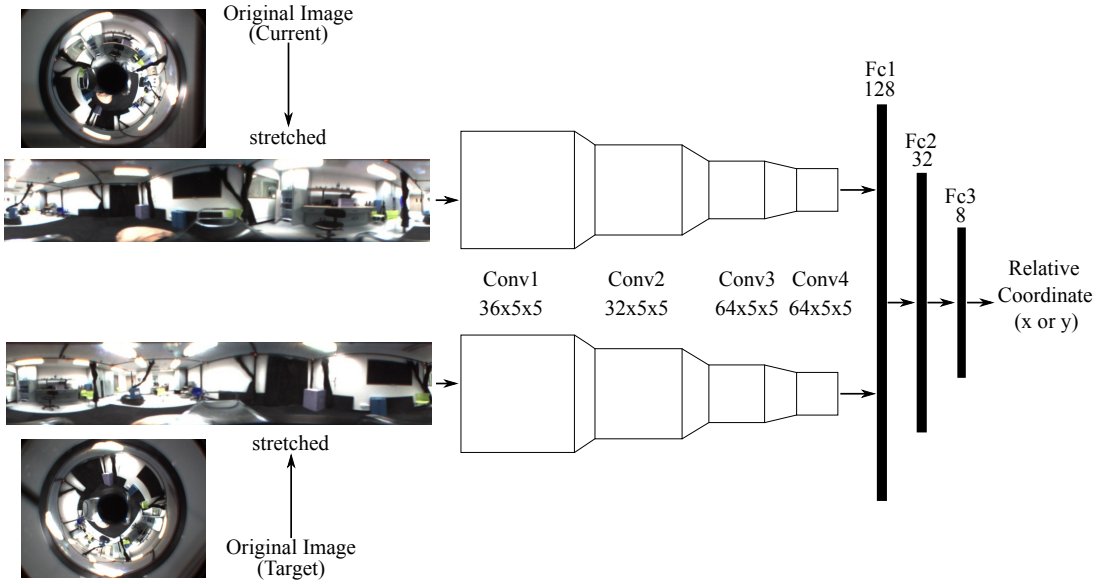

Figure 1: The network structure for the deep visual homing

As shown in figure 1, we firstly stretched the images taken from the omnidirectional camera to a square size (879x170 pixels). They were separately taken into a convolutional neural networks with 4 convolutional layers. Every convolutional layer here was a combination of a convolutional layer, pooling layer and rectified linear unit (ReLU) layer. After the convolutional process, the features extracted from the conv layers are concatenated and taken into 4 fully-connected layers. Every fully-connected layer also consists a ReLU layer. Finally, the relative coordinate between the two images is taken as the output. The loss is the euclidean distance between the label and the prediction. We separately trained two networks for the relative x or y coordinate.

2.2 Implementation

The coordinates of the images are taken from a motion track system which is accurate to centimeters. The range of the samples is in a $3 \times 3\ m^2$ indoor area. we took 2357 images and divided them into the training set (2000 images) and test set (357 images). In the training dataset, we choose one image as the target image for all of this set. And calculate the relative distances from all of the other images. The relative coordinate is still based on the original frame of the motion track system. The training was implemented in Caffe framework. It cost 7 hours to train the network to converge.

3 Discussion

The test result showed that the output of the relative distance is very precise based on the same target as the training dataset. However, when we randomly chose two images as input, the model cannot predict the right relative distance. The CNN model is not generalized to the untrained target.

References and Notes

1. M. Liu, C. Pradalier, Q. Chen, and R. Siegwart, "A bearing-only 2d/3d-homing method under a visual servoing framework," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4062–4067.

2. M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "Scale-only visual homing from an omnidirectional camera," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3944–3949.

3. M. Liu, C. Pradalier, and R. Siegwart, "Visual homing from scale with an uncalibrated omnidirectional camera," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1353–1365, 2013.