An iterative refinement method of point cloud for binocular vision 3D reconstruction

Junwei Hu, Jifeng Sun*, Yinggang Li, Qi Zhang, Shuai Zhao and Yibin Lin
School of Electronic and Information, South China University of Technology, Guangzhou, Guangdong, 510641, China
*Corresponding author’s e-mail: ecjfsun@scut.edu.cn

Abstract: This paper introduces a new binocular stereo deep learning network based on point cloud, which can realize higher precision point cloud reconstruction through continuous iteration of the network. Our method directly carries out point cloud processing on the target, calculates the difference between the current depth map and the real depth, estimates the loss according to the predicted point cloud and the information of the dual view input image, and then uses the appropriate loss function to iteratively process the point cloud. In addition, we can customize the number of iterations to achieve higher precision point cloud effect. The proposed network basically achieves good results on KITTI data set.

1. Introduction
Artificial intelligence has become the current trend of scientific and technological development, and has begun to involve all aspects of people's production and life, how to make computers have human visual perception has also become a research hotspot. Binocular vision is similar to the human vision system, which uses binocular camera to collect the image of the target object, and then identifies the location information of the target object through the operation and analysis. However, recognition is only a part of computer vision. Computer vision in the real sense can surpass the recognition and perception of three-dimensional environment. In order to interact and perceive, we must restore the world to three-dimensional space. Therefore, three-dimensional reconstruction technology is of great significance to computer vision. In computer vision, binocular vision 3D reconstruction refers to the process of reconstructing 3D information from two images. Because the information in a single image is incomplete, 3D reconstruction needs empirical knowledge, and binocular vision 3D reconstruction is relatively easy. The vision theory system proposed by Marr[1] lays a foundation for the development of modern computational vision and provides theoretical support for 3D reconstruction methods.

In this paper, we mainly propose a new network-based binocular stereo vision depth learning based on point cloud, which can achieve higher accuracy of point cloud reconstruction through continuous iteration of the network. When the target object is directly processed as point cloud, we can make the 3D high-resolution scene and realize more accurate 3D object reconstruction. Our framework is mainly implemented in two steps: first, the four-dimensional matching cost volume is constructed by connecting the feature volumes of the left and right images. The initial object surface is obtained from the scene by using a 3D convolution network, and then the initial point cloud information is obtained. Then, by predicting the difference between the depth information of the current point cloud and the real point cloud depth information, the loss is estimated according to the predicted point cloud and the information of the dual view input image, and then the point cloud is iteratively processed by using the appropriate
loss function for network training.

2. Related work

Recently, 3D reconstruction methods try to use a priori knowledge to construct 3D reconstruction problems as recognition problems. The development of deep learning technology and, more importantly, the availability of more and more large training data sets has led to a new generation of methods that can recover the three-dimensional geometry and structure of objects from one or more RGB images without complex camera calibration process. Although this is only recent, these methods have shown exciting and promising results in various tasks related to computer vision and graphics[2].

Szelisk et al.[3] first divided the early binocular vision algorithm into four steps: cost calculation, cost aggregation, parallax calculation and parallax optimization, and also proposed a complete set of evaluation criteria for binocular vision algorithm, which promoted the development of binocular vision[4,5]. Min et al.[6] enhanced the robustness of the algorithm by using multi-scale features containing rich information to construct the cost function. Jung et al.[7] used the conditional generation countermeasure network to estimate the depth of a single image, and used the generation network based on the combination of an encoder, decoder and refining network to obtain good experimental results on the objective data set. Zagoruyko et al.[8] proposed a matching similarity calculation network based on convolutional neural network. The network input is two image blocks, and the network directly outputs the similarity of the two image blocks. When the baseline distance is large, the stereo matching problem is solved. Zbontar and Lecun[9] proposed modeling binocular vision as a similarity problem, and intercepting some image blocks from the left and right original images to calculate the similarity; Mayer et al.[10] proposed an end-to-end stereo matching algorithm based on convolutional neural network for parallax calculation and optical flow estimation, which solves the problems of insufficient training data and poor generalization ability of stereo matching model. Pang et al.[11] proposed a Two-level Network CRL, which obtains the parallax map through multi-scale, then refines the parallax map through the idea of residual, and finally obtains the final result through multi-scale parallax fusion. Kendall et al.[12] horizontally shifted the features of the right image and fused them with the features of the left image to generate a four-dimensional feature cost volume. The network uses 3D convolution of encode decode structure to process the cost volume, generates parallax map in an end-to-end manner. Chang et al.[13] introduced spatial pooling pyramid technology into the feature extraction network, used multiple 3D codec modules to process joint features, and finally adopted the training method of deep supervision, which greatly improved the performance of the model.

3. Method

3.1. The network architecture

This section describes the framework structure of the network in detail, as shown in figure 1. According to the steps of three-dimensional reconstruction of binocular vision, our framework is mainly realized in two steps: first, we construct the four-dimensional matching cost by connecting the feature bodies of left and right images. The initial object surface will be obtained from the scene by using the 3D convolution network, and then the initial depth map information will be obtained. Then, the difference between the current point cloud depth information and the real point cloud depth information is predicted through the left and right images, the loss is estimated according to the predicted point cloud and the information of the dual view input image, and then the point cloud is iteratively processed by using the appropriate loss function for network training.
Figure 1. Overview of network architecture. First, a rough depth map is predicted with low computational cost and then projected into the point cloud without the hypothetical point cloud. For each point, features are extracted dynamically from left and right image features. Iterative depth map module uses features to enhance point cloud to predict depth differences and iteratively refine depth maps.

3.2. Initial Depth Map
The first step of our network is dense matching through the deep features of two input images. Here, we learn from the previous method of geometric rules of binocular vision camera. We use eight layers 2D CNN, in which the third and sixth layers are pooling layers. Two convolution layers are used to extract image features in each scale. Except the last layer, there is a batch normalization layer (BN) and a correction linear unit (ReLU) behind each convolution layer, which makes the extracted feature mapping significantly improve the quality of reconstruction.

Then, the 3D cost is constructed from the extracted image features and the input camera. We use $I_i (i = 1, 2)$ to represent the input image. $K_i, R_i, t_i$ represent the camera interior, rotation and translation corresponding to the feature image respectively. Let $n_1$ be the principal axis of the left-view camera, the homography is expressed by a $3 \times 3$ matrix:

$$H_i(d) = K_i \cdot R_i \cdot \left( I - \frac{(t_1 - t_i) \cdot n_1^T}{d} \right) \cdot R_i^T \cdot K_i^T$$

(1)

Homography transforms all feature maps into different parallel planes of the reference camera to form two feature volumes. As the core step connecting two-dimensional feature extraction and three-dimensional regularization network, the end-to-end training of depth map reasoning is realized in a differentiable way. Next, we synthesize multiple feature volumes into a cost volume $C$. in the two input views, let $W, H, D$ and $F$ be the width, height, depth samples of the input image and the number of channels of feature mapping, and $V$ is the size of the feature volume:

$$V = \frac{W}{4} \cdot \frac{H}{4} \cdot D \cdot F$$

(2)

$$C = \frac{\sum_{i=1}^{2} (V_i - \bar{V_i})^2}{2}$$

(3)

Then, we apply softmax operation to normalize the probability along the depth direction.

Finally, the simplest way to retrieve depth map $D$ from the probability volume $P$ is the pixel-wise winner-take-all[14]. We calculate the expected value along the depth direction, that is, the probability weighted sum of all assumptions:
Where $P(d)$ is the probability estimation of all pixels on depth $d$. This operation is also called soft argmin[12]. It is completely differentiable and can approximate the argmax result. Depth hypothesis $[d_{min}, d_{max}]$ in the process of cost construction, the expected value here can produce a continuous depth estimation, and the output depth map is the same size as the two-dimensional image feature map.

3.3. Iterative Calculation

In this section, we use iterative calculation to predict the depth difference. We update the information of point cloud after each iteration and extract point features from image features. Through this method, the features extracted at each voxel are determined by the fixed space partition of the scene. The iterative computing network framework is shown in the figure 2.

Firstly, the input is a feature point cloud, and then the point features of different scales are extracted through two edge convolution layers. Secondly, the point features are transformed through multi-layer perceptron (MLP), and a probability scalar softmax is output at each point. Finally, the depth prediction of the difference can be obtained. In addition, we can also set the number of iterations to continuously reduce the depth difference between the predicted point clouds, so as to capture more detailed features through closer point assumptions to predict more accurate depth information.

![Figure 2. The iterative computing network framework.](image)

3.4. Training Loss

We consider the loss of initial depth map and iterative refinement depth map. The problem is treated as a regression task, and the loss is used to train the network to measure the absolute difference between the predicted depth map and the real depth map:

$$\text{Loss} = \sum_{p \in P_{\text{val}}} \| D(p) - \hat{D}_i(p) \|_1 + \lambda \cdot \| D(p) - \hat{D}_r(p) \|_2$$

(5)

Where $P_{\text{valid}}$ valid is the real pixel set, $D(p)$ is the real depth value of the pixel $\hat{D}_i(p)$ is the initial depth estimation, $\hat{D}_r(p)$ is the refined depth estimation. And $\lambda$ is set to 1.0 in the experiment.

4. Experimental Analysis

4.1 Training

In this experiment, we mainly use the binocular RGB images in Scene Flow and KITTI data sets as the input of network training. KITTI dataset[15] was mainly proposed by Karlsruhe Institute in Germany and is widely used to evaluate algorithms in computer vision fields such as stereo vision and target detection. The data set consists of 194 training images and 195 test image pairs with a resolution of 1240×376. Each image pair is corrected, that is, the way of conversion, so that an object appears in two
images in the same vertical position. A rotating laser scanner, mounted behind the camera on the left, provides the true depth of the ground. The real differences in the test set are retained and an online ranking is provided, on which researchers can evaluate their methods. The goal of KITTI stereo data set is to predict the parallax of each pixel on the left image. Scene Flow is a large synthetic data set, including 35454 training image pairs and 4370 test image pairs. The ground truth disparity map is densely provided, which is enough to directly train the depth learning model. In the data preprocessing, the depth map is generated according to the given ground truth point cloud. During the training, we set the resolution of the input image to 640×512, number of views n=2. For the deep refinement step, we set the number of iterations L=2, use the initial learning rate of 0.0005, and the decrease in each two stages is 0.9. Firstly, the initial prediction network is trained in 4 stages, and then the model is trained in 12 stages.

4.2 Evaluation

We mainly use two method criteria to evaluate the difference between our method and other methods: 3-Pixel-Error, which represents the percentage of pixels with a difference of more than 3 pixels between the predicted parallax and the real parallax. The error is measured by the percentage of pixels with a difference of more than 3 pixels between the real parallax and the predicted parallax, which is converted into depth, which means, for example, The error of an object 2 meters away from the camera is 3 cm, and the error of an object 10 meters away from the camera is 80 cm. End-Point-Error (EPE), that is, the average difference between the predicted difference and the real difference.

| Method         | EPE (%)  |
|----------------|----------|
| PCBP[17]       | 4.04%    |
| PR-Sf+E[16]    | 4.02%    |
| StereoSLIC[17] | 3.92%    |
| PCBP-SS[17]    | 3.40%    |
| SPS-St[18]     | 3.39%    |
| Ours           | 3.15%    |

Table 1: Compares the 3PE of stereo algorithms on this dataset.

| Method            | Scene Flow EPE (px) |
|-------------------|---------------------|
| MC-CNN (Zbontar and LeCun 2016) | 3.79 |
| GC-Net (Kendall et al. 2017) | 2.51 |
| iResNet-i2 (Kendall et al. 2017) | 1.40 |
| SegStereo (Yang et al. 2018) | 1.45 |
| PDS (Tulyakov, Ivanov, and Fleuret 2018) | 1.12 |
| Ours              | 1.08    |

Table 2: EPE of stereo algorithm on Scene Flow is compared.

As can be seen from table 1 and table 2, the results of our method on the data set show that the accuracy of our method is significantly improved compared with other methods. Our network is a good method suitable for computing binocular vision 3D point cloud reconstruction.

5. Conclusion

In this paper, we mainly propose a new point cloud network based on binocular vision, which can achieve higher accuracy of point cloud reconstruction through continuous iteration of the network. When
using the target object directly as a point cloud processing. Our method directly carries out point cloud processing on the target, calculates the difference between the current depth map and the real depth, estimates the loss according to the predicted point cloud and the information of the dual view input image, and then uses the appropriate loss function to iteratively process the point cloud. In addition, we can customize the number of iterations to achieve higher precision point cloud effect. We believe that our approach will help facilitate deep learning research for challenging visual tasks such as stereo, flow, and scene flow estimation.

Acknowledgments
The work was supported by the National Natural Science Foundation of China No.62071183.

Reference
[1] D.Marr. (1982) Vision: A computational Investigation into the human representation and processing of visual information. W. H. Freeman and Company.
[2] Xian-Feng Han, Hamid Laga. (2019) Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. IEEE CVPR.
[3] Kang S B, Szeliski R, Chai J. (2003) Handling occlusions in dense multi-view stereo. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
[4] Hirschmuller H, Scharstein D. (2007) Evaluation of Cost Functions for Stereo Matching. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
[5] Scharstein D, Hirschmüller H, Kitajima Y, et al. (2014) High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. German Conference on Pattern Recognition.
[6] Min D, Sohn K. (2008) Cost Aggregation and Occlusion Handling with WLS in Stereo Matching. IEEE Transactions on Image Processing a Publication of the IEEE Signal Processing Society, 17(8):1431-42.
[7] JUNG H, KIM Y, MIN D, et ai. (2017) Depth prediction from a single image with conditional adversarial networks. IEEE Transactions on IEEE International Conference on Image Processing (ICIP), 1717-1721.
[8] ZAGORUYKO S, KOMODAKIS N. (2015) Learning to compare image patches via convolutional neural networks. IEEE Transactions on Computer Vision and Pattern Recognition, 4353-4361.
[9] Zbontar J, Lecun Y. (2015) Computing the stereo matching cost with a convolutional neural network. Proceedings of the IEEE conference on computer vision and pattern recognition, 1592-1599.
[10] Mayer N, Ilg E, Hausser P, et al. (2016) A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4040-4048.
[11] Pang J, Sun W, Ren J S J, et al. (2017) Cascade residual learning: A two-stage convolutional neural network for stereo matching. Proceedings of the IEEE International Conference on Computer Vision, 887-895.
[12] Kendall A, Martirosyan H, Dasgupta S, et al. (2017) End-to-end learning of geometry and context for deep stereo regression. Proceedings of the IEEE International Conference on Computer Vision, 66-75.
[13] Chang J R, Chen Y S. (2018) Pyramid stereo matching network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5410-5418.
[14] Collins, R.T. (1996) A space-sweep approach to true multi-image matching. Computer Vision and Pattern Recognition.
[15] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R.. (2013) Vision meets robotics: The KITTI dataset. International Journal of Robotics Research (IJRR).
[16] Vogel, C., Schindler, K., and Roth, S.. (2013) Piecewise rigid scene flow.” In Computer Vision (ICCV), IEEE International Conference on.
[17] Yamaguchi, K., Hazan, T., McAllester, D., and Urtasun, R.. (2012) Continuous markov random fields for robust stereo estimation.
[18] Yamaguchi, K., McAllester, D. and Urtasun, R. (2014) Efficient joint segmentation, occlusion labeling, stereo and flow estimation, in Computer Vision–ECCV.