Proceedings of the 37th Chinese Control Conference
July 25-27, 2018, Wuhan, China

Dense Depth Estimation with Absolute Scale

Xing Jin, Zhiwen Yao, Jingjing Zhang*
1. School of Automation, China University of Geosciences, Wuhan 430074, China
2. Hubei key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China
E-mail: work.zhang@cug.edu.cn

Abstract: Considering the difficulties in estimating depth from single image, in this paper, we propose a method to obtain the absolute scale depth map by combining the convolution neural network and depth filter. We compute relative transformation between consecutive frames by direct tracking features, which are extracted from RGB images and whose depthes are predicted by deep network, and then optimize relative motion by searching for a better feature alignment in epipolar line, and finally update every pixel depth of the reference frame by depth filter. We evaluate the proposed method on the open dataset comparison against the state of the art in depth estimation to evaluate our method.

Key Words: Depth Estimation, Pose Estimation, Depth Filter

1 Introduction

Depth map is a discipline as old as computer vision and encompasses several techniques that have been developed throughout the years. One of the most successful among these techniques is Simultaneous Localization and Mapping (SLAM) [1] which leverages camera motion to estimate camera poses in different moments and, in turn, estimate depth via triangulation from pairs of consecutive views. Alternatively to motion, other efficient assumptions can be used to estimate depth, such as variations in illumination [2] or focus [3].

In absence of such environmental assumptions, depth estimation from a single image of a generic scene is an ill-posed problem, due to the inherent ambiguity of mapping an intensity or color measurement into a depth value. Hence, much prior work on estimating depth is based on stereo images or motion [4] while provided accurate image correspondences, depth can be recovered deterministically in the stereo case [5]. But its always complicated to find association between each pixels of stereo thus depth map based on stereo or motion is usually sparse [6] or semi-dense [7]. As for motion case, there is a main limitation that even if pose estimation and scene reconstruction are carried out accurately, the absolute scale of such reconstruction remains inherently ambiguous.

By contrast, depth estimation from a single image requires the use of monocular depth cues such as line angles and perspective, object sizes, image position, and atmospheric effects whereas local disparity is sufficient for motion. Recently, Convolutional Neural Network (CNN) has been employed to learn an implicit relation between color pixels and depth [8, 9, 10] which give a relatively high resolution and a good absolute accuracy even under the absence of monocular cues (texture, repetitive patterns). The major limitation of such depth maps is the fact that, although globally accurate, depth borders tend to be locally blurred.

For this reason, a monocular SLAM called CNN-SLAM [11] has been proposed which fuses together depth prediction via deep networks and direct monocular depth estimation to yield a dense scene reconstruction that is at the same time unambiguous in terms of absolute scale and robust in terms of tracking. This monocular SLAM system recovers blurred depth borders by using the CNN predicted depth map as initial guess for dense reconstruction and refines the depth by means of small-baseline stereo matching [12]. Importantly, small-baseline stereo matching holds the potential to refine edge regions on the predicted depth image, which is where they tend to be more blurred.

In this paper, we also fuse together depth prediction via a deep networks and direct monocular depth estimation to avoid inherently ambiguous of monocular SLAM and blurred borders of depth prediction via deep networks. But what is different is that we use depth filter similar to the one in [7] to refine depth map carried out by CNN which is closer to the true depth distribution. Instead of minimizing the photometric error on all pixel lying within high color gradient regions, we adopt sparse model-based image alignment algorithm for pose estimation to increase speed.

For evaluation we use the RGB-D benchmark provided by the Technical University of Munich [13]. We validate our method with comparison on this public benchmark against the state of the art in depth estimation, focusing on the efficiency and accuracy of reconstruction.

2 Proposed Method

The proposed method contains three parts. (1) The pose estimation process, in which predicted depth from deep network is used as initial guess to compute relative transformation between two consecutive frames by minimizing the intensity residuals. (2) The pose optimization, in which we search for a patch alignment on the epipolar line of current frame to get best feature alignment, and then use the feature alignment to optimize camera pose. (3) The depth filter, in which we update every pixels depth value of the reference frame by depth filter, in the process current observation is integrated into the last estimate until the depth value converges.

This work was supported by the National Key Research and Development Plan of China (2016YFF010020002), the National Key Scientific Instrument and Equipment Development Projects of China (2012YQ0901670102), the National Natural Science Foundation of China (NSFC) (61604135), and the China Postdoctoral Science Foundation (2016M602285).

2.1 Pose estimation

The intensity image collected at timestamp k is denoted with I_k and the pixel x = $(x, y)^T$. We reconstruct a 3D point p = $(X, Y, Z)^T$ from its pixel coordinates and a corresponding depth measurement $Z = \mathcal{Z}(\mathrm{x})$ using the inverse projection function π^{-1}, i.e.,

$$p = \pi^{-1}(\mathrm{x}, Z) = (\frac{x - o_x}{f_x} Z, \frac{y - o_y}{f_y} Z, Z)^T \tag{1}$$

where f_x, f_y are the focal lengths and o_x, o_y are the coordinates of the camera center in the standard pinhole camera model which can be known from camera calibration, and Z is acquired by deep network. At the same time, The pixel coordinates for a point can be computed using the projection function π:

$$\mathrm{x} = \pi(\mathrm{p}) = (\frac{X f_x}{Z} + o_x, \frac{Y f_y}{Z} + o_y)^T \tag{2}$$

The camera pose (position and orientation) at timestamp k can be expressed with the rigid-body transformation $T_k \in SE(3)$. The relative transformation between two consecutive frames can be computed with $T_{k,k-1} = T_k \cdot T_{k-1}^{-1}$. During the optimization, we need a minimal representation of the transformation and, in the other way, the transformation matrix $T_{k,k-1}$ is an over-parametric representation of pose, i.e., $T_{k,k-1}$ has twelve parameters where pose only has six degrees of freedom. Therefore, usually use the Lie algebra $se(3)$ corresponding to the tangent space of $SE(3)$ at the identity. We denote the algebra elements, also named twist coordinates, with $\xi = (\omega, \nu)^T$, where ω is called the angular velocity and ν the linear velocity. The twist coordinates ξ are mapped to $SE(3)$ by the exponential map, i.e.,

$$T(\xi) = \exp(\widehat{\xi}) \tag{3}$$

The sparse model-based image alignment algorithm [7] for the rigid body transformation $T_{k,k-1}$ between two consecutive camera poses is defined by minimizing the intensity residuals which is defined by the photometric difference between pixels observing the same 3D point. It can be computed by back-projecting a 2D point x from the previous image T_{k-1} and subsequently projecting it into the current camera view:

$$\gamma_{\mathcal{I}} = \mathcal{I}_k(\pi(T_{k,k-1}\pi^{-1}(\mathrm{x}, Z))) - \mathcal{I}_{k-1}(\mathrm{x}) \tag{4}$$

We seek to determine the motion ξ^* by maximizing the probability given the pixel-wise error, i.e.,

$$\xi^* = \arg\max \mathrm{p}(\xi|\gamma_{\mathcal{I}}) \tag{5}$$

After applying Bayes rule, assuming all errors are independent and identically distributed (i.i.d.) and using the negative log-likelihood, we get

$$\xi^* = \arg\min -\log(\mathrm{p}(\gamma_{\mathcal{I}}|\xi)) - \log(\mathrm{p}(\xi)) \tag{6}$$

In case $\log(\mathrm{p}(\gamma_{\mathcal{I}}|\xi))$ is defined as a Gaussian distribution, this results in a standard least-squares problem. In practice, the distribution has heavier tails due to occlusions and, a robust t-distribution $\mathrm{p}_t(0, \sigma^2, \nu)$ with zero mean, variance σ^2 and ν degrees of freedom is proposed in [14]. For the sake of simplicity, we assume in the intensity residuals are normally distributed with unit variance as follow:

$$\xi^* = \frac{1}{2}\arg\min \sum \gamma_{\mathcal{I}}^2 \tag{7}$$

For this nonlinear problem, we solve it in an iterative Gauss-Newton procedure.

2.2 Pose optimization

The last step aligned the camera with respect to the previous frame. Through back-projection, the relative pose $T_{k,k-1}$ implicitly defines an initial guess for the feature positions of all visible 3D points in the new image. Due to inaccuracies in the 3D points positions predicted via CNN and, thus, the camera pose, this initial guess can be improved. Because of the inaccuracy of the pixel depth estimation, the pixels will have a certain deviation on the epipolar line. To achieve a higher correlation between the continuous frames, we search for a patch (2 x 2 pixel) on the epipolar line in the new image T_k that has the highest correlation with the reference patch.

In this process, we assume that the intensity of the patch in a continuous frame remains invariable. We use Normalized Cross Correlation (NCC) to measure the similarity between two patches, and NCC is defined as follows.

$$S(A, B)_{NCC} = \frac{\sum_{i,j} A(i,j)B(i,j)}{\sqrt{\sum_{i,j} A(i,j)^2 \sum_{i,j} B(i,j)^2}} \tag{8}$$

In order to obtain stronger robustness, we used the mean NCC to measure the similarity of the patches, so as to adapt to the changes in the exposure of the camera during the moving process.

According to established accurate feature correspondence between the continuous frames, we define a Bundle Adjustment (BA) to optimize the camera relative motion, which also known as reprojection error minimization, and we adopt Gauss-Newton procedure to solve the optimization problem.

2.3 Depth filter

In the previous step, we get preliminary pose estimation by minimizing the photometric error, and then through epipolar search to obtain accurate features alignment, and final minimize reprojection error with features alignment to obtain an accurate position estimate.

Once the precise relative pose is obtained, we can update the depth value of the reference frame through the depth filter. For efficiency reason, we adopt Gaussian model distribution instead of Gaussian-Uniform mixture model distribution in [7] to refine depth map. Assuming distribution of the pixel depth is $P(d) = N(\mu, \sigma^2)$, and the observation of depth values also obeys the Gaussian distribution, $P(d_{obs}) = N(\mu_{obs}, \sigma_{obs}^2)$. As we know that product of two Gaussian distributions still obeys the Gaussian distribution, so that the depth value after fusion is $N(\mu_{fuse}, \sigma_{fuse}^2)$ for which,

$$\mu_{fuse} = \frac{\sigma_{obs}^2 \mu + \sigma^2 \mu_{obs}}{\sigma^2 + \sigma_{obs}^2}, \sigma_{fuse}^2 = \frac{\sigma^2 \sigma_{obs}^2}{\sigma^2 + \sigma_{obs}^2} \tag{9}$$

3 Experimental Results

A set of experiments is demonstrated in this section to evaluate the performance of the proposed method. We pro-

vide here an experimental evaluation to validate the contributions of our method in terms of reconstruction accuracy and efficiency, by means of a quantitative comparison against the state of the art on public benchmark datasets.

The evaluation is carried out on a desktop PC with an Intel Core i5-3317U CPU at 1.7GHz with 8GB of RAM and an Nvidia GTX640M GPU with 2GB of VRAM. As for the implementation of our method, although the CNN network works on an input resolution of 304×228 [10], both the input frame and the predicted depth map are converted to 320×240 as input for all other stages. Also, the CNN based depth prediction is run on the GPU, while all other stages are implemented on the CPU.

We use sequences from benchmark datasets, i.e. the TUM RGB-D SLAM dataset [13] acquired with a Kinect sensor. Every dataset accompanies an accurate groundtruth trajectory obtained with an external motion capture system. We selected one of the datasets which is acquired about an office desk to evaluate our algorithm, and the part of the dataset is shown in figure 1.

Fig. 1: Some images of the dataset, the depth value of first image will be optimized by the rest of images

Making the first image as a reference frame, we firstly use the convolution neural network [10] to estimate the depth of the reference frame, and making this depth value as the initial estimate of the reference frame. And then estimate the relative motion between continuous frames using the estimated depth value, we extract the fast features in the image and minimize the intensity residuals of these features to estimate the relative motion, and the accuracy of relative motion is evaluated, as shown in figure 2, by features alignment in continuous frames.

Finally, we optimize the feature alignment by epipolar search, and using the optimized feature alignment to compute accurate relative motion, and then use depth filter to optimize the depth estimated via deep network. As figure 3 shows, compared with estimating depth through the convolution neural network directly [10], computing the relative motion by direct method, and then optimizing depth value according to the relative motion has much higher accuracy.

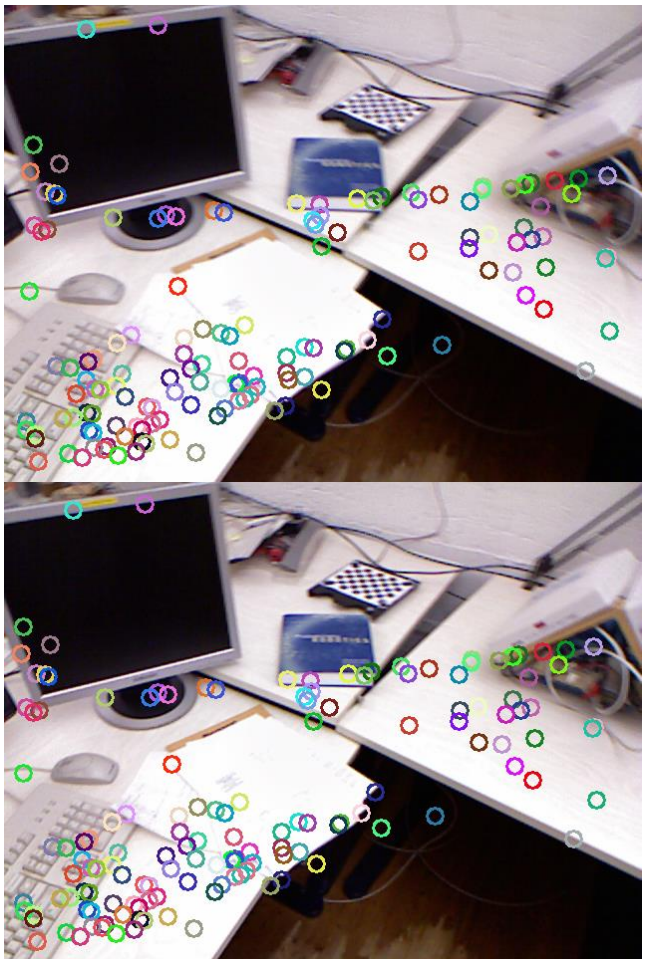

Fig. 2: We extract fast features in the reference frame and, then reproject these features to the current frame, if the relative motion is estimated precise, the features will have good alignment.

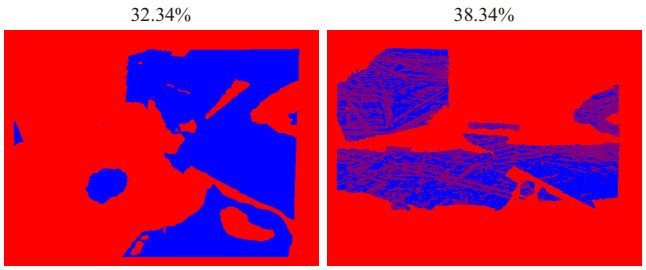

Fig. 3: Comparison with direct CNN-depth prediction(left image), the accuracy of depth estimation after refinement(right image) is much better. In which blue pixels depict correctly estimated depths, i.e. within 10% of ground-truth.

4 Conclusion

We have shown that the integration of depth filter with depth prediction via a deep neural network can not only obtain dense depth map with absolute scale that is inherent limitations of traditional monocular reconstruction, but also can improve the accuracy of CNN-predicted depth map. A future research avenue is represented by expanding this method to construct a global dense map for monocular SLAM.

References

[1] Cadena C, Carlone L, Carrillo H, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age[J]. *IEEE Transactions on Robotics*, 2016, 32(6):1309-1332.

[2] Prados E, Faugeras O. Shape From Shading[J]. Mathematical Models in Computer Vision the Handbook, 2009, 21(8):375-388.

[3] Suwajanakorn S, Hernandez C, Seitz S M. Depth from focus with your mobile phone[C]. Computer Vision and Pattern Recognition. IEEE, 2015:3497-3506.

[4] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International journal of computer vision, 2002, 47(1-3): 7-42.

[5] Hartley R, Zisserman A. Multiple view geometry in computer vision[M]. Cambridge university press, 2003.

[6] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.

[7] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]. Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014: 15-22.

[8] Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network[C]. Advances in neural information processing systems. 2014: 2366-2374.

[9] Garg R, BG V K, Carneiro G, et al. Unsupervised cnn for single view depth estimation: Geometry to the rescue[C]. European Conference on Computer Vision. Springer, Cham, 2016: 740-756.

[10] Laina I, Rupprecht C, Belagiannis V, et al. Deeper depth prediction with fully convolutional residual networks[C]. 3D Vision (3DV), 2016 Fourth International Conference on. IEEE, 2016: 239-248.

[11] Tateno K, Tombari F, Laina I, et al. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6565-6574

[12] Engel J, Sch?ps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]. European Conference on Computer Vision. Springer, Cham, 2014: 834-849.

[13] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]. Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012: 573-580.

[14] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014:2100-2106.