Optimization Algorithms for Feature Point Processing and Pose Recovery in 3D Reconstruction Systems (Postprint)

Authors: Xu Jianpeng, Bu Fanliang

Date: 2018-09-12T00:00:00+00:00

Abstract

Improving the accuracy of feature point detection and matching results, and better optimizing camera pose recovery outcomes, are key factors in enhancing the overall efficiency of 3D reconstruction. Based on the principles of the SIFT algorithm, we construct a novel algorithmic framework that employs FCN (fully convolutional networks) and BP (back propagation) neural networks, comprehensively considering influences from semantic segmentation of the image' s main target, image gray-level co-occurrence matrix, and other aspects, to achieve adaptive adjustment of feature point detection range and quantity. In the feature point matching stage, mismatches are eliminated by leveraging camera pose offset stability, while a graph optimization-based method is adopted for nonlinear optimization of the pose recovery results, yielding more accurate camera poses. Finally, through comparative analysis with existing mainstream algorithms, experimental results validate the effectiveness of the proposed algorithm, improving the scene adaptability of feature point detection and the accuracy of feature point matching and pose recovery, thereby achieving more efficient 3D reconstruction.

Full Text

Preamble

Feature Point Processing and Position Recovery Optimization Algorithm in 3D Reconstruction System

Xu Jianpeng, Bu Fanliang†

School of Information Technology & Cyber Security, People' s Public Security University of China, Beijing 102628, China

Abstract: Improving the accuracy of feature point detection and matching results and optimizing camera pose recovery are critical factors for enhancing the overall efficiency of 3D reconstruction. Based on the principles of the SIFT algorithm, this paper constructs a novel algorithmic framework that employs Fully Convolutional Networks (FCN) and Back Propagation (BP) neural networks. By comprehensively considering semantic segmentation of image main targets and gray-level co-occurrence matrices, the algorithm achieves adaptive adjustment of feature point detection range and quantity. During the feature point matching stage, it eliminates false matches by leveraging camera pose offset stability, while employing a graph-based optimization method for nonlinear pose refinement, yielding more accurate camera poses. Comparative analysis with existing mainstream algorithms demonstrates the effectiveness of the proposed approach, which improves scene adaptability in feature point detection, enhances feature point matching and pose recovery precision, and enables more efficient 3D reconstruction.

Key words: three-dimensional reconstruction; feature points; artificial neural networks; pose recovery; nonlinear optimization

0 Introduction

Feature point detection and matching constitute the most fundamental components of the entire 3D reconstruction pipeline. Their results serve as input data for epipolar geometry and triangulation steps to obtain camera poses and 3D coordinates of spatial point clouds. Camera pose represents one of the core data elements in the reconstruction process, reflecting the outcomes of the feature point processing stage on one hand, while serving as crucial input data for subsequent dense point cloud recovery and texture mapping steps on the other, thereby directly determining the final 3D reconstruction quality.

In existing mainstream feature point detection algorithms such as SIFT, SURF, and ORB, the detection range covers the entire image. However, 3D reconstruction typically focuses only on primary targets such as buildings or specific objects. When excessive edge points exist outside the main target or when similar pixel patches are present, false matches readily occur. Furthermore, current mainstream algorithms cannot control the number of feature points according to application requirements. An excessive number of feature points leads to numerous false matches that degrade pose recovery accuracy, while too few feature points fail to meet the data volume requirements for 3D reconstruction. Finally, conventional feature point matching optimization methods based on K-Nearest Neighbor (KNN) and RANSAC depend entirely on existing matching data, making them susceptible to cumulative errors from false matches and yielding unstable optimization results whose accuracy directly impacts pose recovery.

The rapid development of artificial neural networks in computer vision has provided new avenues for 3D reconstruction research. Hou et al. utilized the AlexNet model under the Caffe framework for feature extraction, achieving superior robustness and faster extraction compared to traditional features. McCormac et al. proposed SemanticFusion, a dense 3D semantic mapping method based on convolutional neural networks that predicts pixel-level object category labels to generate dense 3D semantic maps. Reference [9] employed convolutional neural networks for automatic image segmentation, effectively addressing the labor-intensive target segmentation problem in 3D reconstruction. Regarding the aforementioned issues, references [10,11] adjusted relevant coefficients based on actual feature point counts and image contrast factors to control feature point quantity, achieving favorable results. However, due to the limited factors considered, the algorithm' s adaptability remains inadequate. Reference [12] optimized feature point matching through improved BRIEF descriptors and epipolar geometry principles, improving matching accuracy but still suffering from insufficient stability due to false matches. Mainstream pose optimization methods first recover 3D point clouds and then perform nonlinear optimization using reprojection error, yet the point cloud recovery process accumulates errors from the pose recovery stage, resulting in insufficient precision.

To address these problems, this paper proposes a novel algorithm framework based on SIFT principles. First, we construct training datasets tailored to 3D reconstruction requirements to train an FCN neural network for detecting primary targets in images, thereby enabling adjustment of the feature point detection range. Simultaneously, we analyze the image gray-level co-occurrence matrix and combine it with the contrast threshold in SIFT to construct training data for a BP neural network. This network fits the nonlinear relationship between feature point quantity, image gray-level co-occurrence matrix, and contrast threshold to achieve adaptive control of image feature point numbers. Second, by analyzing triangulation principles, we leverage pose offset stability to optimize feature point matching results, constraining the pixel position range of matching feature points in images to avoid cumulative errors from false matches during optimization. Finally, we employ graph optimization methods to construct a least-squares problem between matching feature point pixel positions and camera poses for nonlinear pose refinement, fully capitalizing on the precision advantages of our feature point matching optimization algorithm. Experimental comparisons with SIFT demonstrate the effectiveness and efficiency of the proposed algorithm.

1.1 Neural Networks

A BP neural network is a multi-layer feedforward network trained using the error backpropagation algorithm. By calculating the error between expected and actual outputs, it adjusts node parameters and thresholds layer by layer from output to input through repeated iterations until achieving the target accuracy. FCN neural networks are primarily used for image semantic segmentation. Compared to conventional convolutional neural networks, the key difference lies in FCN' s final layer remaining a convolutional layer that performs upsampling to restore pixel classification and position information.

[Figure 1: see original paper]

1.2 Epipolar Geometry

Epipolar geometry addresses the fundamental problem of recovering camera spatial motion. As shown in Figure 2, for a 3D point P, cameras (optical centers) O_1 and O_2 respectively obtain images I_1 and I_2, with pixel representations p_1 and p_2. The line connecting the optical centers is called the baseline, and its intersection with the image planes yields epipoles e_1 and e_2. The lines connecting p_1 and p_2 to the epipoles are called epipolar lines l_1 and l_2. The position of p_2 is obtained by applying a spatial transformation $[R|t]$ to p_1. During feature point matching, we have already obtained several pairs of matching feature points, such as those generated by point P as p_1 and p_2. If the matching relationship were unknown, the corresponding point for p_1 could be anywhere, such as p_2', with the corresponding spatial coordinate being P'. Therefore, correct feature point matching is the most critical factor determining 3D reconstruction results.

[Figure 2: see original paper]

2 Algorithm Framework

Based on the above analysis, the proposed algorithm framework consists of three main components: feature point detection, feature point matching, and pose optimization, as illustrated in Figure 3.

[Figure 3: see original paper]

During the feature point detection stage, the algorithm calculates the contrast and gray-level co-occurrence matrix for each input image to analyze image complexity and determine the required number of feature points for the application. This information is fed into an adaptive control module based on a BP neural network, which adjusts the contrast threshold to control the overall number of feature points in the image. If the adjusted feature point count still does not meet requirements after the first iteration, further iterative adjustments are performed until compliance is achieved. Concurrently, the image undergoes processing by a trained FCN neural network to obtain the pixel range of the primary target, which is then used to constrain the feature point detection region and avoid cumulative errors from background noise.

In the feature point matching stage, we analyze the stability of camera pose offset in 3D reconstruction. During camera movement, the displacement between consecutive shots should not be too small, yet excessive displacement may cause feature disappearance or scene appearance changes. As shown in Figure 4, where Δe represents the error between recovered point cloud results (P') and ground truth (P) under the same matching error ($\Delta\theta$) but different camera displacements (t_1, t_2), with O_1 and O_2 representing camera optical center positions. Therefore, for adjacent images in a sequence, the pixel positions of corresponding matching feature points should not change drastically. Leveraging this property, we can optimize matching results by eliminating numerous false matches and improving reconstruction accuracy.

During the camera pose recovery stage, at least five matching point pairs are required to recover the camera pose R and t. For all optimized matching results (typically far exceeding five pairs), our algorithm analyzes the problem from a probabilistic perspective and employs graph optimization as the solution tool. We construct a least-squares problem between feature point pixel coordinates and camera poses, seeking the displacement transformation that most likely yields the current pixel positions of matching feature points in the second image from those in the first image. This approach enables nonlinear optimization of camera pose to obtain more realistic results.

3.1 BP Neural Network Construction and Training

The contrast threshold is a critical parameter in the SIFT algorithm for improving feature point stability, and different image contrasts significantly affect feature point quantity. For quantitative contrast analysis, we typically employ root-mean-square (RMS) calculation as shown in Equation 1, where n represents the number of pixels, x_i denotes the grayscale value of the i-th pixel, and $\bar{x}$ is the average grayscale value of the image pixels.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Contrast represents only one aspect of overall image information. Under the same contrast, complexity also substantially influences feature point quantity. Image complexity describes the inherent intricacy of an image and aggregates all image information. To obtain quantitative image complexity metrics, we typically utilize gray-level co-occurrence matrices to analyze important information including information entropy (H), correlation degree (COV), and energy (J), as detailed in Equation 2, where k represents the number of grayscale levels, N_i indicates the total count of the i-th grayscale level, N denotes the total number of image pixels, and $p(i, j)$ represents the element in the i-th row and j-th column of the gray-level co-occurrence matrix.

$$H = -\sum_{i=1}^{k}\sum_{j=1}^{l}\frac{N_{ij}}{N}\log\frac{N_{ij}}{N}$$
$$COV = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l}(ij)p(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y}$$
$$J = \sum_{i=1}^{k}\sum_{j=1}^{l}p(i,j)^2$$

Based on the above analysis, we comprehensively represent image complexity using four metrics: information entropy, correlation degree, energy, and edge ratio. After obtaining the computational model for image complexity, we need to construct a mathematical model relating feature point quantity, image complexity, image contrast (RMS), and contrast threshold to achieve adaptive control of feature point numbers. We first define the functional relationship shown in Equation 3, which can be interpreted as: for a given range of feature point quantities, the model automatically matches an appropriate contrast threshold to align the actual feature point count with requirements.

$$contrast\ threshold = f(complexity, RMS, point\ range)$$

We employ a BP neural network to fit this nonlinear mathematical relationship. From the Technical University of Munich dataset, we selected 3,000 images and constructed three data groups for each image as shown in Table 1, forming a training set of 9,000 total data groups. The contrast thresholds were obtained through repeated linear iteration.

According to Table 1, we use complexity (three metrics), contrast, and required feature point quantity as inputs to the BP neural network, with contrast threshold as the output. The BP neural network is chosen primarily because, compared to single-layer networks such as LMS (least mean square) and LM (Levenberg-Marquardt), it is a multi-layer network with superior nonlinear fitting capabilities. We constructed the BP neural network shown in Figure 5 using Matlab to fit the mathematical model of Equation 3. The hidden layer contains 9 nodes to enhance generalization capability, and the output layer is configured with 9 nodes using binary (0,1) representation for the contrast threshold. To ensure network accuracy, we set the training error precision to 0.251 and performed normalization on all inputs to eliminate dimensional effects. After 7,424 iterations as shown in Figure 6, the neural network reached the target precision and completed training. This neural network subsequently serves as the adaptive module in the new SIFT framework, automatically adjusting the contrast threshold based on image information and application requirements to achieve expected detection results.

[Figure 5: see original paper]

[Figure 6: see original paper]

3.2 FCN Neural Network Construction and Training

Traditional methods for detecting the pixel range of primary targets in images typically employ edge detection algorithms such as Canny or Sobel operators. However, since images contain numerous background patterns that generate substantial edge information, these methods cannot correctly compute the primary target' s pixel range. Therefore, we adopt an FCN neural network for pixel-level classification to effectively extract the primary target pixel range from the scene, improving feature point detection efficiency and matching precision. The specific network structure is shown in Figure 7.

[Figure 7: see original paper]

The network comprises five convolutional layers (Conv1 and Conv2 with two convolutions each, the others with three convolutions), five pooling layers, three fully connected layers, and three upsampling layers. The output results from the second and third pooling layers are applied to the upsampling process to compensate for lost pixel position information during successive downsampling and improve segmentation accuracy. We built the network using the Caffe deep learning framework and trained it on the PASCAL VOC2012 dataset. The trained FCN network is then applied to our algorithm framework for extracting primary target pixel ranges from images.

3.3 Feature Point Matching Optimization Based on Pose Offset Stability

Based on the previous analysis of camera pose offset stability in 3D reconstruction, assume there are n pairs of matching points. Building upon traditional KNN algorithm optimization, we retain matches that satisfy Equation 4, which constrains that the position of corresponding matching points in the second image should maintain a certain range constraint relative to the matching point position in the first image. Here, Th represents the constraint range, typically set to 10% of the image' s diagonal pixel length based on image pixel dimensions. This approach yields precise matching points and solves for the initial pose $[R|t]$ according to epipolar geometry principles.

$$\{(x_n, y_n), (x'_n, y'_n)\}_{n=1}^{N}$$
$$\sqrt{(x'_n - x_n)^2 + (y'_n - y_n)^2} \leq Th$$
$$Th = 0.1 \times \sqrt{Image_{col}^2 + Image_{row}^2}$$

3.4 Camera Pose Optimization Based on Graph Optimization

Based on the principles in Section 1.3, we design the graph structure shown in Figure 8 for the camera pose optimization problem.

[Figure 8: see original paper]

Using the previous camera pose as the reference frame, the current camera pose can be expressed as $[R|t]$, representing the vertex (optimization variable) in the graph structure. The edges (error terms) in the graph represent the pixel positions of feature points in the second image after $[R|t]$ rotation transformation from the first image. These pixel positions serve as error terms for calculation against the actual pixel positions we have obtained (i.e., the pixel positions of points in the second image from corresponding matched pairs). This constructs a least-squares problem with camera pose as the optimization variable, which can be implemented and solved using the graph optimization library g2o (General Graphic Optimization). The detailed analysis follows.

We first construct the observation equation shown in Equation 5, which represents observing feature points under the $[R|t]$ camera pose. Here, v_n represents noise conforming to a Gaussian distribution. Let the state variable be $x = \{[R|t], (x_n, y_n)\}$, transforming the original problem into constructing a maximum conditional probability problem as shown in Equation 6: given the matched point positions in the first and corresponding second images, what is the most probable camera pose?

$$\begin{aligned} &(x'_n, y'_n) = f(x_n, y_n, [R|t]) + v_n \\ &v_n \sim N(0, Q_n) \end{aligned}$$

$$P([R|t]|\{(x_n, y_n), (x'_n, y'_n)\}_{n=1}^{N}) \propto \prod_{n=1}^{N} P((x'_n, y'_n)|(x_n, y_n), [R|t])$$

The maximum likelihood estimation seeks to maximize Equation 6. Let the error term be:

$$e_n([R|t]) = (x'_n, y'_n) - f(x_n, y_n, [R|t])$$

This transforms Equation 6 into the form of Equation 9:

$$[R|t]^* = \arg\min_{[R|t]} \sum_{n=1}^{N} e_n^T Q_n^{-1} e_n$$

During the minimization of $E([R|t])$, if $e_n \geq Th$, the corresponding matching point pair is directly removed. This eliminates false matches on one hand

and reduces cumulative errors during the $E([R|t])$ solving process on the other, yielding more accurate results.

4.1 Feature Point Detection

Based on the analysis in Sections 3.1 and 3.2, experimental results are shown in Figure 9. In Figure 9(a), the SIFT algorithm detected 4,865 feature points, which were reduced to 1,084 after adjustment by our algorithm. For the case of insufficient feature points in Figure 9(b), our algorithm increased the count from 231 to 973. Figures 9(c)-(e) show the left images representing results without feature point detection range control, containing numerous background feature points outside the main target. The middle images show results after FCN neural network processing, extracting the primary target range to constrain feature point detection and yielding results concentrated on the main target, as shown in the right images.

The experimental results demonstrate that our algorithm can effectively adjust feature point quantity and control detection range. The computational time increases by only 1.7% compared to SIFT, primarily due to additional time consumed in the FCN detection stage. However, since the detection range is reduced, the overall time consumption remains comparable to SIFT.

4.2 Feature Point Matching

Based on the analysis in Section 3.3, experimental results are shown in Figure 10. Figure 10(a) displays feature point matching results without any optimization, containing numerous false matches. Figure 10(b) shows results after traditional KNN algorithm optimization, where matching improves but significant false matches persist due to similar pixel patches. After processing with our pose offset stability-based optimization algorithm, we obtain the results shown in Figure 10(c). Combined with feature point detection range constraints, we achieve Figure 10(d) where all matches are precise. Similar processing is illustrated in Figures 10(e) and 10(f).

Compared to traditional KNN-based matching optimization, our pose offset stability-based feature point optimization algorithm effectively eliminates false matches and improves result accuracy, providing accurate raw data for camera pose recovery. The computational time increases by 5.3% compared to KNN, mainly due to additional pixel position calculations for feature points. However, since these additions involve linear computation, the overall time consumption remains essentially comparable to KNN.

4.3 Camera Pose Optimization

By integrating the three algorithmic components described above, we performed point cloud recovery on the image pair shown in Figure 12. The results are presented in Figure 13.

[Figure 12: see original paper]

Based on the analysis in Section 3.4, we designed the following pseudocode to implement the graph optimization method, with experimental results shown in Figure 11.

```
Input: Initial pose [R|t]_init, feature point matching results M
Output: Optimized pose [R|t]_new

g2o {
    [R|t]_new = [R|t]_init
    for i in M:
        error = ||(x_i', y_i') - f(x_i, y_i, [R|t]_new)||^2
        if error   threshold:
            Delete (x_i, y_i)'
        if error(error) > threshold:
            [R|t]_new = [R|t]_new -  error/ [R|t]_new
        else:
            continue
}
```

The point cloud recovered from unoptimized camera poses is shown in Figure 11(a), while the result after optimization by our algorithm is shown in Figure 11(b), exhibiting more precise point cloud distribution and clearer detailed contours of the main target. For quantitative accuracy analysis, we employ reprojection error calculation: observing the spatial point cloud from the current camera pose viewpoint, computing its projection onto the pixel coordinate system, and comparing this position with the actual corresponding feature point pixel coordinates to statistically analyze errors in the x and y axes (in pixel units).

The experimental results demonstrate that our algorithm effectively improves camera pose accuracy. Compared to traditional optimization methods, the x-axis error and y-axis error are reduced by 65% and 91% respectively, while computational time improves by 20%. This improvement primarily stems from the elimination of mismatched pairs with errors greater than threshold Th according to Equation 9, which increases computational load. However, since our algorithm employs adaptive control modules and FCN neural networks to extract appropriate quantities of feature points from primary target regions in images, eliminating substantial background feature points, and utilizes our feature point matching optimization and pose refinement algorithms, we obtain high-precision initial spatial point cloud data, thereby enhancing the accuracy and efficiency of the entire 3D reconstruction process.

---

5 Conclusion

This paper addresses the current limitations of low adaptability in feature point detection and insufficient accuracy in feature point matching and pose recovery by proposing a novel algorithm framework based on SIFT. We implemented the algorithm and verified its effectiveness through experimental results, achieving more efficient feature point detection, matching, and pose recovery, while providing high-precision spatial point cloud data for the 3D reconstruction process and improving overall reconstruction efficiency. However, the threshold Th in our algorithm should possess better adaptability, requiring adjustment based on subsequently obtained camera poses to enable secondary refinement and acquire larger quantities of precise matching feature points for improved point cloud recovery.

References

[1] Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91-110.

[2] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features [J]. Computer Vision & Image Understanding, 2008, 110 (3): 404-417.

[3] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF [C]// Proc of IEEE International Conference on Computer Vision. 2012: 2564-2571.

[4] 徐建鹏, 卜凡亮. SIFT 动态对比度阈值算法 [J]. 中国人民公安大学学报: 自然科学版, 2017, 23 (4): 61-64. (Xu Jianpeng, Bu Fanliang. Mining. SIFT dynamic contrast threshold algorithm [J]. Journal of People' s Public Security University of China (Science and Technology), 2017, 23 (4): 61-64.)

[5] Keller J M, Gray M R, Givens J A. A fuzzy K-nearest neighbor algorithm [J]. IEEE Trans on Systems Man & Cybernetics, 2012, SMC-15 (4): 580-585.

[6] Kim J, Lee M. Robust Lane Detection based on convolutional neural network and random sample consensus [M]// Neural Information Processing. Springer International Publishing, 2014: 454-461.

[7] Costante G, Mancini M, Valigi P, et al. Exploring representa-tion learning with CNNs for frame-to-frame ego-motion estimation [J]. IEEE Robotics and Automation Letters, 2016, 1 (1): 18-25.

[8] Mccormac J, Handa A, Davison A, et al. SemanticFusion: dense 3D semantic mapping with convolutional neural networks [C]// Proc of IEEE International

Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2017: 4628-4635.

[9] 温佩芝, 苗渊渊, 周迎, 等. 基于卷积神经网络改进的图像自动分割方法 [J]. 计算机应用研究, 2018, 35 (9): 2848-2852. (Wen Peizhi, Miao Yuanyuan, Zhou Ying, et al. Improved image automatic segmentation method based on convolution neural network [J]. Application Research of Computers, 2018, 35 (9): 2848-2852.)

[10] 卫保国, 张海曦. 一种二次阈值调整 SIFT 算法 [J]. 计算机科学, 2014, 41 (6): 282-286. (Wei Baoguo, Zhang Haixi. Twofold adjusted threshold SIFT [J]. Computers Science, 2014, 41 (6): 282-286.)

[11] 王强, 李柏林, 罗建桥, 等. 一种局部二值模式图像特征点匹配算法 [J/OL]. 计算机应用研究, 2019, 36 (2). [2018-01-19]. http://www.arocmag.com/article/02-2019-02-046.html. (Wang Qiang, Li Bailin, Luo Jianqiao, et al. Local binary pattern image feature point matching algorithm [J/OL]. Application Research of Computers, 2019, 36 (2). [2018-01-19]. http://www.arocmag.com/article/02-2019-02-046.html.)

[12] 秦晓飞, 皮军强, 李峰. 基于极线约束的 ORB 特征匹配算法 [J]. 计算机应用研究, 2018, 35 (9): 2865-2868. (Qin Xiaofei, Pi Junqiang, Li Feng. ORB feature matching based on epipolar constraint [J]. Application Research of Computers, 2018, 35 (9): 2865-2868.)

[13] Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2014, 39 (4): 640-651.

[14] 高振宇, 杨晓梅, 龚剑明, 等. 图像复杂度描述方法研究 [J]. 中国图象图形学报, 2010, 15 (1): 129-135. (Gao Zhenyu, Yang Xiaomei, Gong Jianming. Research on image complexity description methods [J]. Journal of Image and Graphics, 2010, 15 (1): 129-135.)

[15] Kümmerle R, Grisetti G, Strasdat H, et al. g2o: a general framework for graph optimization [C]// Proc of IEEE International Conference on Robotics and Automation. 2011: 3607-3613.

Note: Figure translations are in progress. See original paper for figures.