Indirect Visual Odometry with Optical Flow

Chenguang Huang, Andong Tan
Department of Mathematics and Informatics
Technical University of Munich
Munich, Germany
chenguang.huang@tum.de, andong.tan@tum.de

***Abstract*—With the increasing need of robotics application in industry...**

***Index Terms*—Optical Flow, Visual Odometry, Computer Vision**

I. Introduction

Computer vision related tasks in robotics are attracting more and more attention. One typical task is the visual odometry. It is used widely in applications which require the depth information of objects without directly relying on distance measurement sensors like Lidar. Besides, many applications uses visual odometry to help with the motion planning of some specific agent. Therefore, this function could cause critical problems if it is not reliable. To ensure the safety of the agent, a common choice is to use mathematically provable methods to realize the function rather than using currently unexplainable techniques like deep learning. Thus it worths looking into the implementation details of visual odometry using traditional explainable methods.

To estimate the depth of some specific object, at least two images are needed if there is no prior assumption in how the world is constructed. However, before estimating the depth through triangulation, corresponding point in two images which describes the same 3D point should be found. The first way to achieve this is through key point detection in both images, and find the matching point pairs through similarity comparison. The second way to achieve this is through optical flow, which computes an estimated position of a pixel in the second image according its position in the first image. The above two ways are also suitable to find the point matching pairs between consecutive frames.

This paper mainly compares the above two methods.

The following sections are structured as below: Section II presents some basic concepts used in the two methods. Section III describes the pipeline of the two methods. Section IV shows the implementation details. Section V evaluates the difference between these two methods, and Section VI concludes the work.

II. Basic Concepts

To understand the methods better, some basic concepts are summarized in this section.

A. Visual Odometry

Odometry means the estimation of the change in position over time. And visual odometry refers to the estimation of the motion of a camera in real time through sequential images. In the context of vision based navigation, where cameras are often integrated in the robot, visual odometry can be used to estimate the ego motion of the mobile robot, and thus help to build a map in real time around the robot to support the navigation.

B. Optical Flow

The optical flow is apparent 2D motion which is observable between consecutive images. It can also be understood as pixel-wise motion estimation, because the optical flow calculates the motion of a specific pixel between two consecutive frames. Two main types of optical flow calculation are the Lukas & Kanade method (indirect method) and Horn & Schunck method (variational method). These two methods have different assumptions but both calculates the optical flow through an optimization process.

In Lukas & Kanade method, it assumes that (i) the motion is constant in a local neighborhood (ii) the brightness of a specific pixel is constant in different frames. Under the above assumptions, the energy function in the Lukas & Kanade method is formulated as follows:

$$E(v) = \int_{W(x)} |\Delta I(x', t)^T v + I_t(x', t)|^2 dx' \tag{1}$$

where $I(x', t)$ denotes the brightness of position x' at time t in the image, I_t denotes the derivative of brightness with respect to the time t, and $W(x)$ denotes the neighborhood of pixel x'. The optical flow v is calculated via the minimization of the above energy function. It generates sparse flow vectors.

In contrast, the Horn & Schunck method assumes (i) the brightness of a specific pixel is a constant in different frames (ii) the motions are spatially smooth. This method generates dense flow vectors. The energy function of this method is:

$$E(u, v) = \int\int [(I_x u + I_y v + I_t)^2 + \alpha^2(||\Delta u||^2 + ||\Delta v||^2)] dx dy \tag{2}$$

where I_x and T_y are the derivatives of brightness with respect to x and y axis, respectively. u and v are velocity in vertical and parallel directions. In the end, optical flow is obtained through solving for u and v variable.

C. Key Point Detection

The environment is continuous and complex. To successfully navigate in such en environment, some key information is more important than the other information. In the image, one kind of typical key information is the corner point.

One usual method to detect such corner points is through the analysis of the gradient of the image [1]. Typical methods include Foerstner and Harris detector [2]. More recently, detectors like BRIEF, SURF, FAST, and Shi-Tomasi are quite popular, and have many extended versions. In our experiment, key point detection is used to extract important information from the image frames to build the map.

D. Point Matching

To construct a 3D map, the matching relationship of points between frames are important, otherwise it will cause big error in the triangulation and possibly make the optimization process in determining the 3D position of a landmark unable to converge.

Finding matches between point in consecutive frames are typically done through the comparison of descriptors of two points. If the similarity between two descriptors is high enough, the two corresponding points are considered to be different projections of the same real world point.

One method to compute the descriptor is called the ORB descriptor [3]. ORB descriptor uses the BRIEF descriptor [4] and the orientation of the detected key points to represent the feature of a corner point. According to a set of principles, the correspondence of points between frames will be calculated.

III. Structure Design

Our pipeline assumes that the stereo cameras have already been calibrated, which means intrinsic matrix and relative transformation between two camera are already known. The structure of our pipeline is different in three cases. Given the stereo images of each time frame, we decide whether this frame is key frame or not. If this frame is key frame, the pipeline is also different according to whether this frame is the first frame. Therefore, the pipeline should adapt to three circumstances. So we will demonstrate our pipeline structure in these three cases. The criteria of deciding whether a certain frame is key frame will also be discussed later.

If this frame is key frame and the first frame, we firstly use Shi-Tomasi method to detect key points in the left image and use optical flow to acquire the corresponding key points in the right image. To increase the correctness of the result of optical flow, we perform optical flow backward and see whether the result key points are adequately close to input key points in the left image. Moreover, to filter out more outliers, we use the relative transformation acquired from calibration to calculate the essential matrix and perform epipolar constraint check. In the next step, we localize the camera by initializing the first pose of left camera with identity matrix. Finally, we use triangulation rule to calculate landmarks, the 3 dimensional position of key points in world coordinate frame.

If this frame is key frame and the second or later frame, we firstly use optical flow to acquire corresponding key points from last left image to current left image. Also a backward check is performed. Later we make a grid on the current left image and detect empty cells in the grid which do not contain key points. In the next step, we use Shi-Tomasi method to detect new key points inside these empty cells. In this period, we have two kinds of key points in the left image. The one is key points acquired with optical flow from last frame, which we will call old key points. The other is newly detected key points with Shi-Tomasi method in empty cells, which we will call new key points. Given these key points in the left image, we perform optical flow from left image to right image and backward check. And then we use epipolar constraint to find inliers. Later on we use RANSAC to localize camera with only inliers of old key points in the left image. Finally we use triangulation to calculate new landmarks with new key points and add observations of old key points to old landmarks.

If this frame is not key frame, we firstly perform frame to frame optical flow and make grid as in the second case above. But instead of detecting key points in empty cells, here we only count the empty cells number which we will use as one of key frame determination criteria. Later we localize camera with RANSAC.

Basically the pipeline of three circumstances is shown in the figure below.

IV. Implementation

A. Key framing strategies

The criteria of key framing is of significant importance. It will affect the accuracy and efficiency of our pipeline. On the one hand, if one frame can be easily regarded as a key frame, the interval between key frames will become very small. So the optimization will be limited in a small range, focusing too much on local features instead of a global view. In the meantime, more key frames means more optimization time, which makes the algorithm inefficient. On the other hand, if the key framing criterion is too strict so that only very few frames can be selected as key frames, the pipeline is more likely to collapse during the run. This is because we only detect new key points in key frames and if the interval between key frames is too big, the the base line is too large for optical flow to find correspondences that flow through all the frames. Once there are not enough correspondences, the camera localization will become very unstable and the algorithm will fail.

In our pipeline we use two criteria to decide whether next frame is key frame. The first criterion is inliers number. One necessary step in each time frame is camera localization. By using RANSAC method, a certain number of inliers are selected for determining the pose of camera. To guarantee

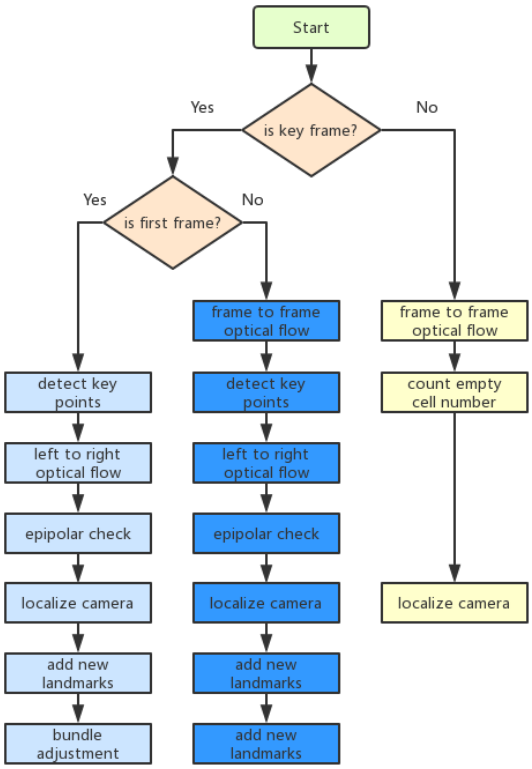

Fig. 1. Pipeline structure

there are enough key points for localization, we set a threshold for the number of inliers. If the number of inliers is lower than the threshold, we will treat next frame as key frame. The second criterion is empty cells number. As stated in above section, for every non-keyframe a grid is made on the left image and the number of empty cells will be recorded. A large empty cells number means the feature points calculated by optical flow gather in a small area of the image, indicating that the existing landmarks are leaving observation view. Once the empty cells number exceeds a threshold, we must set the next frame as key frame to detect new key points. In this way, the key points in the image will be adjust to distribute uniformly from time to time, making localization more stable.

B. Key points detection strategies

In every key frame, we have to detect key points in empty cells. We try two different implementations of key points detection in our pipeline. In the first method, we detect key points in the whole image and then only keep key points in empty cells. In the second method, we extract empty cells as a series of subimages and detect key points in each subimage separately. The maximum limit key points detection in the first method is set to 1500 per image. In the second method, it is set to 1 per subimage. The difference of key points detection is shown as below. We can easily see that the detected key points number in the second method is larger than in the first method. More importantly, the distribution of key points in the second method is more uniform. To quantify the influence of key point detection methods, we run the whole pipeline on euroc dataset and compare the root mean square error of camera trajectories. Finally, we find that the second method performs slightly better than the first method.

Fig. 2. detect key points in subimage

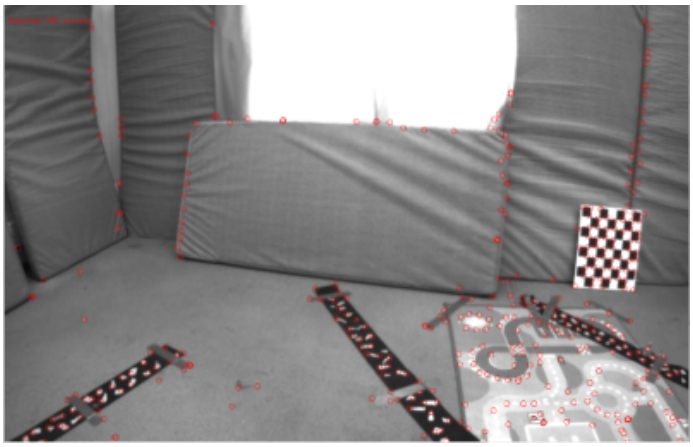

Fig. 3. detect key points in whole image

V. Evaluation

This section evaluates the performance difference between the matching method using optical flow and the matching method fully relying on key point detection, as well as the performance of the optical flow method under different set of hyper parameters. The evaluation focuses on three main aspects: precision, execution time, and visualization.

As described in the previous section, we use a grid to segment the image frame and decide whether the next frame is a key frame. A basic set of parameters for the grid of our experiments are as follows:

The grid size is designed to be so such that it's divisible by the image height and width. Besides, we find through experiments that square grids brings higher precision in the following

TABLE I
BASIC SET OF PARAMETERS

Image height (pixel) h	**Image width (pixel)** w	**Grid size (pixel $\times$ pixel)** s
480	752	32×32

TABLE II
RMSE PRECISION UNDER DIFFERENT NUMBER OF MAX. EMPTY CELLS

Max. number of empty cells (%)	0.35	0.4	0.42	0.44	0.46
RMSE	0.1303	0.1295	0.1099	0.1563	0.1304
Rmse using method without optical flow	0.1012				

experiments than rectangular grids. Similarly, the minimum number of key points 100 is found through experiments. To simplify the following discussion, we set the grid size and the minimum number of key points to be a constant, and discuss the influence of other hyper parameters in the rest of the section.

A. Precision

The table II shows how precision measured as Root Mean Square Error (RMSE) changes when the maximum number of empty cells percentage changes. This variable is designed to control the maximum area of the image where there is no key point. When the empty area is too big, it means we losses a lot of information from these areas and our construction of the map could be not precise enough. When this happen, the next frame will be set as a key frame, and key point detection will be executed.

From the table it is clear that when the maximum number of empty cells percentage is too small or too large, the precision of the built map will decrease. This means, if we do new key points detection too frequently, we would construct the map using too many key points, which means too many noises, as the key point detection itself is also not 100% precise. However, if we do key point detection too sparsely, we would have too less key points to construct the map, which indicates that even some small noises could cause a big error in map construction and thus cause an increased RMSE value. In the best case, the RMSE will reach a similar value as the method without using optical flow. Figure V-A shows more experiment data in a visualized way.

B. Execution Time

The evaluation on execution time includes three parts: detection time, optimization time, and key point matching time. Choosing these three aspects is because: all these three steps are common in two methods, and the difference between these two methods has a big impact on the execution time of these three steps.

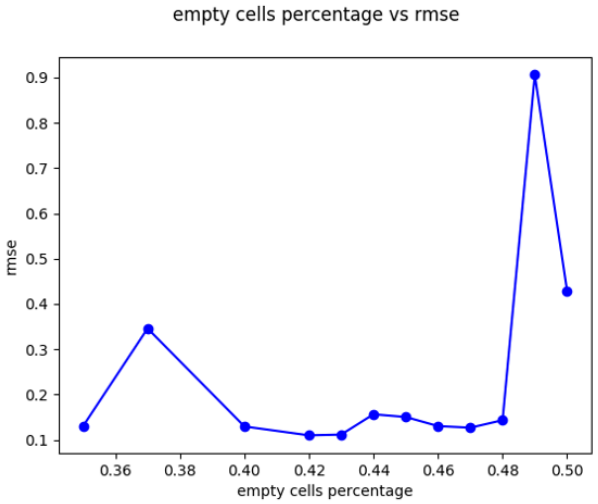

Fig. 4. Visualized version of RMSE change with different parameter of max. number of empty cells percentage

TABLE III
DETECTION TIME UNDER DIFFERENT NUMBER OF MAX. EMPTY CELLS PERCENTAGE

Max. number of empty cells (%)	0.35	0.4	0.42	0.44	0.46
Detection time (s)	3.03643	2.73232	2.43286	2.62328	2.26326
Detection time using method without optical flow	44.0214				

1) Detection Time: The table III shows how detection time changes with change of max number of empty cells percentage. Firstly, it is obvious that the detection time in the method using optical flow is much less than the method without optical flow, because the method using optical flow has replaced a lot of key point detection operation and find the matching points between frames using optical flow. Secondly, it's reasonable to see that with an increasing threshold for maximum empty cells percentage, the detection time generally decreases. This is due to the decreasing frequency of key frames, therefore the detection operation is executed less frequently. The experiment result shows that there is a slight increase on detection time when the threshold is set to 0.44 in comparison to 0.42, we think this could due to the fact that some frames have more textures and corners than the others, and a specific choice of 0.44 occasionally lets us have more key frames in these pictures with a lot of texture and corners. And thus increases the detection time slightly.

2) Optimization time: Optimization takes the most time of the whole pipeline. Here we choose the best parameter for maximum percentage of the number of empty cells, which is 0.42 and compare it with the method without optical flow. The results are shown in table IV.

The optimization time of the optical flow method is much higher than the method fully depending on key point detection. Although the number of landmarks are in similar level, the number of key frames and observation in optical flow method

TABLE IV
OPTIMIZATION TIME COMPARISON

	Max. number of empty cells(%) = 0.42	Method witho
Opt. time (s)	305.043	65.8619
Num. landmarks	290657	221886
Num. observation	1939460	559563
Num. key frames	414	170

TABLE V
KEY POINT MATCHING TIME COMPARISON

Point descriptor matching time (s)	5.79
Optical flow calculation (s)	117.4 143.1

is much higher than the other one, which causes the huge difference in optimization time. As the optimization process only happen in key frames, an increased number of key frames naturally increases the total optimization time. Here comes automatically the question: Why do we have more key frames and observations in optical flow based method?

Firstly, selection of key frames based on counting empty cells (optical flow method) rather than key point number (non optical flow method) in the image leads to a denser choice of key frames. Therefore the number of key frames is much higher.

Secondly, a possible reason for the large number of observations is that our pipeline forces a key point detection operation in every small grid of key frames. Although this has the advantage that the detected key points are more widely distributed in the whole image to help with a preciser optimization process, it has the disadvantage that many noises are also introduced and many noisy points are considered as an observation of some specific landmark.

The above reasons together leads to the difference in optimization time between two methods.

3) Key Point Matching Time: As a replacement of point descriptor comparison between frames, optical flow is used to track the key point between frames. Therefore, we compare the execution time of optical flow and key point matching through descriptors in the following table V:

The optical flow calculation time is much higher than the key point matching using descriptors. This is due to the fact that optical flow calculation includes an optimization process and the result is calculated iteratively, but descriptor comparison is a rather direct method which only needs several basic mathematical operations. Besides, the descriptor used (ORB) is itself also a very efficient and quick method for comparison.

C. Visualization

A visual comparison can give more intuition for the performance of different methods.

Comparing the Figure V-C and Figure V-C, we could find that the detected points using method of optical flow are more uniformly distributed, this is because in the implementation, we force each grid cell of the segmented image to find feature points, which helps to increase the precision of the estimated camera position.

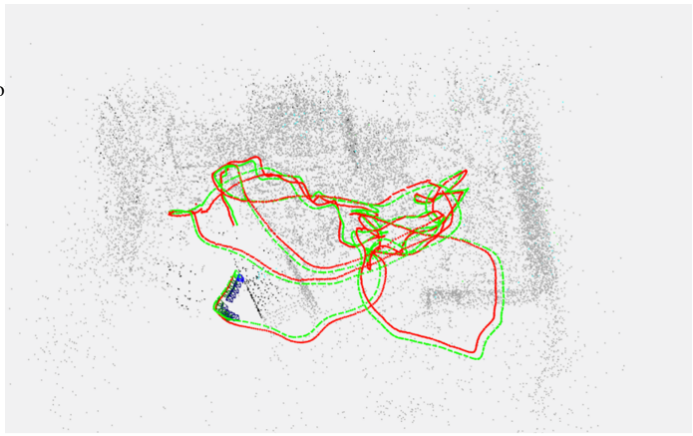

Fig. 5. Ground truth path (red) and calculated path (green) using optical flow based method. Gray points represent detected key points.

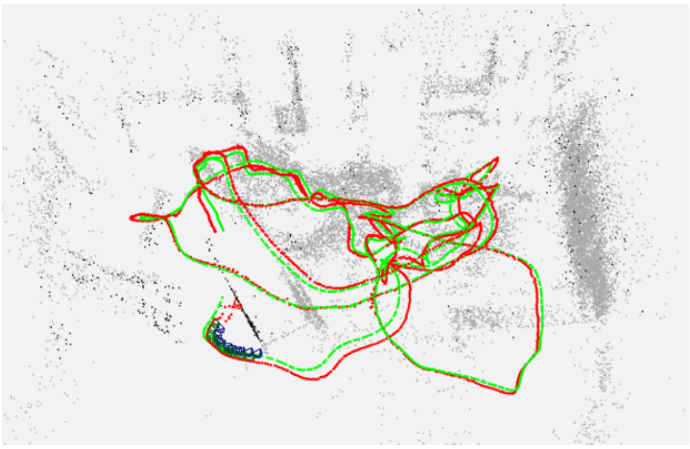

Fig. 6. Ground truth path (red) and calculated path (green) using method without optical flow. Gray points represent detected key points.

In Figure V-C and Figure V-C, there are two gray lines representing the calibration board in the vertical direction. We could find that the two gray lines in Figure V-C are closer than Figure V-C, which means a better reconstructed map in this area. A real world picture of this area is offered in Figure V-C.

VI. CONCLUSION

P1

REFERENCES

[1] H. Wang and M. Brady, Real-time corner detection algorithm for motion estimation, Image and vision computing, vol. 13, no. 9, pp. 695 703, 1995.

[2] Rodehorst, V.; Koschan, A. Comparison and evaluation of feature point detectors. In Proceedings of 5th International Symposium Turkish-German Joint Geodetic Days, Technical University of Berlin, Germany, March, 2006; ISBN 3-9809030-4-4.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, ORB: an efficientalternative to SIFT or SURF, inIEEE International Conference onComputer Vision (ICCV), Barcelona, Spain, November 2011, pp. 25642571

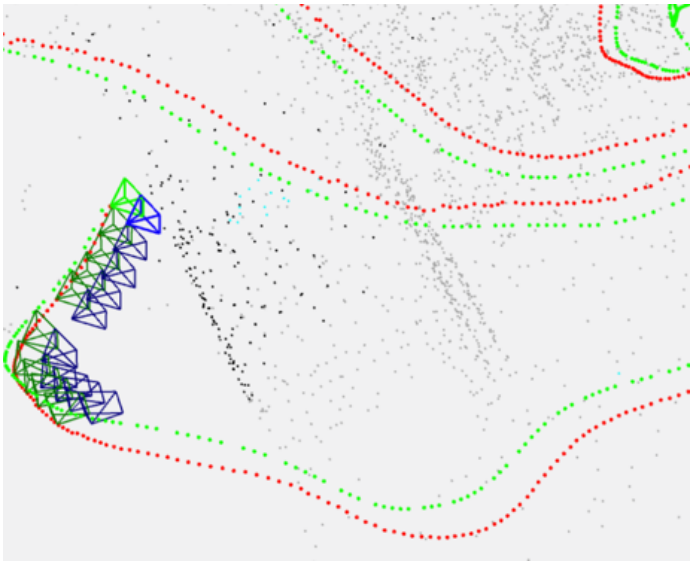

Fig. 7. Map constructed using optical flow based method. Gray points represent detected key points.

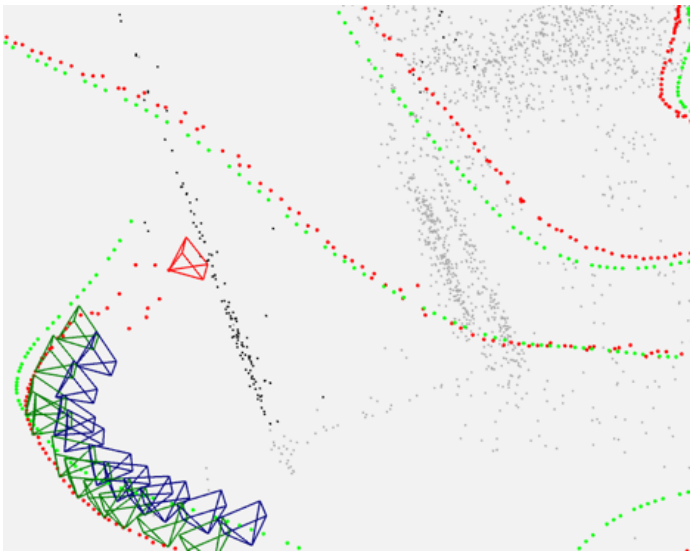

Fig. 8. Map constructed using method without optical flow. Gray points represent detected key points.

[4] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2564-2571.

Fig. 9. The desk image in real world.