STEREO TRACKING AND THREE-POINT/ONE-POINT ALGORITHMS – A ROBUST APPROACH IN VISUAL ODOMETRY

Kai Ni and Frank Dellaert

College of Computing, Georgia Institute of Technology
Atlanta, GA
nikai,dellaert@cc.gatech.edu

ABSTRACT

In this paper, we present an approach of calculating visual odometry for outdoor robots equipped with a stereo rig. Instead of the typical feature matching or tracking, we use an improved stereo-tracking method that simultaneously decides the feature displacement in both cameras. Based on the matched features, a three-point algorithm for the resulting quadrifocal setting is carried out in a RANSAC framework to recover the unknown odometry. In addition, the change in rotation can be derived from infinity homography, and the remaining translational unknowns can be obtained even faster consequently . Both approaches are quite robust and deal well with challenging conditions such as wheel slippage.

Index Terms— Stereo vision, tracking, motion estimation

1. INTRODUCTION

Visual odometry is a technique that estimates the ego-motion from images perceived by moving cameras. A typical use is autonomous navigation for outdoor robots, where getting accurate pose estimates is a crucial capability in many settings. The problem can be solved in various ways using different sensors, one of which is stereo. It does not emit any detectable energy into the environment in contrast to ladars. In addition, it does not require the presence of other signals, in contrast to GPS.

Visual odometry is at its heart a camera pose estimation [1], and has seen considerable renewed attention in recent years. Olson[2] uses visual odometry and incorporates an absolute orientation sensor to prevent drift over time. Nister et al. [3] developed a real-time system using a 3-point algorithm, which works in both monocular and stereo settings. Levin and Szeliski [4] used loopy belief propagation to calculate visual odometry on the base of the map correlation, which best fits offline processing. Some systems [5] also used omni-directional sensors from which panoramic images can be constructed.

This work was funded in part by DARPA under the IPTO/LAGR program (contract FA8650-04-C-7131)

In this paper, we present an approach of calculating visual odometry for outdoor robots equipped with a stereo rig. A stereo tracker is implemented to simultaneously decide the feature displacement in four successive images. Based on the matched features, a three-point algorithm for quadrifocal setting is carried out in a RANSAC framework to achieve the motion recovery. In addition, rotation and translation can be derived separately in a divide-and-conquer manner and then the simplified, translation only problem can be solved by one-point algorithm. We have implemented both algorithms on an outdoor robot that is used in challenging terrain and present extensive outdoor experiments.

2. STEREO FEATURE TRACKING

Most of the trackers currently available are not designed for stereo pairs. They either work for single images, or do not utilize the intrinsic stereo constraints, which may result in low efficiency. We designed a novel stereo tracking algorithm, which tracks features in four images simultaneously. After acquisition of the stereo images, new features in the left image are detected by the Harris detector [6]. We then track features from the left image to the right image of the stereo pair. As is well known from feature-based stereo, the only possible displacement is along the scan-line of the rectified image. So it can be found quite efficiently (less than 0.02 seconds/frame) and the two matched features in the same scan-line are grouped as a "stereo feature". The details of this stereo step are not presented, but it uses the same inverse warping scheme that will be described below, albeit simpler.

Similarly, the displacement of stereo features across frames also has the scan-line constraint. It can be represented by two horizontal offsets and a shared vertical offset $D = (d_l, d_r, d_y)$ with three degrees of freedom *instead of four when tracking them separately.*

As illustrated in Fig. 1, for each stereo feature, two 7×7 windows around the feature points in stereo images define the templates of the feature. Suppose we have templates $T_j (j = l, r)$ of a stereo feature at time t and a new stereo image pair at time $t + 1$ is $I_j (j = l, r)$. We want to know the displace-

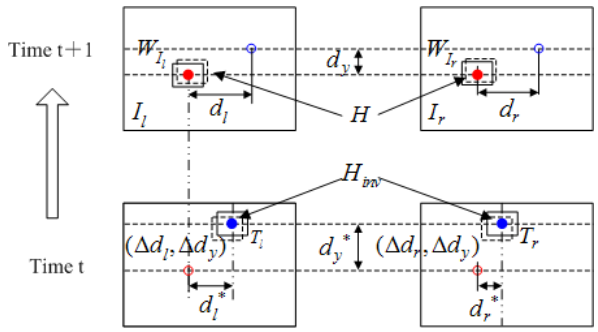

Fig. 1. Stereo tracking from time t to time $t+1$. The solid dots are the projections of the same feature in four cameras. With the inverse warping, H_{inv} is calculated in the template $T_j(j = l, r)$, instead of the image $I_j(j = l, r)$.

ment of the feature point, which is equal to minimizing the difference:

$$(d_l^*, d_r^*, d_r^*) = \min_{d_l, d_r, d_y} \sum_{j=l,r} \sum_{x} \left[W_{I_j}(x; (d_j, d_y)) - T_j(x) \right]^2$$

where $W_{I_j}(x; (d_j, d_y))$ is the warped pixel in I_j and corresponds to pixel x in T_j. The warping is defined by (d_j, d_y).

In each iteration, assume the current best estimate is (d_l, d_r, d_y) and we are trying to find the incremental offset $(\triangle d_l, \triangle d_r, \triangle d_y)$. As the warping (d_l, d_r, d_y) is affine invertible, we may directly add the incremental values[7] and update the estimate by

$$(d_l, d_r, d_y) \leftarrow (d_l, d_r, d_y) + (\triangle d_l, \triangle d_r, \triangle d_y)$$

at the end of each iteration.

The objective function with $(\triangle d_l, \triangle d_r, \triangle d_y)$ becomes

$$\sum_{j=l,r} \sum_{x} \left[W_{I_j}(x; (d_j + \triangle d_j, d_y + \triangle d_y)) - T_j(x) \right]^2 \quad (1)$$

By approximating $W(I_j, x; (d_j + \triangle d_j, d_y + \triangle d_y))$ with the first order Taylor expansion, we have

$$\sum_{j=l,r} \sum_{x} \left[W_{I_j}(x; (d_j, d_y)) + \begin{bmatrix} \frac{\partial W_{I_j}}{\partial d_j} \\ \frac{\partial W_{I_j}}{\partial d_y} \end{bmatrix}^T \begin{bmatrix} \triangle d_j \\ \triangle d_y \end{bmatrix} - T_j(x) \right]^2 \quad (2)$$

For simplicity of equations, we define the notation $I'_{m,n} = \frac{\partial W_{I_m}}{\partial d_n}$ and $T'_{m,n} = \frac{\partial W_{T_m}}{\partial d_n}$, and the objective function in Equation (2) can be rearranged to

$$\begin{aligned} & |e - H \triangle D|^2 \\ = & \sum_{x} \left[\begin{bmatrix} W_{I_l}(x; (d_l, d_y)) - T_l(x) \\ W_{I_r}(x; (d_r, d_y)) - T_r(x) \end{bmatrix} \right. \\ & \left. + \begin{bmatrix} I'_{l,l} & 0 & I'_{l,y} \\ 0 & I'_{r,r} & I'_{r,y} \end{bmatrix} \begin{bmatrix} \triangle d_l \\ \triangle d_r \\ \triangle d_y \end{bmatrix} \right]^2 \end{aligned} \quad (3)$$

where e is the error image between the stereo templates and the warped windows in $I_j(j = l, r)$ given the current best estimate, and the columns of H contain the "Jacobian images" with respect to each incremental displacement, and is evaluated at warped position (d_l, d_r, d_y) in image $I_j(j = l, r)$.

As (d_l, d_r, d_y) keeps changing in the iteration, the Hessian matrix H has to be re-evaluated at the beginning of each round. To avoid this expensive computation, we may alternatively use the inverse wrap scheme in [8] to make the warping $(\triangle d_l, \triangle d_r, \triangle d_y)$ at the template T_j instead of the image I_j, so that evaluation of H_{inv} in template T_j is not influenced by the change of (d_l, d_r, d_y) as shown in Fig.1:

$$\sum_{j=l,r} \sum_{x} \left[W_{I_j}(x; (d_j, d_y)) - W_{T_j}(x; (\triangle d_j, \triangle d_y)) \right]^2$$

The objective function in Equation (3) becomes

$$\begin{aligned} & |e - H_{inv} \triangle D|^2 \\ = & \sum_{x} \left(\begin{bmatrix} W_{I_l}(x; (d_l, d_y)) - T_l(x) \\ W_{I_r}(x; (d_r, d_y)) - T_r(x) \end{bmatrix} \right. \\ & \left. - \begin{bmatrix} T'_{l,l} & 0 & T'_{l,y} \\ 0 & T'_{r,r} & T'_{r,y} \end{bmatrix} \begin{bmatrix} \triangle d_l \\ \triangle d_r \\ \triangle d_y \end{bmatrix} \right)^2 \end{aligned}$$

It can be solved using the pseudo-inverse $(H_{inv}^T H_{inv})^{-1} H_{inv}^T e$. As a result of inverse warping, the estimate is updated by subtracting incremental offset:

$$(d_l, d_r, d_y) \leftarrow (d_l, d_r, d_y) - (\triangle d_l, \triangle d_r, \triangle d_y)$$

3. MOTION RECOVERY USING THREE-POINT ALGORITHM

First define the notation as follows: three 3D points X_i $(i = 1, 2, 3)$ are projected into the stereo cameras with the projection matrix

$$\Pi_j(R_j, t_j, x) = K_j R_j^T (x_i - t_j) \quad (4)$$

where K_j is the intrinsic calibration parameter for the left/right camera and (R_j, t_j) is the camera pose which also contains transformation from the world coordinate system to the camera coordinate system. There are four camera projections for left and right cameras at time t and $t+1$, which are notated as $\Pi_j(x)$ for $1 \le j \le 4$. The projections in these cameras are $p_{i,j}$.

The odometry (R, t) we are trying to calculate acts as the constraint between two successive camera poses (R_j, t_j). When three stereo features are tracked across two successive frames, there are twelve 2D projections in four images, with $12 \times 1.5 = 18$ degrees of freedom (every two features share a common y-coordinate value). As the stereo rig is constrained

by a known baseline, the unknown in this problem is a six-dimension robot odometry and three 3D points, which have $6 + 3 \times 3 = 15$ degrees of freedom. Obviously, there is unique solution if it exists.

The problem can be formulated as finding the maximum-likelihood estimate for the odometry (R, t) and the 3D points X_1, X_2, X_3. We are trying to minimize the following objective function:

$$\begin{aligned} &((R,t),(X_1,X_2,X_3))^* \\ &= \underset{(R,t),(X_1,X_2,X_3)}{\operatorname{argmin}} \left(\sum_{i,j} \|p_{i,j} - \Pi_j(R_j, t_j, X_i)\|^2 \right) \end{aligned} \quad (5)$$

where $\|y\|_\Sigma^2 = y^T \Sigma^{-1} y$ is the squared Mahalanobis norm.

To minimize the objective function, first we filter out the outliers in the putative correspondences, which is carried out in a RANSAC framework[9]. As the necessary feature number is three, we randomly pick three groups of features, and estimate the odometry based on these features. The estimation is done by using the Levenberg-Marquardt algorithm and solving the minimization problem in Equation (5). It can achieve fast convergence to the global minimum based on a good initial estimate.

4. INFINITY HOMOGRAPHY AND ONE-POINT ALGORITHM

Can we estimate the odometry even faster than the three-point algorithm in Section 3? Yes. It is straightforward to come to this idea: estimate the rotation and translation separately in a divide-and-conquer manner. The speed of solving a decoupled problem is always faster than the coupled one. One clue for getting the rotation is that the infinity homography is only related to the rotation matrix:

$$A_T^I(\psi, \theta, \phi) = KR(\psi, \theta, \phi)K^{-1}$$

where $A_T^I(\psi, \theta, \phi)$ is the infinity homography of the camera with the calibration matrix K (e.g. left camera in our system) from the image I (time $t + 1$) to the template T (time t), and R is the corresponding rotation matrix, which is decided by the yaw, pitch and roll angles ψ, θ, ϕ.

We are trying to optimize the best rotation

$$(\psi^*, \theta^*, \phi^*) = \min_{\psi,\theta,\phi} \sum_x \left[W_I\left(x; (\psi, \theta, \phi)\right) - T(x) \right]^2 \quad (6)$$

where $W_I\left(x; (\psi, \theta, \phi)\right)$ is the warped pixels at infinity in I and corresponds to pixel $x$in T. The warping is defined by the homography $A_T^I(\psi, \theta, \phi)$. Using the inverse scheme described in Section 2, we may get the object function:

$$\begin{aligned} &|e - H_{inv} \triangle D|^2 \\ &= \sum_x \left(\left[W_I\left(x; (\psi, \theta, \phi)\right) - W_T(x, (\triangle\psi, \triangle\theta, \triangle\phi)) \right] \right. \\ &\quad \left. \begin{bmatrix} \frac{\partial T}{\partial \psi} & \frac{\partial T}{\partial \theta} & \frac{\partial T}{\partial \phi} \end{bmatrix} \begin{bmatrix} \triangle\psi \\ \triangle\theta \\ \triangle\phi \end{bmatrix} \right)^2 \end{aligned}$$

Given that $K = \begin{bmatrix} f & & x_0 \\ & f & y_0 \\ & & 1 \end{bmatrix}$, the derivatives $(\frac{\partial T}{\partial \psi}, \frac{\partial T}{\partial \theta}, \frac{\partial T}{\partial \phi})$ can be formulated as:

$$\begin{aligned} \frac{\partial T}{\partial \psi} &= \begin{bmatrix} \frac{\partial T}{\partial x} & \frac{\partial T}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \psi} \\ \frac{\partial y}{\partial \psi} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial T}{\partial x} & \frac{\partial T}{\partial y} \end{bmatrix} \begin{bmatrix} -\frac{f^2 + (x - x_0)^2}{f} \\ -\frac{(x - x_0)(y - y_0)}{f} \end{bmatrix} \\ \frac{\partial T}{\partial \theta} &= \begin{bmatrix} \frac{\partial T}{\partial x} & \frac{\partial T}{\partial y} \end{bmatrix} \begin{bmatrix} -\frac{(x - x_0)(y - y_0)}{f} \\ -\frac{f^2 + (y - y_0)^2}{f} \end{bmatrix} \\ \frac{\partial T}{\partial \phi} &= \begin{bmatrix} \frac{\partial T}{\partial x} & \frac{\partial T}{\partial y} \end{bmatrix} \begin{bmatrix} -y + y_0 \\ x - x_0 \end{bmatrix} \end{aligned}$$

The incremental rotation can be calculated in the same way with Section 2 using pseudo-inverse $(H_{inv}^T H_{inv})^{-1} H_{inv}^T e$.

Now the problem of pose estimation is simplified to estimating translation only. Two points are not enough in the 6DOF (Degree of Freedom) case, becuase geometrically there is a rotation ambiguity around the line which connected these two 3D points. But the rotation is known in the 3DOF case, which means that with only one points perceived in two successive frames, the problem is solved: the unknowns are one 3D points and three-dimensional translation, which have totally $1 \times 3 + 3 = 6$ degrees of freedom. Meanwhile, the constraints are four projections of this 3D point with $1 \times 4 \times 1.5 = 6$ degrees of freedom, which is enough to discover the translation.

Similar with the approach in Section 3, but even better, the objective function in Equation (5) becomes:

$$(t, X_1)^* = \underset{t,X_1}{\operatorname{argmin}} \left(\sum_j \|p_{1,j} - \Pi_j(R_j, t_j, X_1)\|^2 \right)$$

where the odometry (R, t) in Equation (5) is replaced by the translation t.

As the procedures for RANSAC and least square optimization can be easily derived from the Section 3, we will not present them in detail here. The average time for three-point algorithm and one-point algorithm on 200 artificial data sets is 0.18s/frame and 0.11s/frame and one-point algorithm is 63.6% faster.

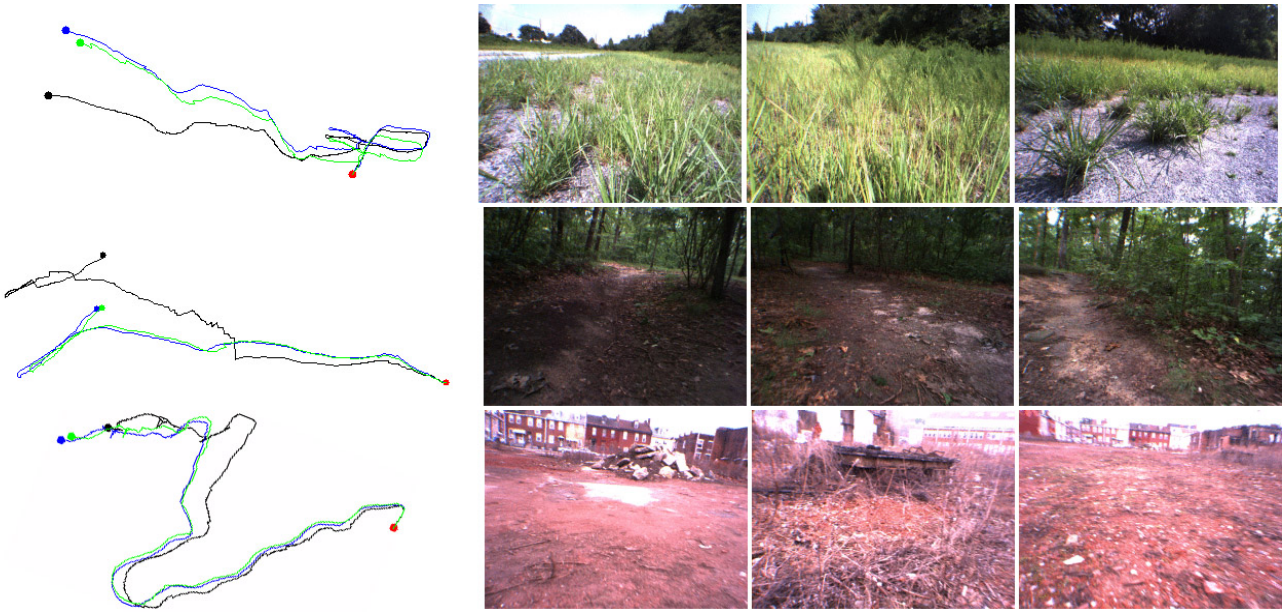

Fig. 2. Test results for visual odometry in tall grass, forest and test site. The black, green and blue lines are the trajectories of the ground truth from GPS, and visual odometry using three-point/one-point algorithms respectively. Every run starts from the red points, and ends at the point which has the same color with its trajectory.

	Ground Truth	V.O.(3PT/1PT)	Ave. Error
Tall Grass	90.9m	93.8m/91.7m	3.2%/0.9%
Forest	115.5m	118.5m/114.4m	2.5%/1.0%
Test Site	119.1m	115.4m/112.8m	3.1%/5.3%

Table 1. Metric accuracy of visual odometry using three-point/one-point algorithms for the tests in Fig. 2.

5. RESULTS

The visual odometry is well tested under different terrains in our outdoor robot, include tall grass, dirt bike site, forest, and test sites. Some results are shown in Fig. 2. The stereo computer has a 2.00GHz Intel Pentium M processor and 1G memory. The visual odometry runs at the speed of 7Hz with itself, and the actual value changes a little bit according to the quality of tracking result. With the other processes running at the same computer in real navigation, the visual odometry works at 4-5Hz.

The total distances got from the gps ground truth and visual odometry are compared in Table (1). It shows that visual odometry is effective in ego-motion estimation for the robot.

6. REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[2] C. F. Olson, "Stereo ego-motion improvements for robust rover navigation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2001, pp. 1099–1104.

[3] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2004, pp. 652–659.

[4] A. Levin and R. Szeliski, "Visual odometry and map correlation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[5] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, August 1988.

[7] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Intl. J. of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[8] F. Dellaert and R. Collins, "Fast image-based tracking by selective pixel integration," in *1999 ICCV Workshop on Frame Rate processing, Corfu, Greece*, 1999.

[9] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," *Commun. Assoc. Comp. Mach.*, vol. 24, pp. 381–395, 1981.