Monocular Visual Odometry Based on Hybrid Parameterization

Sherif A.S. Mohamed[a], Mohammad-Hashem Haghbayan[a], Jukka Heikkonen[a], Hannu Tenhunen[b], and Juha Plosila[a]

[a]University of Turku (UTU), 20500 Turku, Turku, Finland
[b] Royal Institute of Technology (KTH), Stockholm, Sweden

ABSTRACT

Visual odometry (VO) is one of the most challenging techniques in computer vision for autonomous vehicle/vessels. In VO, the camera pose that also represents the robot pose in *ego-motion* is estimated analyzing the features and pixels extracted from the camera images. Different VO techniques mainly provide different trade-offs among the resources that are being considered for odometry, such as camera resolution, computation/communication capacity, power/energy consumption, and accuracy. In this paper, a hybrid technique is proposed for camera pose estimation by combining odometry based on triangulation using the long-term period of direct-based odometry and the short-term period of inverse depth mapping. Experimental results based on the EuRoC data set shows that the proposed technique significantly outperforms the traditional direct-based pose estimation method for Micro Aerial Vehicle (MAV), keeping its potential negative effect on performance negligible.

Keywords: Visual odometry, Monocular camera, Inverse-depth map

1. INTRODUCTION

Obtaining the position and orientation of a platform by analyzing images captured by single or multiple cameras, i.e., visual odometry (VO), is becoming one of the most interesting and challenging topics in the field of machine vision. VO techniques can be based on one camera or more than one camera. Monocular VO overcomes those issues by analyzing the extracted features from images captured by a single camera *monocular-camera*.[1] Usually in VO based on monocular camera, the pose of the camera is calculated by extracting some correspondent points in consecutive camera frames, i.e., *keyframe*, while the camera is moving. The most common way to represent points observed in images is using Euclidean XYZ coordinates and the depth of these points is computed using the triangulation technique from two[2] or more[1] keyframes. Another important issue is the *parallax* angle that is the angle between inclination of two *lines of sight*, i.e., *ray*, between the camera and a point in two consecutive sample times. The measurement of *parallax* angle helps to estimate the distance of the point from the camera.

Using monocular camera has some well-known challenging drawbacks that increase the error of estimating the pose based on one camera. For instance, newly observed 3D points initially have unknown depth until they are observed in two or more keyframes. This procedure introduces a delay, which means the newly detected points with highly uncertain depths cannot be immediately used to improve the camera motion estimation. Another problem of using the euclidean representation is that in the case of 3D points that retain a low parallax angle, i.e., the ones that are very far away from the camera, not many frames are included in the estimation of the camera pose. In indoor applications, mostly few centimeters of movement is enough to measure sufficient parallax. However, in very large-scale outdoor scenes, for distant points, it might take a few kilometers of camera movement before their parallax is properly observed. Unlike the Euclidean representation, inverse depth can represent points with a low parallax angle and at infinity, i.e., $\rho = 0$. The only drawback of the inverse-depth parameterization is that it needs six parameters to represent a point rather than the three points of the Euclidean approach, which increases the computational load.

In this paper, we propose a hybrid parameterization technique by using both euclidean and inverse depth approaches to to use both euclidean and inverse depth mapped point for the sake of odometry. We exploit computational efficiency of the euclidean coordinates to connect newly observed points to the old observed points of the same object. This finding the correspondence can be used for odometry. On the other hand, points with low parallax angles are added to odometry via the inverse depth mapping. To increase the computational efficiency of the proposed system, we switch all points with the linearity index below a threshold from inverse depth to Euclidean coordinates.

We organize the remaining part of this paper as follows. In Section 2, we illustrate the overall system work flow which consists of three steps: initialization, image alignment, and depth mapping. In Section 3, we present the experimental results. Finally, in Section 4, we draw the conclusions.

2. SYSTEM OVERVIEW

The overall system architecture of the proposed approach is shown in Figure 1. The system contains two main tasks concurrently running with each other, 1) the task for estimating the motion of the camera using a direct method, i.e., *image alignment* and 2) the task for building a local 3D map of the surrounding. For estimating the camera motion the incremental sequence of camera movements are estimated by minimizing the intensity of pixel windows that observe the same 3D point. The 3D points stored in the local map are back-projected into the image and a patch of 4x4 pixel window is selected around those points. The second task builds a local 3D map of the surrounding using two parameterization techniques. a) In case the current frame is selected as a keyframe, the current and previous keyframes are triangulated and a set of new Euclidean points, which are coded with three Cartesian coordinates (XYZ), are stored in the local depth map. b) Contrarily, if the captured frame is not a keyframe, a feature-based algorithm, e.g, FAST, is applied to detect new features. These features have a parallax angle ($\alpha = 0$) and initial depth equal to infinity. The inverse-depth technique is used to represent these points, which are coded by 6-vector containing the Euclidean position of camera from which the feature was observed, azimuth and elevation angles, and the inverse of the Euclidean distance from the camera to the 3D point.

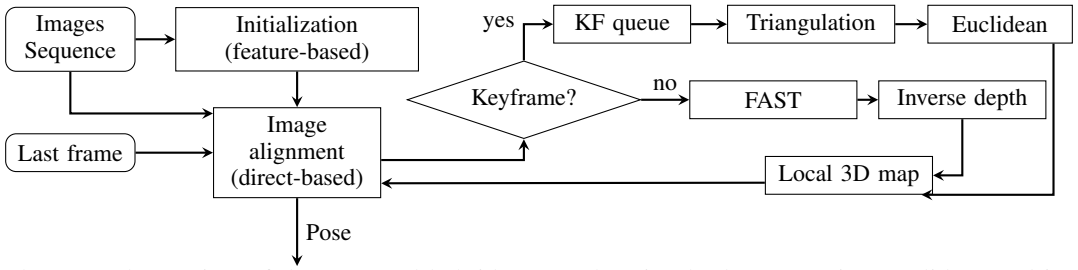


Figure 1: The general overview of the proposed hybrid monocular visual odometry using Euclidean and inverse-depth parameterization.

2.1 Initialization

The initialization phase in the proposed technique is based on the Nister five-point algorithm.[3] The first two captured keyframes are used to compute the essential matrix. Subsequently, the rotation and translation of the camera are determined from the QR-factorization of the essential matrix.[3] First, the features from the captured images are extracted. To extract and track key points, i.e., *corners and edgelets*, from the captured images, FAST[4] and Canny[5] algorithms are used, respectively. It is worth mentionning that the advantage of FAST, compared with other feature extraction methods like SIFT,[6] is its computational efficiency and high speed. To reduce the computational complexity, only a small part of the strong edges, i.e., edgelets, are processed. Moreover, the limited number of keyframes are selected in the case there are too many extracted features. To track the features from the first frame, the Kanade-Lucas-Tomasi (KLT) feature tracker is used. To label a frame as the next keyframe, the number of trackable features among the current frame and next frame must be more than 50, and the average disparity between features in the current frame and next frame must be more than 50 pixels. By this, we make sure that most of the detected features have a large parallax angle. After this, the five-point algorithm is used to obtain the essential matrix. Then, the initial transformation matrix is decomposed from the essential matrix by using QR-factorization.

2.2 Image Alignment

In image alignment, the camera pose is estimated by minimizing the photometric error (*intensity error*) of already extracted patches, in 4x4 pixels, of the observed similar 3D points in two frames as is shown in Figure 2.

First, the 3D points stored in the local map are projected into the reference and current frame using camera projection matrix Mp. A Euclidean point $\mathbf{P}^E \in \mathbb{R}^3$ in the world coordinate space can be projected to the image coordinates $\mathbf{w} \in \mathbb{R}^2$ using the camera projection model as follows:

$$w_i = KR^T(P_i^E - t) \tag{1}$$

Whereas inverse depth points $\mathbf{P}^{ID}$ are coded by 6 parameters, which can be projected to the 2D image using the following equation:

$$w_i = KR^T(m_i(\theta_i, \psi_i) - \rho_i(t - p_c)) \tag{2}$$

where K denotes the intrinsic camera matrix, R and t are the rotational matrix and translation vector, respectively. For the inverse depth representation, p_c denotes the camera position that is $(x_i y_i z_i)$ in which the point is observed first. The

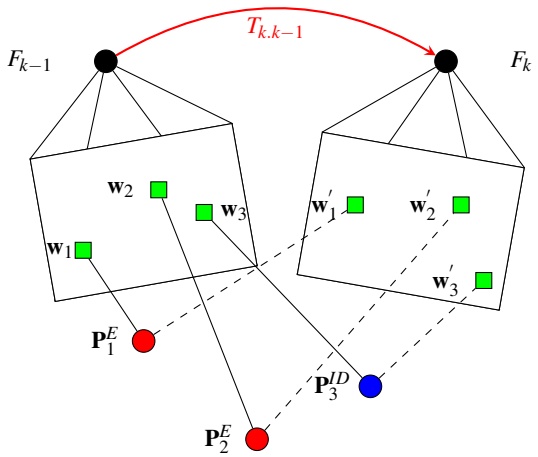


Figure 2: An example of Euclidean points that are shown in red and inverse depth points that are shown in blue for two consecutive frames. A direct method is applied to obtain the transformation matrix, i.e., $T_{k.k-1}$, by minimizing the photometric error of the patches that are shown in green squares in the figure. As can be seen, the method uses both Euclidean and inverse-depth points

unit directional vector is denoted by m, where θ_i ψ_i are the azimuth and the elevation in the world frame, respectively. The depth (d_i) of the 3D point is encoded by its inverse $\rho_i = 1/d_i$.

After all 3D points have been projected into two consecutive frames, a patch of 4x4 pixels is selected around each point. The current frame (F_k) is aligned with the previous frame (F_{k-1}) by minimizing the intensity residual δ. The intensity residual is defined as the photometric difference between the pixel patches that observe the same point of $\mathbf{P}$. The incremental motion of the camera is estimated by optimizing the intensity residual as follow:

$$T_{k,k-1} = \arg\min_{T_{k,k-1}} \sum_{i=1}^{n} \frac{1}{2} \left\| \delta(I_k - I_{k-1}, \mathbf{P}_i) \right\|^2 \tag{3}$$

where $T_{k.k-1}$ denotes the transformation matrix between the two consecutive frames F_k and F_{k-1}, that minimizes the photometric error between pixels observed in the same 3D point $\mathbf{P}_i$.

The residual is computed using the inverse compositional algorithm,[7] in which all the computationally demanding parts (e.g. the Hessian matrix) are computed only once in the pre-computational part to avoid over computation. As shown in Algorithm 1, the alignment process is done in each pyramid level starting from the smaller image size. In our case, the algorithm starts from the pyramid level 3 and ends with the pyramid level 0. The optimization stops at each pyramid level when the error increases or the algorithm detects convergence, i.e., the obtained results are close tothe previous one. The bilateral filter, which is non-linear and non-iterative,[8] is used in the algorithm to smooth images.

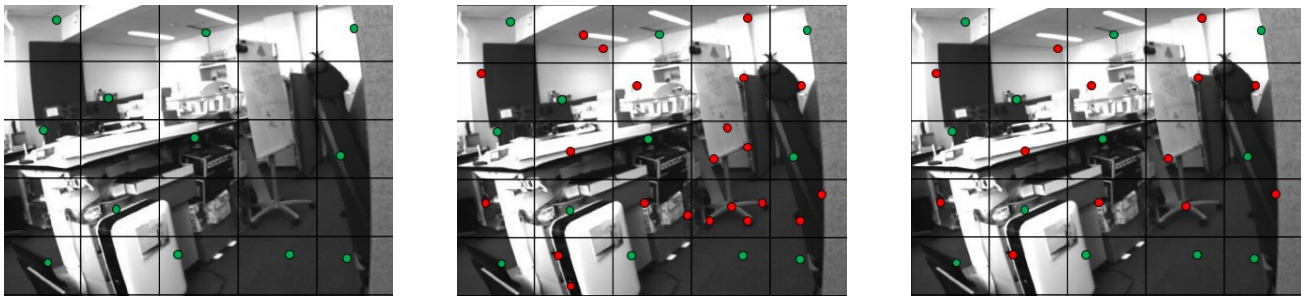
2.3 Depth Mapping

The mapping part builds a map of the 3D points observed by the camera. These points are used later to reduce the searching region in the localization part. The uncertainty of the depth is reduced by updating the map with each entering new frame. When the points are converged they are stored in the Local 3D Map data structure, see Figure 1. To make the system more feasible for the resource-constrained embedded systems, only Euclidean points observed in the set of keyframes and the highest score Inverse-depth points in the normal frames are stored in the map. Moreover, when the keyframe queue is full, points extracted from the oldest keyframe are deleted from the map to maintain a low computational time. A feature extraction procedure is performed on each new keyframe, which increases the robustness of the system by inserting new 3D points in the map.

First, the local depth map must be initialized. From the first two keyframes selected at the beginning of the pipeline, a large number of extracted and matched features are used to initialize the local depth map. The two keyframes must share the same field of view and the displacement between these two keyframes must be sufficiently large. Based on this local depth map initialization, the initial translation and rotation of the camera are computed by decomposing the Essential matrix and the 3D point depth of the matched points and applying triangulation based on this.

Input: two consecutive frames F_{k-1}, F_k
Result: Transformation matrix $T_{k,k-1}$
initialization and compute T_{init} ;
for *Image pyramid* $\leftarrow$ 3 *to* 0 **do**
  pre-compute $Jacb_{ref}$;
  for *iter* $\leftarrow$ 0 *to N* **do**
    for *Fts* $\leftarrow$ 1 *to M* **do**
      initialization and compute bilateral filter;
      for *pixel* $\leftarrow$ 0 *to* 15 **do**
        compute residual;
      end
      compute chi-squared, compute Jacobian, compute Hessian;
    end
    if *converged* **then**
      break ;
    end
    update $T_{k,k-1}$;
  end
end

Algorithm 1: Image alignment using direct-based method



(a) 3D point projection (b) Feature extraction in empty cells (c) Shi-Tomasi selection

Figure 3: Depth map initialization and update. (a) stored points in the map are projected on the current keyframe (green circles). (b) using the FAST detector, new features are extracted from empty cells (red circles). (c) based on the Shi-Tomasi score, only one point is kept in each cell

After this, normal frames are split into small sub-images of a fixed size, in our experiment 30x30 pixels. The 3D points already stored in the local depth map are projected to the frame and those sub-images that are non-empty, i.e., some projected points are inside their area, are discarded (see Figure 3). Then, features are extracted in empty sub-images using the FAST feature detector. If more than one features are detected in a cell, the point with the highest Shi-Tomasi score[9] remains and the other points are deleted, see Figure 3c. The formula to calculate the score is as follows:

$$R = min(\lambda_1, \lambda_2) \tag{4}$$

where R denotes the score, λ_1 and λ_2 are the eigenvalues. Newly extracted points are represented by inverse depth parametrization. A 3D point observed in a scene can be defined by a 6-D state vector:

$$p_i = (x_i,\ y_i,\ z_i,\ m(\theta_i\ \psi_i),\ \rho_i\ )^T \tag{5}$$

The p vector denotes the ray from the camera position $(x_i y_i z_i)$ in which the point is firstly observed. The uni-directional vector denoted by m, where θ_i ψ_i are the azimuth and the elevation in the world frame. The depth (d_i) of the 3D point is encoded by its inverse $\rho_i = 1/d_i$. Unlike Euclidean points, inverse depth can represent points in infinity when $\rho = 0$. The only drawback of inverse depth parametrization is its need for six parameters to represent a point instead of the three points in Euclidean parametrization, which increases the computational load. Therefore, in our algorithm, nearby and initialized points, with low uncertain depths, are converted to Euclidean points to reduce the computational load. An inverse depth point can be converted to Euclidean coordinates according to the following equation:

Table 1: Execution times of the different parts of the proposed algorithm

Thread	Time(ms)
initialization	58.901 ms
image alignment	8.79 ms
mapping	0.79 ms

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}(\theta_i, \psi_i) \tag{6}$$

Inverse-depth coordinates are used immediately at the time the point is observed, which is beneficial in sudden camera movements as there is no delay between the feature detection and odometry. Moreover, points with large uncertain depths can be used to improve gradually the orientation estimation. The depth of points is refined with every new frame until the depth uncertainty is small enough. The linearity index L_ρ[10] is used to determine at which time a point is converged and is calculated as follows:

$$L_\rho = \frac{4\sigma_\rho}{\rho_0}|1 - \frac{d_0}{d_1}\cos\alpha| \tag{7}$$

where σ_ρ^2 is the inverse-depth variance and $\rho_{0=1/d_0}$ is the inverse-depth and d_0, d_1 are the distances between the point and the current and next camera position, respectively. The parallax angle which is the angle between the two cameras' optic axes is denoted by α. When a feature is newly observed in the first few frames, the parallax angle is likely to be very small and thus the linearity index L_ρ is low (good) unlike the Euclidean linearity index which is high (bad) for a small parallax angle. Proceeding with the estimation and increasing the parallax angle lead to a lower depth uncertainty, and therefore it is more computationally efficient to switch from inverse depth to Euclidean coordinates. A point is switched when the linearity index L_d is smaller than a threshold, i.e., 0.1 in our experiments.

3. EXPERIMENTAL RESULTS

The proposed algorithm is tested on a Jetson TX2 board with a quad-core ARM Cortex-A57 CPU @ 2GHz clock frequency. We use the EuRoC dataset to evaluate the efficiency of the technique.[11] It is a visual-inertial dataset collected on-board for a Micro Aerial Vehicle (MAV). Figure 4 shows the estimated camera pose using the proposed technique in comparison against the real camera pose, the *groundtruth* in the figure, and the common position estimation technique, *EP* in the figure. The results are reported separately for both X and Y dimensions of the camera pose of a MAV. In the case where the number of features in a frame is too high, only the first 100 features are selected, see Section 2.1. For the camera calibration procedure, a target-based calibration method is used.[12] As can be seen, the EP+IDP method significantly outperforms the pose estimation nmthod based on anly EP. This demonstrates the effect of using the contribution of inverse depth technique in increasing the accuracy of pose estimation.

Table 1 shows the execution time of different parts of the algorithm. As can be seen, the mapping part, that represents the extra processing the algorithm imposes on the traditional techniques, has only a small penalty on the overall execution time, less than 2%.

4. CONCLUSION

Monocular visual odometry is a challenging technique for estimating the position and orientation of the camera by moving a vehicle/vessel and by analyzing the features and pixels extracted from the camera images. This paper proposed a hybrid technique to carry out odometry by combining pose estimation based on long-term triangulation of direct-based odometry and short-term inverse depth mapping. Experimental results based on the EuRoC data set show that the proposed technique significantly outperforms the traditional pose estimation method, keeping its negative effect on performance negligible.

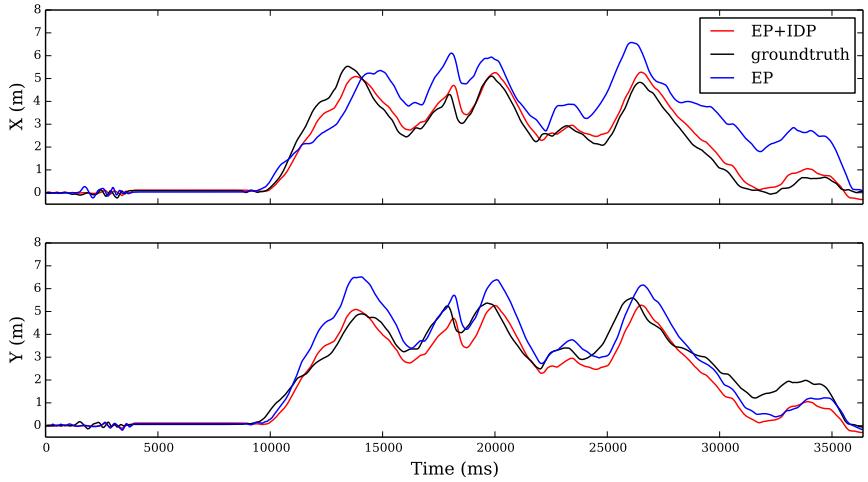


Figure 4: The obtained results of the proposed hybrid technique. Normal estimated position, *EP* in the figure, and the combination of invergse-depth mapping and EP, i.e., *EP+IDP*, are compared against the real position of the camera, i.e., the *groundtruth*

REFERENCES

[1] Mur-Artal, R., Montiel, J. M. M., and Tardós, J. D., "ORB-SLAM: a versatile and accurate monocular SLAM system," *CoRR* (2015).

[2] Klein, G. and Murray, D., "Parallel tracking and mapping on a camera phone," in [*Proc. Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*], (October 2009).

[3] Nister, D., "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(6), 756–770 (2004).

[4] Rosten, E. and Drummond, T., "Machine learning for high-speed corner detection," in [*Proceedings of the 9th European Conference on Computer Vision - Volume Part I*], *ECCV'06*, 430–443, Springer-Verlag, Berlin, Heidelberg (2006).

[5] Canny, J., "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**, 679–698 (Nov 1986).

[6] Lowe, D. G., "Object recognition from local scale-invariant features," in [*Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*], *ICCV '99*, 1150–, IEEE Computer Society, Washington, DC, USA (1999).

[7] Baker, S. and Matthews, I., "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vision* **56**, 221–255 (Feb. 2004).

[8] Hung, K. . and Siu, W. ., "Fast image interpolation using the bilateral filter," *IET Image Processing* **6**, 877–890 (October 2012).

[9] and, "Good features to track," in [*1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*], 593–600 (June 1994).

[10] Civera, J., Davison, A. J., and Montiel, J. M. M., "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics* **24**, 932–945 (Oct 2008).

[11] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., and Siegwart, R., "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research* (2016).

[12] Oth, L., Furgale, P., Kneip, L., and Siegwart, R., "Rolling shutter camera calibration," in [*2013 IEEE Conference on Computer Vision and Pattern Recognition*], 1360–1367 (June 2013).