

Article

Uncertainty Estimation of Dense Optical Flow for Robust Visual Navigation [†]

Yonhon Ng ¹, Hongdong Li ¹ and Jonghyuk Kim ^{2,*} 

¹ College of Engineering & Computer Science, The Australian National University, Canberra, ACT 2601, Australia; yonhon.ng@anu.edu.au (Y.N.); hongdong.li@anu.edu.au (H.L.)

² School of Mechanical and Mechatronics Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia

* Correspondence: jonghyuk.kim@uts.edu.au; Tel.: +61-2-9514-4492

[†] This paper is an extension version of the conference paper: Ng, Y.; Kim, J.; Li, H. Robust Dense Optical Flow with Uncertainty for Monocular Pose-Graph SLAM. In Proceedings of the Australasian Conference on Robotics and Automation, Sydney, NSW, Australia, 13 December 2017.

Abstract: This paper presents a novel dense optical-flow algorithm to solve the monocular simultaneous localisation and mapping (SLAM) problem for ground or aerial robots. Dense optical flow can effectively provide the ego-motion of the vehicle while enabling collision avoidance with the potential obstacles. Existing research has not fully utilised the uncertainty of the optical flow—at most, an isotropic Gaussian density model has been used. We estimate the full uncertainty of the optical flow and propose a new eight-point algorithm based on the statistical Mahalanobis distance. Combined with the pose-graph optimisation, the proposed method demonstrates enhanced robustness and accuracy for the public autonomous car dataset (KITTI) and aerial monocular dataset.

Keywords: monocular visual navigation; dense optical flow; uncertainty estimation; epipolar constraints



Citation: Ng, Y.; Li, H.; Kim, J. Uncertainty Estimation of Dense Optical Flow for Robust Visual Navigation. *Sensors* **2021**, *21*, 7603. <https://doi.org/10.3390/s21227603>

Academic Editor: Dong-Hwan Hwang

Received: 20 September 2021
Accepted: 9 November 2021
Published: 16 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Uncrewed aerial vehicles (UAVs) have drawn significant attention from the research community and industry in the last few decades. The primary advantage of a UAV is the ability to access places that are hazardous and hard to reach, such as for inspection of infrastructure [1,2], precision agriculture [3,4], and search and rescue operation [5,6]. Due to their light weight, rich information, and low cost, cameras have been extensively applied for robot navigation and environment mapping. In particular, visual simultaneous localisation and mapping (SLAM) algorithms have drawn significant interest in both robotics and computer vision communities [7,8]. These algorithms typically consist of a front-end and a back-end part. The front-end part typically performs visual odometry between consecutive images or relative to a known map. The back-end module performs graph optimisation and handles the loop closures.

Most monocular visual odometry methods use sparse feature points matched between images to compute the inter-frame motion [9–11]. The feature matching accuracy is improved by incorporating the kinematic vehicle models [11,12]. Ref. [10] adopts the learning-based method, and [11] utilises the convolutional neural network to train the ground plane detection to estimate the height for a ground vehicle. These approaches, however, make the methods not suitable for aerial or rough terrain applications.

Recently, dense optical flow for monocular visual odometry has received significant attention, and the current state-of-the-art methods are capable of producing, on average, more than 85% of dense correspondences, having less than 3 pixels error [13–15]. Although the computed correspondences from optical flow are not very accurate compared to sparse feature matches, the dense nature of the correspondences helps mitigate the inaccuracy of individual matches. Another benefit of using dense optical flow is that it can avoid the

shortcomings of typical sparse matches. For example, the sparse matches may be clustered around a small area of the image or encounter problems with the planar degeneracy [16], resulting in a biased motion estimate. However, these methods have not explicitly considered the uncertainty in the feature matching and monocular SLAM pipelines.

In this paper, we propose a novel *robust* monocular simultaneous localisation and mapping in a principled probabilistic framework. This is accomplished by using dense optical flow with estimated uncertainty as to the input to our visual odometry pipeline. Utilizing the existing robust back-end pose-graph SLAM [17], the methods will achieve significant robustness to the sensing uncertainty and loop-closure outliers. The contributions of this paper are threefold:

- Estimation of uncertainty from the dense optical flow. The epipolar constraint is first included in the matching cost to improve the matching accuracy. The uncertainty is then recovered by fitting a bivariate Gaussian to the matching cost.
- Development of a new Mahalanobis eight-point algorithm to compute the visual odometry. The estimated uncertainty enables efficient sampling of RANSAC (random sampling and consensus) and accurate pose estimation using the weights based on the Mahalanobis distance.
- Demonstration of the proposed methods for ground and aerial navigation.

To the best of our knowledge, this is the first work that achieves the robustness of visual odometry by utilizing the full uncertainty of the dense optical flow with its demonstration for aerial navigation in an unstructured forest environment. Figure 1 illustrates the new eight-point algorithm with the full uncertainty. Ideally, all matched inliers in the second image should be on the line, called an epipolar line, as shown in the figure. If there is any deviation, the shortest distance to the line, such as Euclidian distance, $\min(d_E)$ can be used as a weighting factor (as a penalty cost). However, the Mahalanobis distance, $\min(d_M)$ can be the shortest in a statistical sense, as illustrated in the figure, thus providing adequate weighting (details discussed in Section 4).

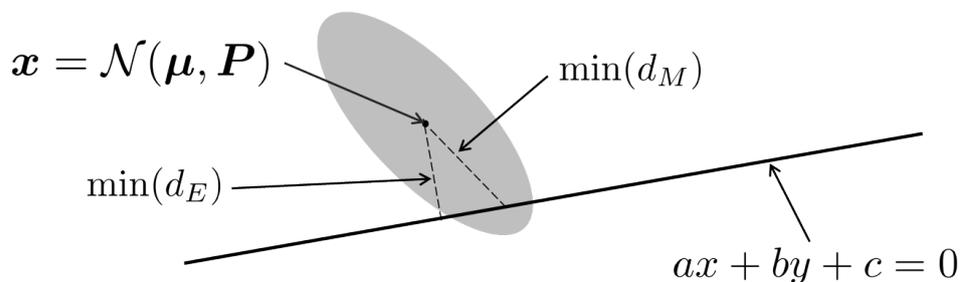


Figure 1. An illustrative figure showing an image feature pixel x represented as a two-dimensional random variable with mean μ and covariance matrix P ; the epipolar line is represented as a straight line with equation $ax + by + c = 0$, $\min(d_M)$ is the minimum Mahalanobis distance, while $\min(d_E)$ is the minimum Euclidean distance. In standard 8-point algorithms, the (inverse) Euclidean distance is used as the weight, while our method utilises the Mahalanobis distance which is statistically more consistent (shorter distance to the line in this example).

The paper is organised as follows. Section 2 discusses related works on optical flow and visual odometry. Section 3 presents our bivariate uncertainty estimation method from the measured optical flow. Section 4 details our new eight-point algorithm by introducing the Mahalanobis distance from the epipolar line and using it as the weighting factor in the eight-point algorithm. Section 5 describes the visual processing pipeline of the visual SLAM, discussing the scale estimation, inter-frame pose estimation, depth estimation, and loop closure computation. Section 6 shows the experimental results that verify the performance of our proposed method, and Section 7 presents the conclusions and future extensions of our work.

2. Related Work

Due to the lack of a fixed baseline, visual odometry using a monocular (single camera) system has an inherent difficulty in estimating the metric distance, unlike a stereo camera (e.g., [18,19]) setup. For the same reason, monocular (single camera) visual odometry also has difficulty with scale drift. Another challenge of visual odometry is in the image feature-matching step, especially for scenes with repetitive texture and dynamic objects. Work by [20,21] also assume the scene contains known objects with known dimensions to assist with the visual odometry task. However, this assumption is not valid when exploring an unknown, unstructured environment.

State-of-the-art monocular visual odometry methods suitable for a forward-looking camera in an unstructured environment are primarily designed for ground vehicles [10,11,22]. These methods prevent scale drift and recover the metric scale by assuming that the camera travels at a fixed, known height above a roughly planar ground. Although this is a valid assumption for a ground-based vehicle, it is not true for an aerial vehicle.

Dense optical flow methods have been investigated to address the difficulty of feature matching in low or repetitive textured scenes with dynamic objects. Some notable work includes the use of full discrete cost volume to compute the dense optical flow [13,23] directly. DCFlow [13] is one such work in which an efficient semi-global matching is proposed. DCFlow also utilises EpicFlow [24] for interpolation and local homography fitting to obtain accurate optical flow estimation with lower computational cost. Recently, convolutional neural network, pyramidal feature extraction, and feature warping have also been applied successfully to compute the optical flow [14,15]. However, these methods, like most state-of-the-art optical flow methods, do not calculate and utilise the uncertainty of the optical flow. In comparison, our proposed method directly makes use of the total discrete cost volume of DCFlow [13], which allows simple incorporation of extra matching cost (epipolar) constraint to improve the optical flow accuracy and direct estimation of the 2D uncertainty.

The dense optical flow correspondences are treated like sparse feature matches, which are used to estimate the so-called fundamental matrix. The well-known normalised eight-point algorithm [25] can be used to compute the fundamental matrix. The conventional eight-point algorithm does not minimise a meaningful geometrical error. Thus, other methods that reduce the Sampson distance [26], symmetric epipolar distance [27], and re-projection error [16] were developed. A review paper on other methods to compute the fundamental matrix is given in [28]. However, these methods usually assume isotropic, homogeneous error in the image features location, which is not suitable for optical flow correspondences. Due to the definition of optical flow, the feature location error is zero in the first image. In general, it has an anisotropic and non-homogeneous error in the feature located in the second image. Based on this knowledge, we derive a new method to compute the fundamental matrix that minimises the Mahalanobis distance of the image feature in the second image frame. Using a calibrated camera, the inter-frame camera pose can then be recovered from the fundamental matrix.

3. Dense Flow with Uncertainty

A new optical flow method is developed based on DCFlow [13]. The accuracy of the optical flow estimate is improved by incorporating the epipolar constraint into the cost computation, and the optical flow uncertainty is extracted by fitting a Gaussian function to the cost volume.

3.1. Dense Flow with Epipolar Constraint

Dense optical flow is a method to estimate the motion of each pixel between two input images. The algorithm typically optimises a cost function consisting of a data cost that penalises visually dissimilar pixels and a regularisation cost that encourages spatially smooth variation of the optical flow field. For each pixel in the first image of $M \times N$, the

matching cost of a set of candidate pixels of $D \times D$ in the second image is computed, such that the full cost volume is four-dimensional (4D).

The epipolar constraint is added to the cost function before regularisation is applied. Figure 2 (bottom) shows an example of the epipolar constraint added to one of the cost volumes slice, which encourages the correspondences to be close to the epipolar line by increasing the cost of finding a match far from the line. This process is accomplished as follows. First, Shi-Tomasi corner features tracked by the Kanade–Lucas–Tomasi (KLT) algorithm [29] are used as sparse correspondences for the well-known eight-point algorithm [25] to obtain an initial estimate of the fundamental matrix. A truncated L_2 cost is added to the cost volume to enforce the epipolar constraint based on the computed fundamental matrix. When the pixel in the first image corresponds to a fixed point of the scene, the cost of finding the match far away from the epipolar line is increased proportionately to the squared distance. Conversely, a truncated cost is applied when a pixel in the first image corresponds to a point on a moving object. This truncation helps avoid matches that satisfy the epipolar constraint but are visually dissimilar and wrongly selected.

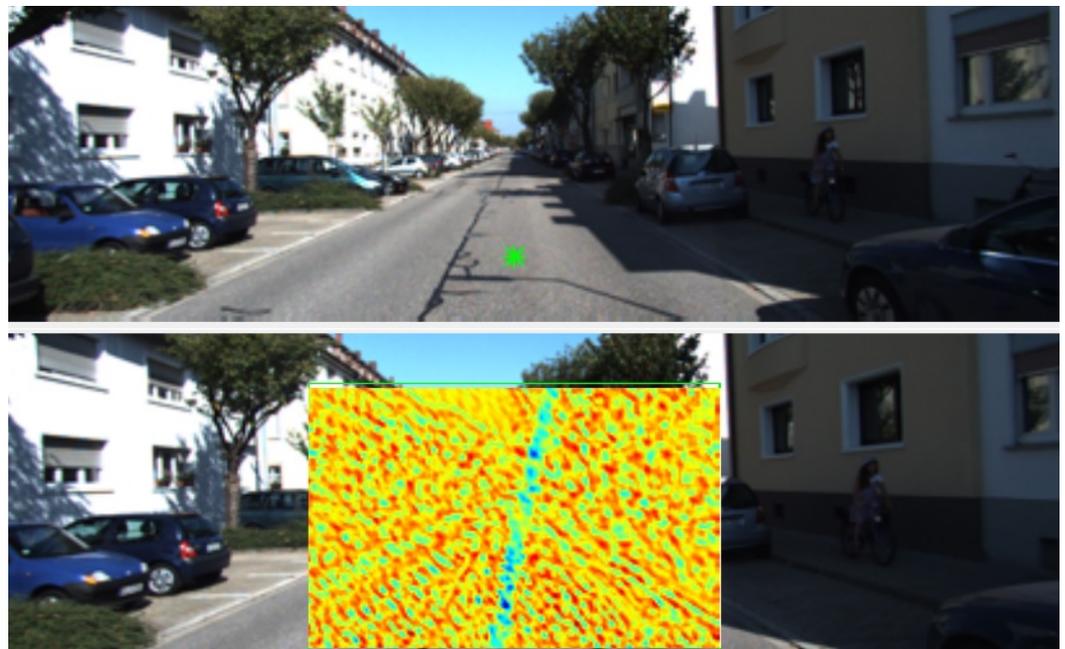


Figure 2. An example illustrating the epipolar constraint added to a cost slice from two consecutive images. **(top)** The first image shows a pixel marked by a green star. **(bottom)** The second image shows a bounding box enclosing the candidate-matching pixels for the corresponding pixel in the first image, with the matching cost of corresponding candidate pixels. Red colour represents low-matching candidates and blue for high matching. The epipolar line (a straight line towards an epipole) shows highly likely matching.

3.2. Uncertainty Estimation

DCFlow [13] and most other state-of-the-art optical flow methods implicitly assume each correspondence has a homogeneous, isotropic Gaussian uncertainty. However, the uncertainty of each correspondence can have different magnitude and correlation depending on the visual similarity of neighboring pixels. Figure 3 illustrates an example of a matching cost slice, in which the negative logarithm of a unimodal Gaussian distribution is fitted to the optical flow-matching cost output.

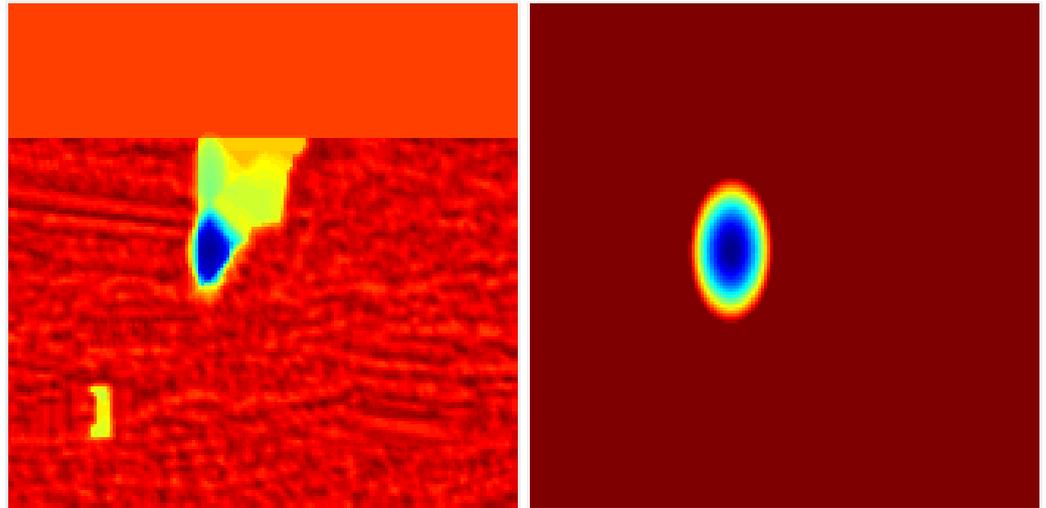


Figure 3. Uncertainty fitting of negative logarithm of a bivariate Gaussian to a matching cost slice (after spatial smoothness regularisation step). From left to right: (left) 2D cost slice, (right) the approximate 2D cost slice using a 2D Gaussian fitting.

For a general two-dimensional Gaussian distribution, we know that the negative logarithm of the likelihood function is half of the squared Mahalanobis distance. The squared Mahalanobis distance, d_M^2 can be computed as [30]:

$$d_M^2(x|\mu, \mathbf{Y}) = (\mathbf{x} - \mu)^T \mathbf{Y} (\mathbf{x} - \mu). \quad (1)$$

We represent the vector $(\mathbf{x} - \mu) = [x, y]^T$, and

$$d_M^2 = [x \ y] \begin{bmatrix} Y_{xx} & Y_{xy} \\ Y_{xy} & Y_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

$$= Y_{xx}x^2 + 2Y_{xy}xy + Y_{yy}y^2, \quad (3)$$

where x is the vector representing the coordinates of a point, μ is the vector representing the coordinates of the mean (optical flow estimate) of the Gaussian distribution, and \mathbf{Y} is the information matrix of the Gaussian distribution which is the inverse of the covariance $\mathbf{Y} = \mathbf{P}^{-1}$.

The elements of information matrix \mathbf{Y} can then be computed using the least squares as

$$\underbrace{\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ x_2^2 & 2x_2y_2 & y_2^2 \\ \vdots & \vdots & \vdots \\ x_N^2 & 2x_Ny_N & y_N^2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} Y_{xx} \\ Y_{xy} \\ Y_{yy} \end{bmatrix}}_y = \underbrace{\begin{bmatrix} d_1^2 \\ d_2^2 \\ \vdots \\ d_N^2 \end{bmatrix}}_d \quad (4)$$

$$\therefore \mathbf{y} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{d},$$

where $\{d_1, d_2, \dots, d_N\}$ are the matching costs at their respective x and y coordinates from the optical flow cost volume. DCFlow computes the matching cost efficiently by using $(1 - f_1 \cdot f_2)$, where f_1 and f_2 are the unit vectors representing the image feature descriptor. This method results in a cost value between 0 (visually similar) and 1 (visually dissimilar). However, the negative logarithm of a Gaussian likelihood function has a value between 0 to infinity. Thus, we can exclude pixels with high cost from our Gaussian fitting by only using pixels with a matching cost below a set threshold.

Similar to most top-performance optical flow methods, DCFlow has a post-processing step to remove unreliable matches from the semi-dense correspondences before EpicFlow [24]

interpolation. This step is accomplished by computing the forward and backward optical flow and eliminating those matches that do not satisfy the forward–backward consistency. This post-processing step changes the uncertainty estimate such that the correspondences that got removed should be assigned a high uncertainty. We replace those values with the maximum uncertainty of the optical flow estimate.

These provide us with three channels (Y_{xx} , Y_{xy} , Y_{yy}), encoding the information matrix for every pixel correspondence for the scaled-down pair of RGB input images. We can scale the uncertainty image back to the original resolution by applying an image resize operation. First, the information matrix parameters are converted to covariance parameters, scaled up to the original image resolution, followed by a multiplication of 9 (squared of image rescaling factor). The scaled-up covariance parameters are then converted back to the information matrix following matrix inverse.

The estimated uncertainty can also be used to determine if the two input images are visually similar, as illustrated in Figure 4, which will be helpful in computing the loop closure constraints (discussed in Section 5). Most local neighbours will have similar optical flow magnitude and direction if the two input images belong to the same scene. The regularisation step will then shrink the region of possible matching locations, and thus, the uncertainty decreases. On the other hand, local neighbours generally have different optical flow magnitude and directions if the two input images belong to different scenes. In that case, the regularisation step will not shrink the region of possible matching locations, and the uncertainty will become high.

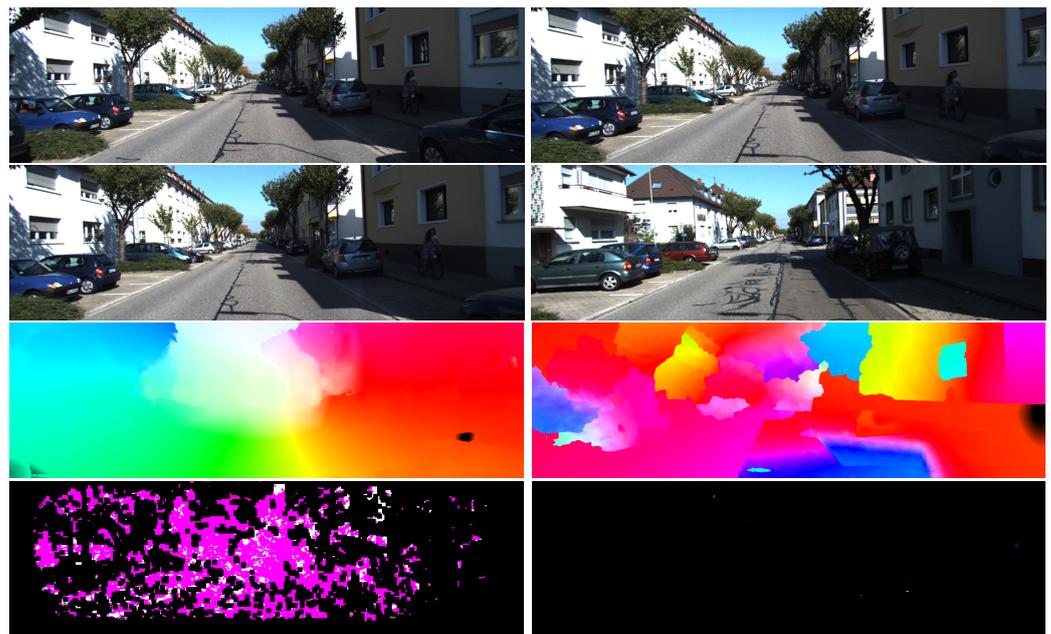


Figure 4. An example of estimated optical flow and uncertainty magnitude. From top to bottom: first input image, second input image, optical flow, estimated information matrix. Left column corresponds to sequential images, while right column corresponds to two input images with high structural similarity (SSIM) index, but is not of the same scene. Black colour for information matrix values corresponds to high covariance (unreliable) pixels.

4. New Mahalanobis 8-Point Algorithm

The computed dense optical flow, dense depth estimate (prior), and their uncertainty are used to estimate the camera motion, called visual odometry. The dense correspondences are used similar to conventional sparse feature matches. In contrast, the optical flow uncertainty is used for the random sampling and consensus and the Mahalanobis eight-point algorithm.

From a pair of input images, we can find a set of matching pixels $x_i \leftrightarrow x_i'$. Then, the two pixels and the 3D feature location stay in a plane called the epipolar plane. Equivalently, a fundamental matrix F exists,

$$x_i'^T F x_i = 0, \quad (5)$$

where x_i and x_i' are represented in the homogeneous coordinates.

Each matching pixel provides a linear constraint on the elements of F . Since the scale of F can be arbitrary, the solution of F can be computed using eight sets of matching pixels. The fundamental matrix can be represented by a vector (f) of length 9 to solve the equations. Given n pairs of matching image features, the linear constraints can be concatenated into a matrix form as

$$\underbrace{\begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots \\ x_n'x_n & x_n'y_n & x_n' & y_n'x_n & y_n'y_n & y_n' & x_n & y_n & 1 \end{bmatrix}}_A f = \mathbf{0}. \quad (6)$$

The solution is then computed by finding the null space of matrix A . When more than eight noisy matching pixels are provided as input, RANSAC is applied to identify reliable matches (inliers) to compute F . Given the inlier set, the solution of f is then refined by calculating the corresponding right singular vector of A with the smallest singular value. This method is the well-known eight-point algorithm, where sparse feature matches are typically used.

However, solving the null space of Equation (6) only minimises the algebraic error $\|x'^T F x\|$, which does not guarantee the minimisation of a meaningful geometrical distance nor reflect any weighting of the inliers. One well-known method minimises the Sampson distance [26,27], which modifies the rows of matrix A by a multiplicative scaling, such that

$$\begin{bmatrix} \phi_1 x_1' x_1 & \phi_1 x_1' y_1 & \phi_1 x_1' & \cdots & \phi_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_n x_n' x_n & \phi_n x_n' y_n & \phi_n x_n' & \cdots & \phi_n \end{bmatrix} f = \mathbf{0}, \quad (7)$$

with

$$\phi_i = \frac{1}{\sqrt{(F x_i)_1^2 + (F x_i)_2^2 + (F^T x_i')_1^2 + (F^T x_i')_2^2}}, \quad (8)$$

where F is the iteratively refined fundamental matrix that is first initialised by computing the null space of A from (6). The rank 2 constraint is also enforced on the solution to obtain the final estimate of F .

In this work, we propose to use the uncertainty of the optical flow to accurately weigh each equation (row of matrix A) during the refinement step of the Mahalanobis eight-point algorithm. It ensures that the fundamental matrix solution minimises the squared Mahalanobis distance to all the inlier correspondences with respect to the uncertainty. This process was illustrated in Figure 1 in the introduction. The weight $\{\phi\}$ can be computed from the minimum Mahalanobis distance in the figure.

Theorem 1. For the estimated information matrix, the new multiplicative scaling to each row in (7) becomes

$$\phi_i = \sqrt{\frac{Y_{xx}Y_{yy} - Y_{xy}^2}{(F x_i)_1^2 Y_{yy} + (F x_i)_2^2 Y_{xx} - 2(F x_i)_1 (F x_i)_2 Y_{xy}}}, \quad (9)$$

where the notation $(v)_k$ is the k^{th} element of the vector v .

Proof. Given an initial fundamental matrix estimate F , and homogeneous coordinates of matching pixels in both images x_i and x_i' , we assume the error is only present in the

second image's features x'_i . From Figure 1, let mean $\mu = [x_0, y_0]^T$, information matrix $Y = P^{-1} = \begin{bmatrix} Y_{xx} & Y_{xy} \\ Y_{xy} & Y_{yy} \end{bmatrix}$, and a point on the line be $[x_1, y_1]^T = [x_1, \frac{-ax_1-c}{b}]^T$.

First, we calculate the minimum Mahalanobis distance between the line l and the mean image feature location μ . The minimum Mahalanobis distance equals the square root of the minimum squared Mahalanobis distance. The squared Mahalanobis distance d_M^2 between the feature pixel and the epipolar line is computed as follows.

$$d_M^2 = \begin{bmatrix} x_1 - x_0 \\ \frac{-ax_1-c}{b} - y_0 \end{bmatrix}^T \begin{bmatrix} Y_{xx} & Y_{xy} \\ Y_{xy} & Y_{yy} \end{bmatrix} \begin{bmatrix} x_1 - x_0 \\ \frac{-ax_1-c}{b} - y_0 \end{bmatrix} \quad (10)$$

Expanding (10) and computing the first derivative of d_M^2 with respect to x_1 equals to zero provides us the solution of x_1 , where d_M^2 is the minimum. We then substitute this solution of x_1 back into (10) and apply a square root to obtain the equation of the minimum Mahalanobis distance, $\min(d_M)$, as follows.

$$\min(d_M) = |ax_0 + by_0 + c| \sqrt{\frac{Y_{xx}Y_{yy} - Y_{xy}^2}{a^2Y_{yy} + b^2Y_{xx} - 2abY_{xy}}} \quad (11)$$

Since the original eight-point algorithm minimises $|ax_0 + by_0 + c|$, the multiplicative scaling is thus

$$\phi = \sqrt{\frac{Y_{xx}Y_{yy} - Y_{xy}^2}{a^2Y_{yy} + b^2Y_{xx} - 2abY_{xy}}} \quad (12)$$

This completes the proof. \square

Similar to the Sampson distance method, F is then iteratively refined and is first initialised by computing the null space of A from Equation (6). In the modified RANSAC step, the uncertainty of the optical flow is used to guide the sampling of the matches by increasing the likelihood of selecting correspondences (or matches) with lower uncertainty. The multinomial resampling method is used as in the particle filtering [31], which helps in decreasing the required number of iterations to ensure good inlier set selection.

From the estimated fundamental matrix F and the intrinsic camera matrix K , the essential matrix E can be recovered as $E = K'^T F K$, and the camera pose (t, R) can be obtained from the well-known factorisation method [16].

5. Visual Processing Pipeline

Unlike stereo odometry, the estimated translation from the monocular camera faces problems with the scale ambiguity and the scale drift over multiple images.

Two methods are proposed to minimise the scale drift: fitting the ground plane and using the computed depth map. First, the scale is determined by fitting a plane through the 3D reconstructed points roughly parallel to the camera axis's zx (forward-right) plane. Assuming the ground is visible roughly in the middle of the image, we use the reconstructed points below the camera (y coordinate of the 3D points is positive) and not too far to the side (image coordinate x within half the image width from the centre) of the camera. Plane fitting provides us with a plane equation satisfying $ax + by + cz + d = 0$. The plane's height with respect to 1 unit of inter-frame translation then equals $-d/b$. If the camera's height, h , is known (calibrated from training data, or estimated throughout the motion), the scale of the inter-frame translation s can be computed as $s = -(bh)/d$. Second, the translational scale is also computed by relating the current and previous depth maps. The median of the multiplicative factor between corresponding depth values provides a robust translational scale estimate.

We can combine the scales estimated from the ground height and depth map for ground-based vehicles using a simple average. For aerial vehicles, we cannot ensure that the height from the ground is fixed, and thus, we estimate the scale using ground height for the first frame and rely on the scale from the depth map for subsequent frames. The

camera's height is continually updated using the reconstructed 3D points of the scene, which is only used to reinitialise the translational scale when not enough (<5%) 3D points from the previous estimate overlaps the current triangulated points. The estimated scale is then multiplied by the estimated camera translation, 3D reconstructed points, depth map, and uncertainty. The information matrix of the reconstructed 3D points is divided by the squared scale.

Given the dense optical flow, there are two methods to estimate the inter-frame poses depending on the magnitude of the pixel motion (or parallax). If there exists enough parallax, the Mahalanobis eight-point algorithm (Section 4) along with the estimated scale can be applied. If the motion is too small (e.g., hovering), the PnP method [32] becomes more effective in estimating the motion of the camera. The poses computed from two methods are averaged to improve the robustness of the solution. When the depth values have not converged accurately, the PnP estimate may return an error-prone result. Thus, we only perform the fusion when the difference in the estimated translation scale is within 30% of the estimated scale and the estimated rotations have a difference less than 0.5 radians. If either of these conditions is not met, we use the pose calculated from the Mahalanobis eight-point algorithm (Section 4) instead. The depth maps are fused using the pixels that are matched and triangulated. The previous frame's fused depth map and reconstructed scene points are propagated to the current image frame using the computed pose. This map provides prior 3D scene information for the next image frame.

The use of scaled-down images (one-third of the original scale) for dense optical flow estimation cannot guarantee the accuracy of the matches when the pixel translation between two images is too small. This case occurs when the vehicle moves very slowly or hovers, causing the motion estimation to be error-prone. Small translational motion estimation is a common problem in most monocular visual odometry, as the small parallax between two images leads to difficulty in estimating both motion and structure accurately. Two conditions determine the sufficient parallax. First, the Shi–Tomasi corner matches have a median displacement magnitude of at least 2.5 pixels. Secondly, the third quantile (75%) of the computed optical flow is larger than 5 pixels. Suppose either of the two conditions is not met. In that case, the inter-frame motion is computed only from the PnP method using the previously calculated depth and the motion of the corresponding pixels.

Loop closure (or revisiting the previous locations) detection is accomplished by using the robust linear pose-graph optimisation method from [17]. Similar to other pose-graph SLAM, this method treats all poses of the vehicle or robot as vertices and the inter-pose constraints (e.g., odometry and close loop constraints) as edges. The candidate frames for loop closure are found in three steps. The first step is by selecting frames with their estimated poses to be within a fixed distance while having a difference in frame index no less than a threshold value. The minimum frame index difference is enforced to prevent finding too many candidates within adjacent frames. We can further reduce the possible candidates by only finding candidate loop closure images for every 10 frames. The second step is to determine which of the candidate loop closure images are valid by using the structural similarity index (SSIM) [33]. Any images are discarded that have an SSIM index less than a set threshold (experimentally set to 0.38) and a maximum of three candidate images with the highest SSIM value are kept. Lastly, the dense optical flow between the images and their possible neighbours is computed. The estimated uncertainty is used to determine if the optical flow is reliable and only calculate their inter-frame motion when the percentage of matches with an uncertainty less than a set threshold is higher than 20% (an example is shown in Figure 4). During the loop closure, the motion estimation step also checks for the small-motion conditions discussed in the previous section, enabling loop-closures between temporally separated poses but spatially close to each other.

6. Experimental Results

We evaluated our proposed SLAM framework using the public KITTI dataset [34] and our UAV dataset. KITTI dataset shows a camera mounted on a vehicle traveling on

the roughly planar ground. Sequence 01, in particular, is a challenging highway scenario, where the car is moving at high speed, and there are few unique feature points within view. The UAV dataset shows a camera mounted on a quadcopter flying in a highly unstructured outdoor environment with dynamically moving objects. The UAV also performs motions such as (almost) pure rotation and extreme height variation. These make an accurate estimation of camera pose difficult for existing monocular visual odometry and SLAM.

6.1. Ground Vehicle

The dataset we used to verify the performance of our proposed algorithm for ground-based vehicles is taken from the KITTI benchmark. For optical flow evaluation, we use the flow 2015 dataset [35], while for the odometry, we use the odometry dataset [34]. For both experiments, we use the monocular RGB images (*image_2* folder). In the odometry experiment, we assume the camera is 1.7 m above the ground, with zero pitch. Due to the post-processing part of the DCFlow code not being made available, we can only verify the optical flow result before homography fitting is applied to the EpicFlow [24] interpolated results. Based on KITTI 2015 optical flow dataset, by utilizing our epipolar constraint on the cost volume, we achieved a 0.6% improvement in accuracy (regarding less than 3 pixels endpoint error criterion). The improvement is small because the epipolar truncation cost is set very low to accommodate for dynamic pixels in the scene. However, we can visually observe a noticeable improvement in the optical flow estimation for the ground pixels, not reflected by the significant (3 pixels error) KITTI accuracy metric. We also implemented a homography fitting step based on the description of their paper.

The uncertainty estimate for the dense optical flow is visually inspected, where it was observed that occluded, out-of-bound, or textureless regions of the image have high uncertainty value. For ground-based vehicle's visual odometry result, we compare our performance with existing methods, specifically VISO2-M [22], MLM-SFM [10], PMO [11] and DOF-1DU+LC [36]. We selected a few of the available sequences that contain a slow-moving vehicle in an urban environment (sequence 00), a fast-moving vehicle on a highway (sequence 01), and a vehicle traveling in a loop (sequence 06) to gauge the performance of our proposed methods. The results are summarised in Tables 1 and 2. Note that VISO2-M [22] and MLM-SFM [10] methods fail to estimate the visual odometry for sequence 01 due to the highly repeated structures of the scene, which cannot be reliably matched by the sparse feature-matching technique their methods employ. Figure 5 shows our estimated trajectory for the vehicle's motion. From the estimated motion trajectory (Figure 5) and computed error from the ground truth (Table 2), we can observe that our proposed method achieved a very accurate estimation of translation. This result is achieved without using bundle adjustment, motion model, or ground segmentation used by other state-of-the-art techniques. From Table 1, we can also observe an improvement in the rotation estimate after fusing the Mahalanobis eight-point algorithm and PnP result.

Table 1. Ablation study of our new proposed methods for selected KITTI dataset. "DOF-2DU" is the pose estimate of our Mahalanobis eight-point algorithm using dense optical flow with two-dimensional uncertainty, "+PnP" is the fused pose estimate with perspective-n-point, and "+LC" is the inclusion of loop closure.

seq	DOF-2DU		DOF-2DU + PnP		DOF-2DU + PnP + LC	
	rot (deg/m)	trans (%)	rot (deg/m)	trans (%)	rot (deg/m)	trans (%)
00	0.0076	1.80	0.0067	1.57	0.0045	1.07
01	0.0082	0.97	0.0050	1.03	0.0050	1.03
06	0.0047	0.96	0.0039	1.11	0.0039	1.17

Table 2. Comparison of visual odometry accuracy for VISO2-M [22], MLM-SFM [10], PMO [11], dense optical flow with 1D uncertainty and loop closure (DOF-1DU+LC) [36], and our new proposed methods (DOF-2DU+PnP+LC) for selected KITTI dataset.

seq	VISO2-M		MLM-SFM		PMO		DOF-1DU + LC		DOF-2DU + PnP + LC	
	rot (deg/m)	trans (%)	rot (deg/m)	trans (%)	rot (deg/m)	trans (%)	rot (deg/m)	trans (%)	rot (deg/m)	trans (%)
00	0.0209	11.91	0.0048	2.04	0.0042	1.09	0.0117	2.03	0.0045	1.07
01	n/a	n/a	n/a	n/a	0.0038	1.32	0.0107	1.149	0.0050	1.03
06	0.0157	4.74	0.0081	2.09	0.0044	1.31	0.0054	1.05	0.0039	1.17

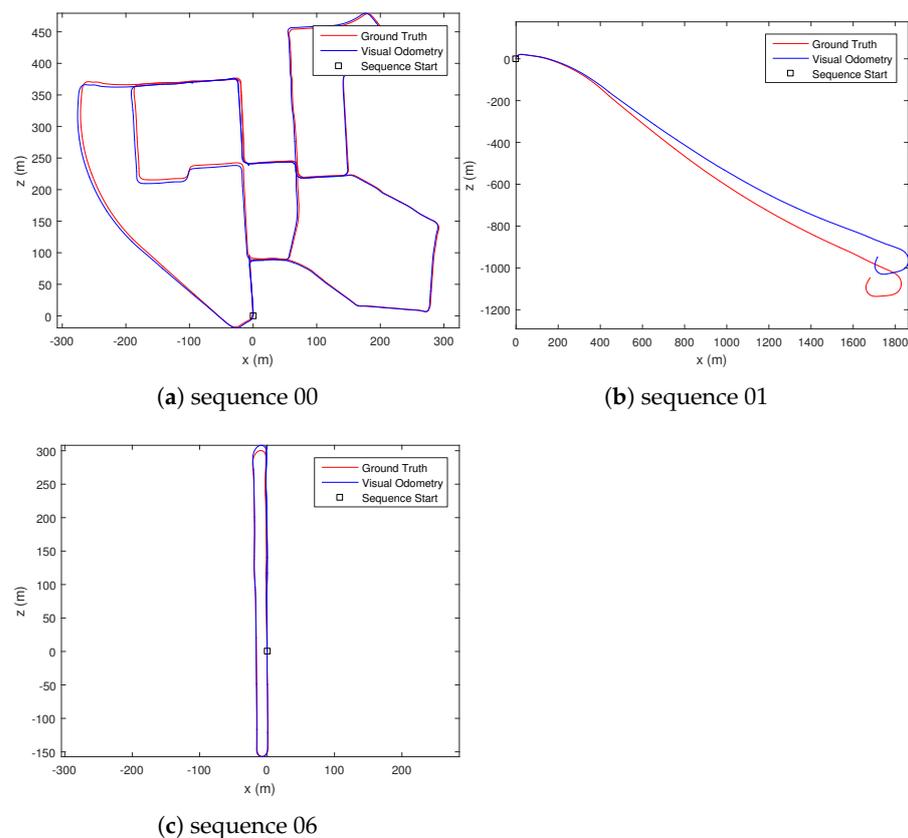


Figure 5. Comparison of the estimated motion trajectory and the ground truth motion in different sequences.

6.2. Aerial Vehicle

Since our visual odometry method does not rely on the restrictive motion model of the vehicle, we can easily apply our proposed method with slight modification to aerial vehicles (e.g., UAV). The difference with a ground-based vehicle is that the camera height is not assumed constant but is updated for each motion since the aerial vehicle can change its height arbitrarily. Another challenge of aerial visual odometry is that it can rotate its yaw with no translation, making pose estimation and 3D scene reconstruction highly under-constraint and error-prone. We also incorporated such motion in the video sequences we used in our experiment.

We captured 500 frames of video from a quadcopter flying among some trees near a road, where the scene has highly repetitive, unstructured, and dynamic objects (e.g., leaves, cars). Due to the lack of ground truth, unlike the KITTI dataset, we evaluate the scale drift by reversing the frames and appending them to the end of the video, where the last frame coincides with the first frame. Figure 6 shows the result of our experiment.

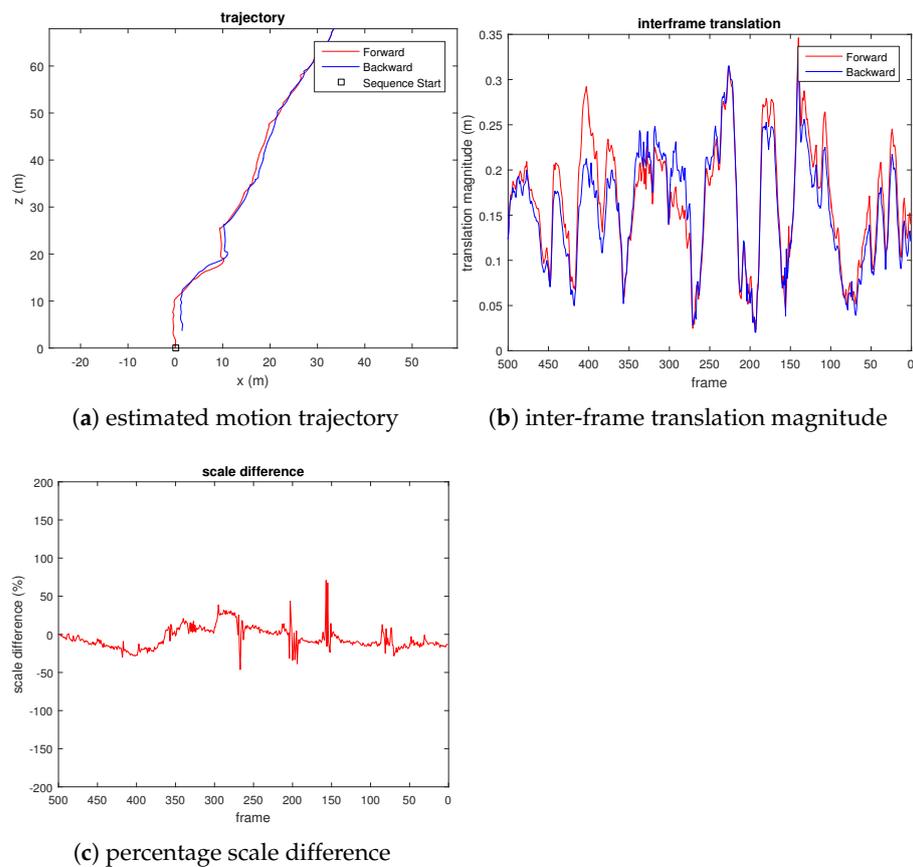


Figure 6. Plots evaluating the scale drift of our proposed visual odometry on UAV video, where percentage scale difference refers to the percentage error of the translation magnitude compared to the true translational magnitude.

Remark 1. We do not compute the loop closure constraint for this video sequence. The result shown in Figure 6 is pure visual odometry.

From the third plot of Figure 6, we can see that the translation scale difference remains close to zero, which shows that the scale drift is small. We also observed sudden spikes in the third plot, which corresponds to small motion, as can be seen from the central plot of Figure 6. As a comparison, we also evaluated the VISO2-M [22] method on the same UAV video, using the constant camera height assumption. Figure 7 shows the result. We observed that the estimated pose has a considerable translational magnitude (wrong) when the quadcopter rotates the yaw with negligible translational motion (e.g., at frames 200, 150, and 100). From the third plot of Figure 7, we can also see that although the estimated scale does not drift (due to fixed camera height assumption), the estimated translational magnitude fluctuates erratically throughout the video sequence.

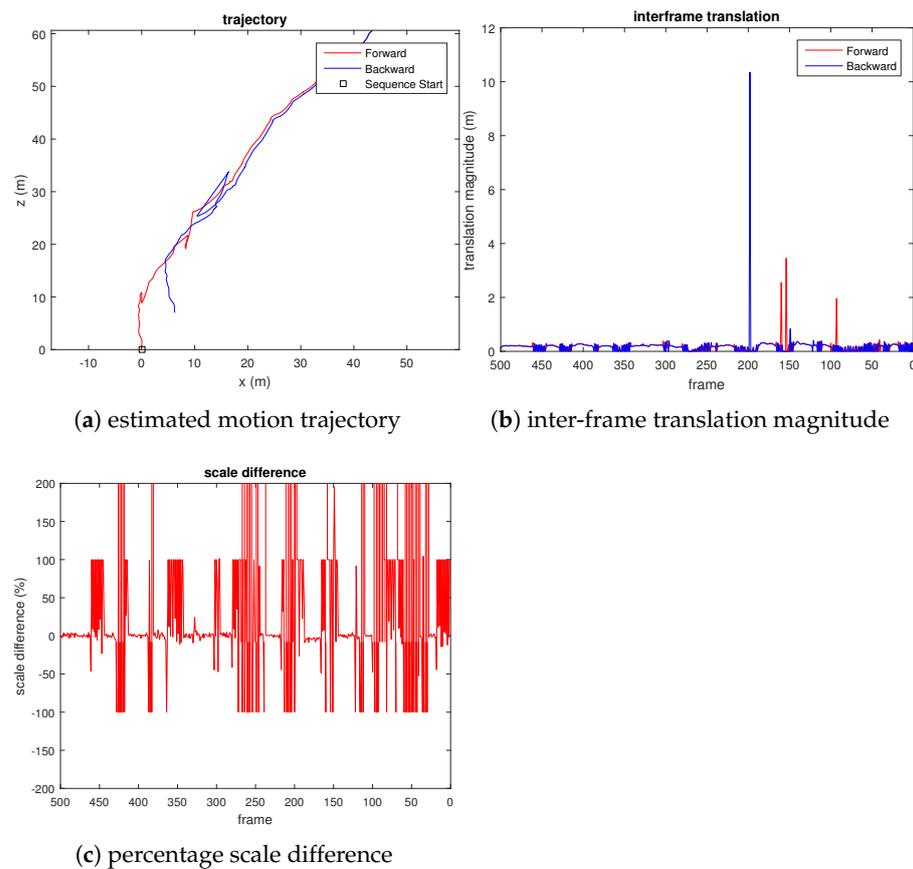


Figure 7. Plots evaluating the scale drift of VISO2-M [22] visual odometry on UAV video, where percentage scale difference refers to the percentage error of the translation magnitude compared to the true translational magnitude.

For the next experiment with UAV video, we captured 563 frames of a UAV flying at high speed with significant variation in height. We have also marked some trees with yellow tapes (1 m apart) to calibrate the first translational scale and obtain a measure of scale drift after the UAV returns to the same spot. The error in the estimated position can also be visually observed by comparing the location of the reconstructed scene points. We plot 3D scene points with a depth standard deviation less than 0.2 m for the first and last frames. Figures 8 and 9 show our results. From Figure 9c, we can see that the error of the estimated camera pose and reconstructed 3D scene points is minimal. The scale drift computed from the known distance between the tape is +5.36%. We have also calculated the distance between the farthest point from the starting location compared to GPS measurement and VISO2-M result. The result in Table 3 shows that our method agrees with GPS measurement more closely compared to the VISO2-M method. Thus, this verifies that our method can accurately estimate the camera motion, regardless of the motion dynamics of the vehicle or scene structure.

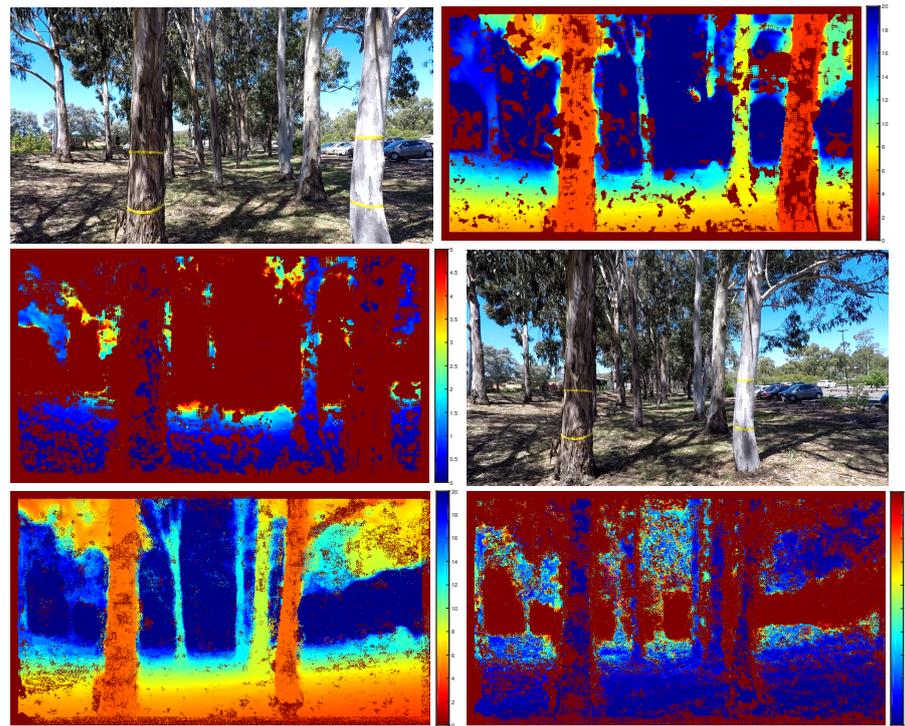
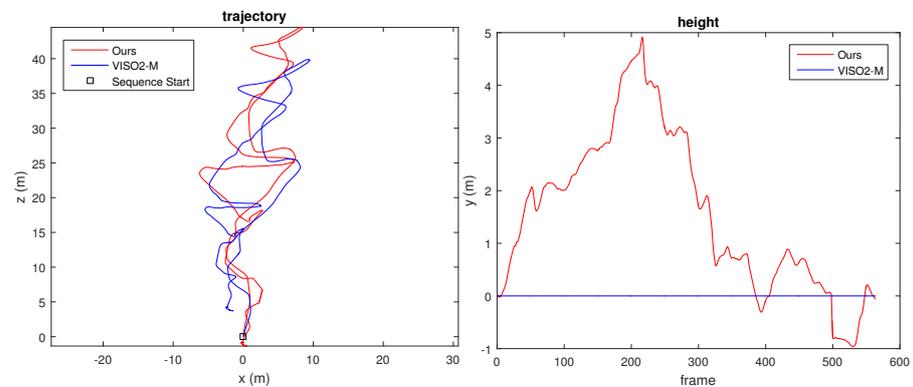
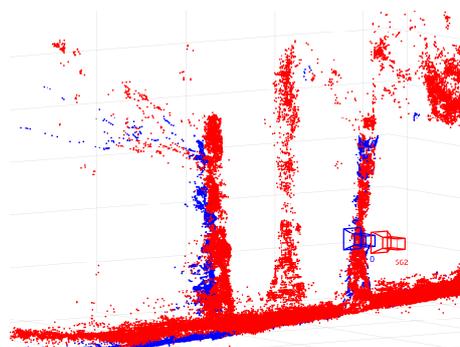


Figure 8. Estimated depth with standard deviation. From left to right: input frame, estimated depth, estimated depth standard deviation. The first row is frame 0, second row is frame 562. The scale of the colour code is in metres. Pixels that are identified as outliers are not triangulated and appear as dark red in the middle plot.



(a) estimated motion trajectory

(b) estimated UAV height



(c) our 3D reconstruction result of the first frame (blue) and last frame (red)

Figure 9. Fast-moving UAV video result, where the initial height is defined as 0 m in (b).

Table 3. Comparison of estimated distance of the farthest point from origin.

Method	Distance of Farthest Point to Origin (m)
GPS	45.81
Ours	45.50
VISO2-M	40.93

7. Conclusions

This paper presented a new Mahalanobis eight-point algorithm using the dense optical flow. The full uncertainty of the optical flow was estimated in a principled manner by using the negative logarithm of a bivariate Gaussian distribution and fitting it to the matching cost. The weighted eight-point algorithm optimised the Mahalanobis distance of each pixel correspondence to obtain a robust inter-frame motion estimate. With the SLAM pipeline of the front-end and back-end modules, the performance of the proposed method was evaluated using real datasets, demonstrating improved performance on a UAV platform compared to other state-of-the-art methods regarding accuracy and robustness. Future work includes the use of high-speed dense optical flow methods proposed as in [37] for real-time processing and collision avoidance in a cluttered environment.

Author Contributions: Conceptualisation, methodology, validation, and writing—original draft preparation are contributed by Y.N.; conceptualisation, writing—review and editing, and supervision are contributed by H.L. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Morgenthal, G.; Hallermann, N. Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures. *Adv. Struct. Eng.* **2014**, *17*, 289–302. [[CrossRef](#)]
- Nikolic, J.; Burri, M.; Rehder, J.; Leutenegger, S.; Huerzeler, C.; Siegwart, R. A UAV system for inspection of industrial facilities. In Proceedings of the Aerospace Conference, Big Sky, MT, USA, 2–9 March 2013; pp. 1–8.
- Herwitz, S.; Johnson, L.; Arvesen, J.; Higgins, R.; Leung, J.; Dunagan, S. Precision agriculture as a commercial application for solar-powered unmanned aerial vehicles. In Proceedings of the 1st UAV Conference, Portsmouth, VA, USA, 20–23 May 2002; p. 3404.
- Gago, J.; Douthe, C.; Coopman, R.; Gallego, P.; Ribas-Carbo, M.; Flexas, J.; Escalona, J.; Medrano, H. UAV's challenge to assess water stress for sustainable agriculture. *Agric. Water Manag.* **2015**, *153*, 9–19. [[CrossRef](#)]
- Bernard, M.; Kondak, K.; Maza, I.; Ollero, A. Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robot.* **2011**, *28*, 914–931. [[CrossRef](#)]
- Waharte, S.; Trigoni, N. Supporting search and rescue operations with UAVs. In Proceedings of the 2010 International Conference on Emerging Security Technologies (EST), Canterbury, UK, 6–7 September 2010; pp. 142–147.
- Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g 2 o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3607–3613.
- Stachniss, C.; Kretzschmar, H. Pose graph compression for laser-based slam. In *18th International Symposium on Robotics Research*; Springer: Puerto Varas, Chile, 2017; pp. 271–287.
- Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality NW, Washington, DC, USA, 13–16 November 2007; pp. 225–234.
- Song, S.; Chandraker, M.; Guest, C.C. High Accuracy Monocular SFM and Scale Correction for Autonomous Driving. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 730–743. [[CrossRef](#)] [[PubMed](#)]
- Fanani, N.; Stürck, A.; Ochs, M.; Bradler, H.; Mester, R. Predictive monocular odometry (PMO): What is possible without RANSAC and multiframe bundle adjustment? *Image Vis. Comput.* **2017**, *68*, 3–13.
- Bradler, H.; Anne Wiegand, B.; Mester, R. The Statistics of Driving Sequences—And What We Can Learn from Them. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, Santiago, Chile, 7–13 December 2015.
- Xu, J.; Ranftl, R.; Koltun, V. Accurate Optical Flow via Direct Cost Volume Processing. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.

14. Hui, T.W.; Tang, X.; Loy, C.C. LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8981–8989.
15. Sun, D.; Yang, X.; Liu, M.Y.; Kautz, J. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
16. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
17. Cheng, J.; Kim, J.; Shao, J.; Zhang, W. Robust linear pose graph-based SLAM. *Robot. Auton. Syst.* **2015**, *72*, 71–82. [[CrossRef](#)]
18. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *arXiv* **2016**, arXiv:1610.06475.
19. Engel, J.; Stücker, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015; pp. 1935–1942. [[CrossRef](#)]
20. Cheviron, T.; Hamel, T.; Mahony, R.; Baldwin, G. Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of UAV. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 2010–2016.
21. Artieda, J.; Sebastian, J.M.; Campoy, P.; Correa, J.F.; Mondragón, I.F.; Martínez, C.; Olivares, M. Visual 3-d slam from uavs. *J. Intell. Robot. Syst.* **2009**, *55*, 299. [[CrossRef](#)]
22. Geiger, A.; Ziegler, J.; Stiller, C. StereoScan: Dense 3D Reconstruction in Real-time. In *Intelligent Vehicles Symposium (IV)*; IEEE: Baden-Baden, Germany, 2011.
23. Chen, Q.; Koltun, V. Full flow: Optical flow estimation by global optimization over regular grids. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4706–4714.
24. Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; Schmid, C. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1164–1172.
25. Hartley, R.I. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 580–593. [[CrossRef](#)]
26. Torr, P.; Zisserman, A. Robust computation and parametrization of multiple view relations. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 7 January 1998; pp. 727–732.
27. Zhang, Z. Determining the Epipolar Geometry and its Uncertainty: A Review. *Int. J. Comput. Vis.* **1998**, *27*, 161–195. [[CrossRef](#)]
28. Armangué, X.; Salvi, J. Overall view regarding fundamental matrix estimation. *Image Vis. Comput.* **2003**, *21*, 205–220. [[CrossRef](#)]
29. Shi, J.; Tomasi, C. Good features to track. 1994. In Proceedings of the CVPR'94, 1994 IEEE Computer Society Conference, New York, NY, USA, 21–23 June 1994; pp. 593–600.
30. Mahalanobis, P.C. *On the Generalised Distance in Statistics*; National Institute of Sciences of India: Kolkata, India, 1936; pp. 49–55.
31. Douc, R.; Cappe, O. Comparison of resampling schemes for particle filtering. In Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, Zagreb, Croatia, 15–17 September 2005; pp. 64–69. [[CrossRef](#)]
32. Gao, X.S.; Hou, X.R.; Tang, J.; Cheng, H.F. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 930–943. [[CrossRef](#)]
33. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
34. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
35. Menze, M.; Geiger, A. Object Scene Flow for Autonomous Vehicles. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
36. Ng, Y.; Kim, J.; Li, H. Robust Dense Optical Flow with Uncertainty for Monocular Pose-Graph SLAM. In Proceedings of the Australasian Conference on Robotics and Automation, Sydney, Australia, 11–13 December 2017.
37. Adarve, J.D.; Mahony, R. A filter formulation for computing real time optical flow. *IEEE Robot. Autom. Lett.* **2016**, *1*, 1192–1199. [[CrossRef](#)]