Article

The International Journal of
Robotics Research
32(5) 526–546
© The Author(s) 2012
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364912472245
ijr.sagepub.com

Semi-parametric learning for visual odometry

Vitor Guizilini and Fabio Ramos

Abstract

This paper addresses the visual odometry problem from a machine learning perspective. Optical flow information from a single camera is used as input for a multiple-output Gaussian process (MOGP) framework, that estimates linear and angular camera velocities. This approach has several benefits. (1) It substitutes the need for conventional camera calibration, by introducing a semi-parametric model that is able to capture nuances that a strictly parametric geometric model struggles with. (2) It is able to recover absolute scale if a range sensor (e.g. a laser scanner) is used for ground-truth, provided that training and testing data share a certain similarity. (3) It is naturally able to provide measurement uncertainties. We extend the standard MOGP framework to include the ability to infer joint estimates (full covariance matrices) for both translation and rotation, taking advantage of the fact that all estimates are correlated since they are derived from the same vehicle. We also modify the common zero mean assumption of a Gaussian process to accommodate a standard geometric model of the camera, thus providing an initial estimate that is then further refined by the non-parametric model. Both Gaussian process hyperparameters and camera parameters are trained simultaneously, so there is still no need for traditional camera calibration, although if these values are known they can be used to speed up training. This approach has been tested in a wide variety of situations, both 2D in urban and off-road environments (two degrees of freedom) and 3D with unmanned aerial vehicles (six degrees of freedom), with results that are comparable to standard state-of-the-art visual odometry algorithms and even more traditional methods, such as wheel encoders and laser-based Iterative Closest Point. We also test its limits to generalize over environment changes by varying training and testing conditions independently, and also by changing cameras between training and testing.

Keywords

localization, Gaussian Processes, visual odometry, field robotics

1. Introduction

Accurate localization is a fundamental capability in autonomous navigation, where a vehicle needs to be constantly aware of its own pose to perform tasks such as mapping and path planning. There are basically two types of sensors that can be used to provide localization estimates: internal and external. Internal sensors (i.e. wheel encoders and inertial measurement units [IMUs]) work isolated from the external world and provide incremental position estimations, based on vehicle velocity and/or acceleration. This arrangement works well in small-scale experiments, but the estimates obtained have a tendency to drift over time due to error accumulation. External sensors (i.e. GPS, range finders and cameras) interact in one way or another with the environment around the vehicle, collecting information that can provide both incremental and absolute localization estimates. These absolute localization estimates are used to eliminate accumulated error, creating an upper bound on navigational uncertainty even after long periods. However, external sensors are sensitive to environment conditions that limit their applicability (GPS does not work indoors, range finders are constrained by object reflectability and their own resolution, cameras need adequate luminosity and sufficient texture on the surrounding environment).

Even so, the ability to eliminate accumulated error and provide precise localization estimations even after long

Australian Centre for Field Robotics, School of Information Technologies, University of Sydney, Australia

Corresponding author:
Vitor Guizilini, Australian Centre for Field Robotics, School of Information Technologies, The Rose Street Building J04, The University of Sydney NSW 2006, Australia.
Email: v.guizilini@acfr.usyd.edu.au

periods of navigation has made external sensors increasingly valuable in motion estimation over the last decades, both independently and in conjunction with other sensors. Of all external sensors, cameras are cheap, compact, with low power consumption, and have several other advantages that can lead to more robust and reliable results. Visual information is insensitive to terrain irregularities, is not restricted to any particular locomotion method, and when used for motion estimation are capable of providing predictions comparable in accuracy to most commercial IMUs (Howard, 2008). Also, recent increases in computational power allow real-time visual motion estimation on standard processors, and the information provided can be readily used in a wide range of other applications, such as object recognition (Lowe, 2004), object tracking (Tomasi and Tomasi, 1994) and map building (Davison, 2003), without the need for cross-calibration.

The process of estimating vehicle pose by analyzing its associated camera images is known as visual odometry, and is fundamentally composed of two stages. Initially, information from consecutive frames is extracted and correlated, to establish correspondences between features in overlapping areas that represent vehicle motion. If the environment is assumed static, any optical flow detected between frames is due to the camera's own motion and can be used to infer relative rotation and translation. Most visual odometry algorithms address this problem geometrically (Hartley and Zisserman, 2004), using a calibrated camera model to minimize the reprojection of 3D points triangulated from matched features. However, calibrating a system can be a daunting process, and there is no guarantee that these parameters will not vary over time, due to vibration, mechanical shocks or changes in temperature.

We propose here an alternative approach, where a Gaussian process (GP) (Rasmussen and Williams, 2006), a powerful non-parametric Bayesian inference technique, is used as a regression tool to learn the underlying function that maps optical flow information directly into camera motion. This is possible by exploring the optical flow pattern over different portions of the image, and using structure similarities to infer camera motion from a static environment. Training data is obtained from a different and independent sensor, and a likelihood function is optimized to fit this data, where a covariance function quantifies the relationship between points. Once the training is completed, the resulting model can be used to estimate translation and rotation between frames from visual information alone. The benefits of this approach are three-fold. (1) It substitutes the need for conventional camera calibration, by introducing a more flexible semi-parametric model that is capable of learning a much wider range of transformation functions from optical flow to vehicle motion. (2) It is able to recover absolute scale from a monocular configuration, by exploring structure similarity between images. (3) It naturally provides uncertainty estimates during the inference process. These benefits are obtained under the assumption that training and testing datasets share a certain similarity in optical flow distribution, and as this similarity decreases (i.e. the vehicle, camera or environment changes) so does the overall performance. A series of tests is conducted to show the impact of each of these changes in the final results.

We also propose several alterations to the standard GP framework, in order to take advantage of the structure of the visual odometry problem and improve results. Vehicle navigation has intrinsically multiple degrees of freedom (linear and angular velocities for each unconstrained axis), and since these velocities are subject to the same vehicle constraints it is natural to assume that they share some dependencies. We use a multiple-output Gaussian process (MOGP) to infer all velocities simultaneously (Boyle and Frean, 2005), incorporating these dependencies into the calculations and using them to eliminate ambiguity and gaps in the training data. We also extend the standard MOGP derivation to allow for joint estimation of all tasks (an extension we call coupled GP [CGP]) (Guizilini and Ramos, 2010), providing the means for a full covariance matrix recovery that can then be used in data fusion and incorporation into filtering techniques. In addition, we propose a novel temporal dependency extension to the CGP framework, where the outputs from one timestep are used as inputs in the next one, a strategy that has been proved valuable in situations where visual information is poor or ambiguous (Guizilini and Ramos, 2012). Finally, we propose a hybrid semi-parametric extension to the CGP framework, where a traditional geometric camera model is used to obtain an initial estimate of vehicle motion that is further refined by the non-parametric model obtained during training. The geometric model is used as the mean function for the CGP, and becomes more prominent as data obtained during navigation deviates from training data.

The rest of this paper is divided as follows. Section 2 provides a brief overview on visual odometry algorithms and multi-task learning methods, with an emphasis on GPs and how they have so far related to visual systems. Section 3 introduces our solution to the visual odometry problem, highlighting each stage of the algorithm from the initial image input to the final vehicle motion estimation output. Section 4 recapitulates the principles and fundamental equations behind GPs, moves on to MOGPs and then introduces the extensions proposed in this paper (CGPs, temporal dependencies and semi-parametric CGPs). Section 5 describes the mechanism used for optical flow extraction and parametrization, in such a way that it can be employed in machine learning inference. In Section 6 we present and discuss results obtained with the proposed methodology in both 2D and 3D scenarios, providing comparisons with other motion estimation algorithms. Finally, Section 7 concludes the paper and discusses future research directions.

2. Related work

The use of visual sensors to guide an autonomous vehicle can be traced back at least to 1976, with Moravec and Gennery (1976) using feature tracking for course correction in the Stanford AI Lab Cart. The functionality of these sensors was later extended to include egomotion estimation (Moravec, 1980), by tracking a set of assumed stationary feature points over a sequence of frames and calculating their relative shift. Over the past few decades similar approaches to visual odometry have been explored extensively and are becoming more and more popular as computational power increases, with applications in areas such as autonomous aircrafts (Kelly and Sukhatme, 2007; Huang et al., 2011), underwater vehicles (Corke et al., 2007; Botelho et al., 2009), space exploration (Cheng et al., 2005) and indoor/outdoor ground terrains (Campbell et al., 2004; Nister et al., 2006; Agrawal and Konolige, 2007; Howard, 2008; Scaramuzza and Siegwart, 2008; Tardif et al., 2008; Scaramuzza et al., 2009). Several modifications to the original scheme have also been proposed in an attempt to improve both quality and applicability of solutions: the use of omnidirectional cameras (Scaramuzza and Siegwart, 2008; Tardif et al., 2008), robust feature extraction and matching (Sunderhauf et al., 2005; Nister et al., 2006), data fusion with other sensors (Agrawal and Konolige, 2007; Kelly et al., 2007) and extension to a simultaneous localization and mapping (SLAM) framework (Se et al., 2001; Davison, 2003; Lemaire et al., 2007).

Visual odometry algorithms can be broadly divided into two categories: stereo and monocular configurations. Stereo configurations (Moravec, 1980; Zhu et al., 2006; Kelly and Sukhatme, 2007; Howard, 2008) use a multi-camera array to capture several images simultaneously, from different vantage points. If the baseline (distance between cameras) is known, it is possible to project detected features into the 3D space, and by tracking them over time to estimate vehicle motion. Monocular configurations (Scaramuzza and Siegwart, 2008; Tardif et al., 2008; Scaramuzza et al., 2009) use a single camera, which is essentially a bearing-only sensor. If a sequence of images taken at different locations is provided, the baseline between frames can be estimated, a scenario commonly known as the structure-from-motion (SFM) problem (Tomasi and Zhang, 1995). One well-known limitation of monocular odometry is the inability to recover absolute scale (Scaramuzza et al., 2009) from a single image, due to the parallax effect (an object could be far away and moving fast, or close by and moving slowly). Both approaches, stereo and monocular, can benefit from advances in feature extraction and matching techniques, such as the five-point and preemptive random sampling consensus (RANSAC) (Nister et al., 2006) and local bundle adjustment (Sunderhauf et al., 2005). The trade-off in resolution for wider field that omnidirectional cameras provide is also usually beneficial (Corke et al., 2004; Scaramuzza and Siegwart, 2008; Tardif et al., 2008; Scaramuzza et al., 2009), mostly because it allows detection of optical flow in any direction. The incorporation of uncertainty measurements to motion estimates allows fusion of visual odometry data with other sensors, such as an IMU (Kelly et al., 2007) or a low-cost GPS (Agrawal and Konolige, 2007), eliminating residual errors and accounting for situations where vision is not a reliable source of information (i.e. dark or textureless areas).

All of these approaches to visual odometry, however, are calibration-dependent, in the sense that the transformation from optical flow to vehicle motion is calculated using a geometric model (Hartley and Zisserman, 2004). This model is governed by the intrinsic parameters of the camera, and any imprecision will introduce a bias in the final estimation. Machine learning algorithms, on the other hand, eliminate the need of a parametric model by introducing a training dataset, which is used to optimize a cost function that quantifies the relationship between different points in the input space. The result is a non-parametric model that is capable of mapping directly from optical flow to vehicle motion, without the need of any prior knowledge of camera system or environment structure. Although intuitive, this approach has been scarcely used in visual odometry, most notably by Roberts et al. (2008) where the authors use a KNN-Learner voting method to estimate changes in pose, with each learner taking as input the average of the sparse optical flow in a grid-divided image. A similar idea is explored by Roberts et al. (2009), where a constant pixel depth is assumed and the expectation–maximization (EM) algorithm (Dellaert, 2002), in conjunction with an extension to PPCA (Tipping and Bishop, 1999), is used to learn a linear mapping between incremental motion and optical flow.

A machine learning technique that has been used with great success over the last few years in various areas of mobile robotics, such as mapping (O'Callaghan et al., 2009), terrain modelling (Vasudevan et al., 2009) and dynamic systems learning (Chai et al., 2008), are the GPs. A GP is a non-parametric regression and classification tool within the Bayesian statistical framework, where any finite linear combination of samples will be normally distributed. The standard derivation of a GP assumes one single output variable (task), using independent models to deal with multiple outputs when necessary. This approach, however, is detrimental when these variables are correlated (i.e. visual odometry, due to vehicle motion constraints), since knowledge of one could lead to a better estimation of all of the others. An alternative is the computation of a single covariance matrix containing observations from all tasks (Cressie, 1993), however in this scenario each inference is still conducted independently. A critical aspect of multi-task estimation is the definition of a valid positive-definite multi-task covariance function, that captures the underlying dependencies between each output variable. Boyle and Frean (2005) presented a convolutional method to define valid multi-task stationary covariance functions and Guizilini and Ramos (2010) presented and discussed an extension to multi-task GPs where all outputs are calculated simultaneously.

3. Algorithm overview

The algorithm proposed in this paper receives as input two images, obtained from a single non-calibrated camera, and returns as output the motion estimate between frames (including absolute scale), along with a full covariance matrix of uncertainties. It is divided into two stages, one concerning the parametrization of visual information into an input vector for the GP framework (optical flow parametrization), and one concerning the estimation of vehicle motion from this visual information alone (GP estimation). A diagram of all steps in each stage of the algorithm is presented in Figure 1, and the next few paragraphs are dedicated to briefly describing their function in the global scheme of the algorithm. An in-depth and more theoretical explanation of each stage is conducted in the next two sections.

Initially, the two input images are processed and each one produces a feature set, FTR_1 and FTR_2. These two feature sets are then matched to generate the matching set MTC_{12}, which describe the sparse optical flow information between frames. Since this matching set will most certainly contain outliers, due to wrong matches and dynamic objects, the fundamental matrix F_{12} is obtained using RANSAC, and the resulting inlier set INL_{12} is then transformed into a vector X_{12} that will serve as input for the GP framework.

Simultaneously, F_{12} and INL_{12} are also used to generate an initial motion estimate, based on a standard SFM algorithm. The geometric model that calculates this initial SFM estimate requires a calibrated camera, and its parameters are obtained from training data X_{TRN} and the corresponding ground-truth Y_{TRN} prior to the beginning of navigation, along with the GP hyperparameters. It is important to note that these are not equivalent to the camera's intrinsic parameters, as the geometric model in this framework provides only an initial guess for vehicle motion, that is further refined by the GP non-parametric inference process to generate the final estimate. The result is a semi-parametric approach to visual odometry that maps X_{12} directly into Y_{12}, which is the vehicle motion between frames. The full covariance matrix Σ_{12} is also obtained, quantifying not only the uncertainties between each component of Y_{12} (two in 2D non-holonomic navigation and six in 3D navigation) but also the cross-dependencies between them. Another way of viewing the proposed algorithm is as a novel calibration methodology in which the model being optimized is the CGP framework with the geometric model incorporated as the mean function (MCGP) itself, and the 'calibration parameters' are both the camera's intrinsic parameters and the GP hyperparameters. This is a much more powerful modelling tool, because it is capable of capturing nuances in the training data that a strictly geometric approach struggles with, due to limitations in the chosen model or imprecisions in the data capture process.

4. Gaussian processes

In the machine learning context, the estimation of vehicle motion from sensor information can be seen as a supervised regression problem: the process of mapping an input $\mathbf{x}$ to an output $y = f(\mathbf{x}) + \epsilon$ using a training dataset $\Lambda = \{\mathbf{x}_n, y_n\}_{n=1}^N$, where $\epsilon = \mathcal{N}(0, \sigma^2)$ represents a Gaussian noise with variance σ^2. In the specific case of visual systems, the vector $\mathbf{x}_n \in \Re^D$ contains optical flow information extracted from a pair of images and $y_n \in \Re$ contains the corresponding vehicle motion information, obtained for training purposes using a different and independent sensor. A GP places a Gaussian prior over the space of functions mapping inputs to outputs, using a positive-definite kernel (or covariance function) $k(\mathbf{x}_i, \mathbf{x}_j)$ that quantifies the relationship between points in the dataset. The parameters of this covariance function (or hyperparameters) are optimized based on a cost function that penalizes model complexity, as a way to avoid over-fitting according to the Occam's razor principle (MacKsay, 2002).

4.1. GPs overview

GPs (Rasmussen and Williams, 2006) are a non-parametric tool in the sense that they do not explicitly specify a functional model between inputs or outputs. Instead, they use information available in the training data Λ to quantify the relationship between different points in the input space, and then extrapolate this information to infer the output of new data in a probabilistic fashion. A GP model is entirely defined by a mean function $m(\mathbf{x})$ and a covariance function $k(x, x')$:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) . \tag{1}$$

If no prior knowledge of the underlying phenomenon is known, it is possible to assume $m(\mathbf{x}) = 0$ without loss of generality by scaling the data appropriately. The covariance function $k(x, x')$ is a positive-definite kernel whose coefficients are optimized to maximize a certain cost function (usually the marginal likelihood or leave-one-out cross-validation). Inference for a single test point $\mathbf{x}^*$, given training inputs $X = \{\mathbf{x}_n\}_{n=1}^N$ and outputs $\mathbf{y} = \{y_n\}_{n=1}^N$, involves the computation of the mean $\overline{f}(\mathbf{x}^*) = \overline{f^*}$ and variance $\mathcal{V}(f^*)$, and is calculated as follows:

$$\overline{f^*} = k(\mathbf{x}^*, X)^{\mathrm{T}} [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y} \tag{2}$$

$$\mathcal{V}(f^*) = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, X) [K(X, X) + \sigma_n^2 I]^{-1} k(\mathbf{x}^*, X) . \tag{3}$$

Both equations arrive naturally by conditioning the joint Gaussian distribution in Equation (1) on the observation $\mathbf{x}^*$. In Equations (2) and (3), σ_n^2 quantifies the noise expected in the observation y and K is the covariance matrix, with elements K_{ij} calculated based on a covariance function $k(\mathbf{x}, \mathbf{x}')$. A large number of covariance functions have been proposed over the years as a way to capture dependencies between

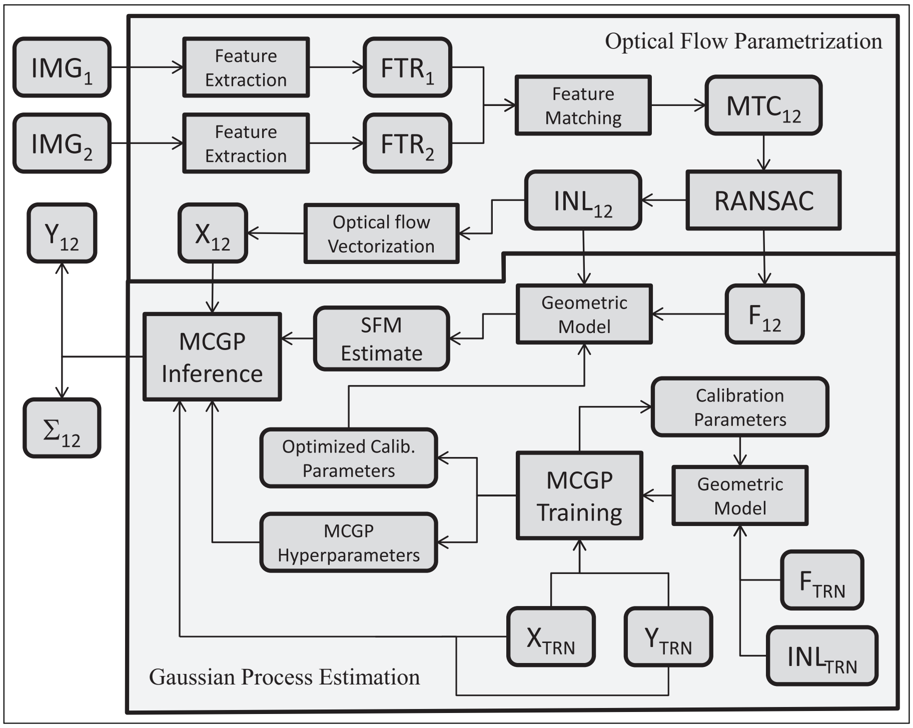


Fig. 1. Diagram of the proposed algorithm.

observations in kernel-based machines, and here we use the neural network covariance function (Williams, 1998) due to its non-stationary properties. The neural network covariance function can be derived from a neural network with a single hidden layer, a bias term and $H \rightarrow \infty$ hidden units. If the hidden weights are assumed to be Gaussian distributions with zero mean and covariance Σ the neural network covariance function can be defined (Neal, 1996) as

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \arcsin\left(\frac{2\tilde{\mathbf{x}}^{\mathrm{T}}\Sigma\tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^{\mathrm{T}}\Sigma\tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^{T}\Sigma\tilde{\mathbf{x}}')}}\right), \tag{4}$$

where $\tilde{\mathbf{x}} = (1, x_1, \ldots, x_D)^{\mathrm{T}}$ is an augmentation of $\mathbf{x}$ with the constant value 1, and σ_f^2 is a signal variance used to scale the correlation between points determined by the neural network covariance matrix Σ (here assumed diagonal with $D + 1$ eigenvalues). It should be noted that this formulation uses weighted dot products to quantify the relationship between $\mathbf{x}$ and $\mathbf{x}'$. As the dot product does not depend on the origin of the coordinate system, this covariance function is considered non-stationary, a valuable properties since visual odometry estimators include angular quantities. The expression also contains a sigmoid-like function, $\arcsin(x)$, to model sharp transitions and nonlinearities.

4.2. CGPs

The standard GP derivation, as described in the previous section, assumes a single output variable y (or task) for each input variable $\mathbf{x}$ (Figure 2(a)). Traditional implementations usually rely on multiple independent GPs to deal with multi-task scenarios, calculating each output separately based on the same input information. This, however, is not an ideal solution when these output variables are in some way correlated, which is the case in visual odometry. In the 3D space, six parameters (degrees of freedom) are necessary to describe vehicle motion: three for translation ($\dot{x}, \dot{y}, \dot{z}$) and three for rotation ($\dot{\gamma}$, $\dot{\beta}$ and $\dot{\alpha}$, in Euler angles). Since these parameters are constrained by the vehicle motion model, it is reasonable to assume that there will be correlations between different degrees of freedom, which if exploited could lead to better localization results.

The approach we describe here is based on the ideas of MOGPs presented by Bonilla et al. (2008) and Boyle and Frean (2005), where cross-correlations are explored through the definition of a valid multi-task covariance function (Figure 2(b)). First of all, the training dataset is expanded to include all T tasks, assuming the form $\Lambda = \{\Lambda_t\}_{t=1}^{T}$ where $\Lambda_t = \{\mathbf{x}_n, y_{(t,n)}\}_{n=1}^{N}$.[1] The covariance matrix K is now defined as

$$K = K_f \otimes K_x + \Sigma_n, \tag{5}$$

where $\otimes$ denotes the Kronecker product, K_f is a $T \times T$ positive-definite matrix that models the amplitude of correlations between each task (a multi-task analog to σ_f^2 in Equation (4)), and Σ_n is a diagonal matrix with noise

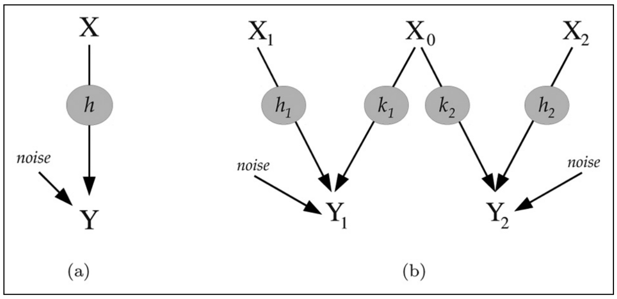


Fig. 2. Diagrams for (a) single-output GP and (b) MOGP (Boyle and Frean, 2005). In (a), the output Y is the sum of two Gaussian white noise processes (X and *noise*), one of which has been convolved with a kernel h. In (b), k is the multi-task kernel and X_0 represents the correlated portion between the two tasks. If X_0 is forced to be zero the outputs Y_1 and Y_2 become independent and the problem reverts to the single-output GP case described in (a).

values. K_x is a $T \times T$ block-matrix defined as

$$K_x = \begin{bmatrix} K_{11} & \dots & K_{1T} \\ \vdots & \ddots & \vdots \\ K_{T1} & \dots & K_{TT} \end{bmatrix}, \tag{6}$$

where

$$K_{ij} = \begin{bmatrix} k_{ij}(\mathbf{x}_1, \mathbf{x}_1) & \dots & k_{ij}(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k_{ij}(\mathbf{x}_N, \mathbf{x}_1) & \dots & k_{ij}(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{7}$$

is the covariance matrix between tasks i and j, calculated using the covariance function $k_{ij}(\mathbf{x}, \mathbf{x}')$. When $i = j$ the standard auto-covariance function (Equation (4)) is used (with the only exception of σ_f^2 which is removed, since in this derivation this parameter is substituted by K_f). When $i \neq j$ a cross-covariance function is used, derived (Higdon, 2002) from the definition of a neural network function in which two smoothing kernels are convolved to obtain a positive-definite function that correlates multiple outputs:

$$k_{ij}(\mathbf{x}, \mathbf{x}') = \frac{\arcsin\left(\frac{2\widetilde{\mathbf{x}}^{\mathrm{T}}\Sigma\widetilde{\mathbf{x}}'}{\sqrt{(1+2\widetilde{\mathbf{x}}^{\mathrm{T}}\Sigma\widetilde{\mathbf{x}})(1+2\widetilde{\mathbf{x}'}^{\mathrm{T}}\Sigma\widetilde{\mathbf{x}'})}}\right)}{(|\Sigma_i||\Sigma_j|)^4\sqrt{|\Sigma_i + \Sigma_j|}}. \tag{8}$$

In the above, $\widetilde{\mathbf{x}} = (1, x_1, \dots, x_D)^{\mathrm{T}}$ is again an augmentation of $\mathbf{x}$ with the constant value 1, $\Sigma = (\Sigma_i(\Sigma_i + \Sigma_j)^{-1}\Sigma_j)$ and Σ_t is a diagonal matrix containing length scale hyperparameters for task t.

4.2.1. Training stage During training, the CGP hyperparameters are optimized according to an objective function, here chosen to be the log-marginal likelihood

$$\zeta = \ln p(\mathbf{y}|X) = -\frac{1}{2}\log(|K|) - \frac{1}{2}\mathbf{y}^{\mathrm{T}}K^{-1}\mathbf{y} - N\log(2\pi) \tag{9}$$

due to its ability to balance between data fit and model complexity (Hastie et al., 2001), thus minimizing the chances of over-fitting. In the CGP framework proposed here, these hyperparameters are the diagonal elements of Σ_t for all tasks (length scales), the coefficients of K_f (amplitudes of correlation) and the diagonal elements of Σ_n (noise levels). The optimization is conducted using a combination of stochastic maximization (simulated annealing) and gradient descent algorithms to reduce the influence of initial conditions. The stochastic maximization is necessary because of the high number of hyperparameters, which creates a very high-dimensional problem and increases the chance of local maxima.

As usual, training and testing datasets should be obtained under similar conditions, which in the visual odometry context means using the same camera configuration to ensure a similar optical flow distribution. Any deviance from this would increase estimation uncertainty, since the observations obtained during navigation no longer correspond to those used to derive the underlying model during the training stage.

4.2.2. Inference stage The main contribution of CGPs over the standard MOGP derivation, as described by Boyle and Frean (2005), is the inference methodology. In the standard MOGP derivation, even though the inference for each task is obtained based on observations from all tasks, each one is still calculated independently, so there is no estimation of the cross-correlation terms. In other words, there is no estimation of a full covariance matrix of uncertainty for all tasks, each one is calculated as a single variable and this information is not incorporated into the framework (the resulting covariance matrix is assumed to be diagonal).

The CGP framework circumvents this limitation by introducing K_s as a T-column matrix containing the covariance function values k_{ij} computed between the test point $\mathbf{x}^*$ and the training points for all tasks. For a test point $\mathbf{x}^*$, the resulting mean vector $\overline{\mathbf{f}^*}$ and covariance matrix $\mathcal{V}(\mathbf{f}^*)$ are now defined as

$$\overline{\mathbf{f}_t^*} = K_s^{\mathrm{T}}K^{-1}\mathbf{y} \tag{10}$$

$$\mathcal{V}(\mathbf{f}_t^*) = K_{ii}(\mathbf{x}^*, \mathbf{x}^*) - K_s^{\mathrm{T}}K^{-1}K_s, \tag{11}$$

where

$$K_s = \begin{bmatrix} k_{1,1}^f k_{1,1}(\mathbf{x}^*, \mathbf{x}_1) & \dots & k_{T,1}^f k_{T,1}(\mathbf{x}^*, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ k_{1,1}^f k_{1,1}(\mathbf{x}^*, \mathbf{x}_N) & \dots & k_{T,1}^f k_{T,1}(\mathbf{x}^*, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k_{1,T}^f k_{1,T}(\mathbf{x}^*, \mathbf{x}_1) & \dots & k_{T,T}^f k_{T,T}(\mathbf{x}^*, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ k_{1,T}^f k_{1,T}(\mathbf{x}^*, \mathbf{x}_N) & \dots & k_{T,T}^f k_{T,T}(\mathbf{x}^*, \mathbf{x}_N) \end{bmatrix} \tag{12}$$

and

$$\mathbf{y} = [y_{1,1} \dots y_{1,N} \dots\dots y_{T,1} \dots y_{T,N}]. \tag{13}$$

Algorithm 1 Temporal dependency training.

```
Require: Training Datasets $\Lambda^1$ and $\Lambda^2$
         Initial Hyperparameters $\theta$
Ensure: Optimized Hyperparameters $\theta$
 1: $likelihood\_new \leftarrow \infty$
 2: repeat
 3:   $likelihood\_old = likelihood\_new$
 4:   for $\mathbf{x}_i$ in $\Lambda^1$ do
 5:     $Z_i^1 \leftarrow (\mathbf{x}_i, \mathbf{y}_{i-1}^1)$
 6:   end for
 7:   % Expectation step
 8:   for $\mathbf{x}_i$ in $\Lambda^2$ do
 9:     $\mathbf{y}_{CGP} = CGP\_INFER(Z^1, \mathbf{x}_i, \theta)$
10:     $Z_i^2 \leftarrow (\mathbf{x}_i, \mathbf{y}_{CGP})$
11:   end for
12:   % Maximization step
13:   $(likelihood\_new, \theta) = CGP\_TRAIN(Z^2, \mathbf{y}^2, \theta)$
14:   $\Lambda^1 \leftrightarrows \Lambda^2$
15: until $likelihood\_new - likelihood\_old = 0$
```

4.3. *Temporal dependency*

In the previous section, we addressed the nature of cross-correlations between tasks, which is a natural assumption in visual odometry applications. This is however not the only one, and here we explore another type of correlation between tasks, which is temporal dependency. It is safe to assume that a real vehicle will change its velocity in a smooth manner, without discontinuities, and therefore its motion estimates will also vary smoothly over time. A first-order temporal dependency between tasks implies that $\overline{\mathbf{f}}_k^*$ will be correlated to $\overline{\mathbf{f}}^*_{k-1}$, with k being the timestep between frames. This is modelled into the CGP framework by incorporating $\overline{\mathbf{f}}^*_{k-1}$ into the input vector $\mathbf{x}_k$. So, for a test point with optical information $\mathbf{x}_k^*$ the new augmented input vector becomes

$$\mathbf{z}_k^* = \{\mathbf{x}_k^*, \overline{\mathbf{f}}^*_{k-1}\}. \tag{14}$$

The introduction of $\mathbf{z}$ as an augmented input vector does not interfere with the CGP inference methodology, other than requiring the corresponding augmentation of the length-scale matrices Σ_t to deal with the new input dimensions that were incorporated. However, this new setup disturbs the training methodology, because the complete set of observations Z (the analog of X in Equation (9)) is not readily available for evaluation, since it needs to be calculated iteratively. It is possible to use ground-truth information to complete Z, but this would generate a best-case scenario that is not consistent with the inference stage, where estimation errors tend to propagate over successive iterations.

We propose here a new training methodology (described step-by-step in Algorithm 1) that allows the incorporation of these estimation errors to the final non-parametric model while maintaining a first-order temporal dependency between tasks. First of all, the training dataset Λ is divided into two subsets, Λ^1 and Λ^2, each composed of half the training data. In the first subset, the ground-truth values of $\mathbf{y}^1$ are used to complete Z^1 directly (lines 4–6), in such a manner that $\mathbf{y}_{k-1}^1$ completes Z_k^1. The observation set Z^1 is then used to evaluate Z^2 iteratively (lines 8–11), employing the CGP inference methodology described previously.

Once the evaluation process is complete the estimated Z^2 is used to optimize the CGP hyperparameters (line 13), according to a gradient-descent method and based on the log-marginal likelihood function (Equation (9)). Once the optimization is complete, the process is repeated with inverted subsets (Λ^2 is now used for inference and Λ^1 for training) until the cost function converges (lines 14 and 15). It was determined empirically that the hyperparameters assigned as length scales for $\overline{\mathbf{f}}^*_{k-1}$ should be kept from assuming too low values, since this would increase the sensitivity to small errors in estimation. Also, the gradient-descent method should be stopped after a few steps, in order to avoid over-fitting in any particular iteration of the training process.

This technique resembles the EM algorithm (Dellaert, 2002), in the sense that it alternates between computing motion estimates from current hyperparameters (the expectation step) and optimizing hyperparameter values using current motion estimates (the maximization step). Also, there is no guarantee of convergence to the global minimum, so heuristic approaches for escaping local minima, such as random restart or simulated annealing, should be considered.

4.4. *Semi-parametric coupled GPs*

As stated previously, most GP implementations assume that the mean value of the input information $m(\mathbf{x})$ is zero, indicating no prior knowledge of the underlying function to be inferred from training data. However, this is not the case in visual odometry, since it is also possible to obtain a estimate of vehicle motion from well-established geometric models (Hartley and Zisserman, 2004). These models depend strictly on camera calibration parameters, and are commonly used as stand-alone solutions to the SFM problem (Sunderhauf et al., 2005; Nister et al., 2006; Howard, 2008).

We propose here the incorporation of this geometric model into the CGP framework, creating a semi-parametric approach to visual odometry that benefits from both the SFM and machine learning strengths. This is achieved by introducing the geometric model estimates as the new mean vector $m(\mathbf{x})$, which provides an initial estimate that is further refined using the training dataset. This initial estimate is obtained via triangulation, based on a calibrated camera model and a set of matched features (the same ones used to obtain the optical flow information that serves as input for the CGP). If these matched features are assumed to be static and their projections on both images are known (Figure 3), it is possible to use this information to constrain camera

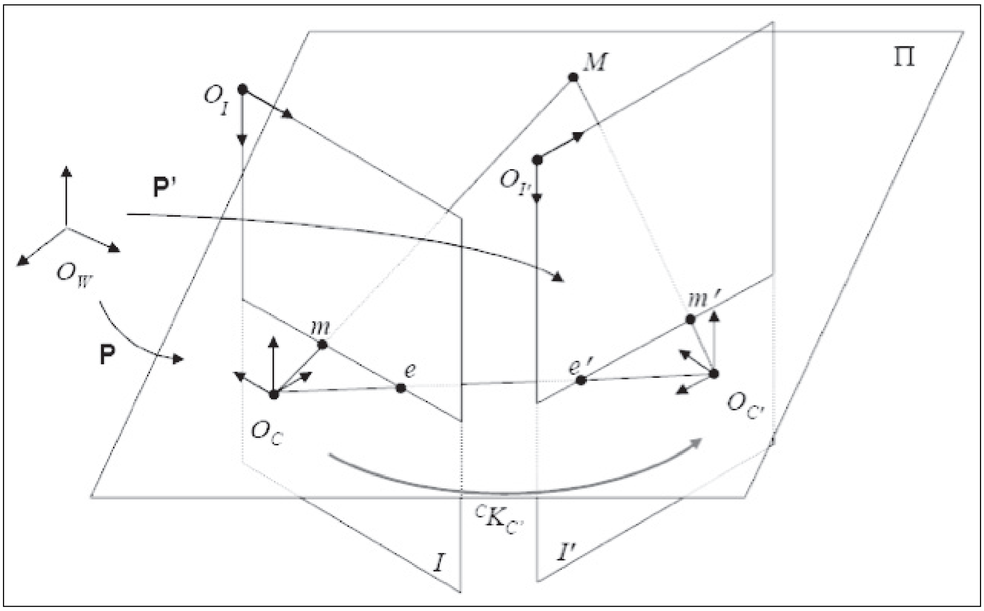


Fig. 3. Diagram of the geometrical constraints used to estimate vehicle translation from O_C to $O_{C'}$ according to a matched feature M and its projections m and m' on each image.

motion between frames and obtain an estimate of translation and rotation. Even though there are several methods that should produce better results (Howard, 2008; Lovegrove et al., 2011), we choose here to use the SFM values directly as the mean vector to test our framework's ability to improve on generally poor estimations. It is natural to assume that more involved geometric models should produce better results, which would then translate into better estimations for the CGP framework to build upon.

The first step is the calculation of the fundamental matrix, based on a set U of $N > 7$ matched features between consecutive frames I and I' (the particular techniques used to obtain these matched features are detailed in Section 4). If $U = \mathbf{u}_{n=1}^{N}$ and $\mathbf{u}_n = (u, v, 1)^{\mathrm{T}}$ contains the homogeneous image coordinates of each individual feature in a particular frame, the fundamental matrix F is given by the optimization of $\mathbf{u}$ in

$$\mathbf{u}'^{\mathrm{T}} F \mathbf{u} = 0. \tag{15}$$

The fundamental matrix is a 3 × 3 matrix that relates corresponding points in a pair of images. If $\mathbf{u}$ describes the homogeneous coordinates of a feature in frame I, $F\mathbf{u}$ will describe a line (known as an epipolar line) in frame I' on which this feature must lie (Figure 4). This relation, however, does not take into account the metrics of camera calibration, a crucial aspect in estimation using a geometric model. The Essential matrix incorporates these metrics by introducing a calibration matrix C defined as

$$C = \begin{bmatrix} l_x & s & p_x \\ 0 & l_y & p_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

where l_x and l_y are focal lengths, s is the skew parameter and p_x and p_y are the image center coordinates (collectively known as the camera intrinsic parameters). If the fundamental matrix F and the calibration matrix for both images C and C' are known, the essential matrix E can be obtained as follows:

$$E = C'^{\mathrm{T}} F C. \tag{17}$$

From the essential matrix it is possible to calculate the camera's extrinsic parameters (translation $\mathbf{t}$ and rotation R) by identifying the correct pair of projection matrices P and P', which represent the camera pose at the instant each image is taken according to a global coordinate system. If $P = [I|0]$, meaning that the camera begins at the center of the coordinate system, then $P' = [R|\mathbf{t}]$ indicates camera motion between frames. This camera motion is then parametrized as the new mean vector for the CGP framework, so $m(\mathbf{x}) = \{\dot{x}, \dot{y}, \dot{z}, \dot{\gamma}, \dot{\beta}, \dot{\alpha}\}$, where $(\dot{x}, \dot{y}, \dot{z})$ are the coefficients of $\mathbf{t}$ (transformed back into the camera's local coordinate system) and $(\dot{\gamma}, \dot{\beta}, \dot{\alpha})$ represent the camera's current orientation in Euler angles (obtained from R).

4.4.1. Training stage Training on the semi-parametric CGP framework is conducted as described previously, with two exceptions. First of all, Equation (9) has to be slightly altered to account for the fact that $m(\mathbf{x})$ is no longer assumed to be zero. This is achieved by defining $\epsilon = (\mathbf{y} - m(\mathbf{x}))$ as the difference between the ground-truth information $\mathbf{y}$ and the mean vector $m(\mathbf{x})$. If this value is small, it means that the geometric model is doing a good job and there is no need for further refinement of the estimate. If this value is large, the non-parametric model takes over and compensates the difference. The new log-marginal

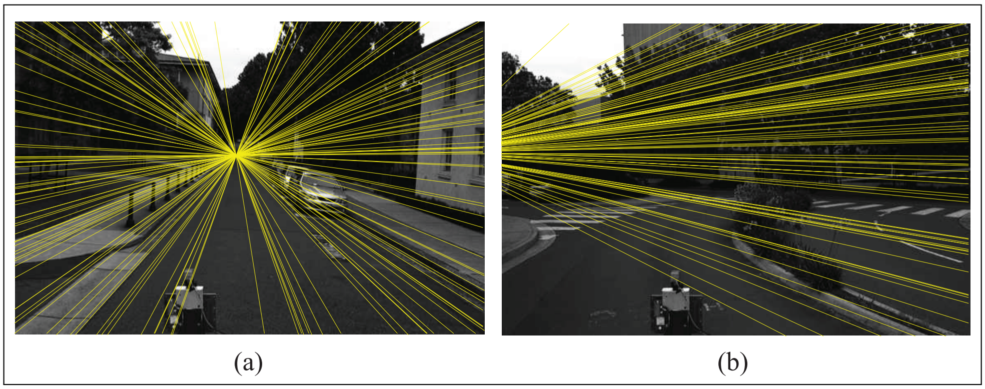


Fig. 4. Examples of epipolar lines during vehicle translation (a) and rotation (b).

likelihood cost function now becomes

$$\zeta = \ln p(\mathbf{y}|X) = -\frac{1}{2}\log(|K|) - \frac{1}{2}\epsilon^{\mathrm{T}}K^{-1}\epsilon - N\log(2\pi)\,. \tag{18}$$

Also, the introduction of a geometric model into the CGP framework introduces a new set of hyperparameters, the calibration parameters present in C (focal lengths l_x and l_y, skew s and image center coordinates p_x and p_y). A straightforward solution to this problem would be to provide these calibration parameters manually, however we propose here their incorporation into the optimization process. This approach maintains the CGP assumption that no traditional camera calibration is necessary, and if these parameters are known they can be used as the initial guess during the optimization process. In fact, since the geometric model is now used in tandem with the non-parametric model, the final calibration parameters may differ from those provided by an independent calibration.

4.4.2. Inference stage Inference for a single test point $\mathbf{x}^*$ is now conducted according to Equations (19) and (20). By adding the mean function to $\overline{\mathbf{f}}^*$ we assure that, as the testing point $\mathbf{x}^*$ deviates from the training dataset Λ, the outputs converge to the estimates provided by the geometric model. As expected, $\mathcal{V}(\overline{\mathbf{f}}^*)$ remains unaltered by the introduction of $m(\mathbf{x})$ as a non-zero term, since the geometric model is not capable of providing any uncertainty estimates. The new equations are

$$\overline{\mathbf{f}}^* = m(\mathbf{x}^*) + K_s^{\mathrm{T}}K^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{19}$$

$$\mathcal{V}(\overline{\mathbf{f}}^*) = K_{ii}(\mathbf{x}^*, \mathbf{x}^*) - K_s^{\mathrm{T}}K^{-1}K_s. \tag{20}$$

5. Optical flow parametrization

Our method uses sparse optical flow information, extracted from consecutive pairs of monochromatic images obtained using a single camera configuration. This optical flow information is then processed to generate the vectors $\mathbf{x}_k$ that will serve as input for the CGP framework described in the previous section. We assume that most of the environment around the vehicle is static (any optical flow detected is due solely to camera motion), and we also assume that the frames-per-second rate is constant throughout navigation (which is important for absolute scale recovery based on training information). No other prior knowledge of the environment or the visual system is necessary. A histogram filter is initially applied to all images to minimize the effects of global changes in luminosity.

5.1. Feature extraction

Owing to its robustness and invariance properties, the feature extraction and matching processes are performed using the scale-invariant feature transform (SIFT) algorithm (Lowe, 2004) with sub-pixel accuracy and frame-to-frame tracking, although any other similar method could be readily applied for speed purposes (Bay et al., 2006) or to increase the amount of information obtained from each frame.[2] Examples of initial feature sets in 2D navigation for a particular frame are shown in Figure 5(a), and their corresponding matching sets in relation to the subsequent frame are shown in Figure 5(b), where each matching pair is connected with a line. It is possible to see a substantial amount of false matches, mostly due to structure similarity, poorly texturized regions and occlusion caused by changes in viewpoint.

To remove these false matches (or outliers) we use the seven-point RANSAC algorithm (Fischler and Bolles, 1981), a probabilistic tool that elects the predominant motion hypothesis between frames and discards matches that do not comply to the constraints it imposes. These constraints are calculated based on the geometric model described in the previous section, and if most features in the environment are assumed static the predominant motion hypothesis will be the camera's own motion. This step is also useful in minimizing the impact of dynamic objects, since their features will generate an optical flow that is not consistent with the rest of the image and therefore will be eliminated as false matches. The resulting feature sets are presented in Figure 5(c), and they constitute the sparse optical flow information that best represents vehicle motion between frames according to this framework. Features are tracked for an average of six frames, and the overlapping

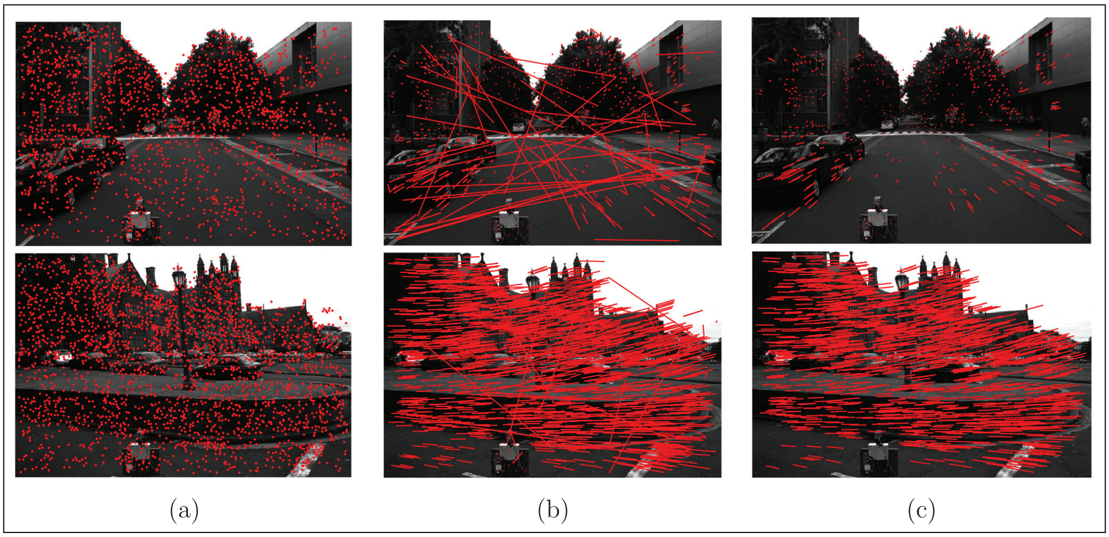


Fig. 5. Examples of the three stages of feature extraction for translation (top line) and rotation (bottom line) motion in 2D navigation: (a) initial features; (b) initial matches; (c) matches after RANSAC.

regions range from 90% (forward motion) to 75% (hard turns).

5.2. *Preserving spatial structure*

A straightforward way of generating the input vector $\mathbf{x}_k$ would be to use the individual optical flow information from each matching pair directly. In this scenario, $\mathbf{x}_k$ would be a vector of size $2N$, where N is the number of successful matches and 2 is the number of optical flow components. However, the direct use of individual optical flow parameters to generate $\mathbf{x}_k$ would incur in two problems. First of all, two different pairs of images will most certainly produce matching sets of different sizes, thus changing the dimension of $\mathbf{x}_k$ and the nature of the underlying function. Also, two different pairs of images will most certainly produce matching sets that are distributed differently throughout the image, and since optical flow information is heavily dependent on pixel coordinate (each region of the image reacts differently to camera motion) any comparison would be rendered moot.

It is therefore necessary to generate an input vector $\mathbf{x}_k$ that both has a constant dimension regardless of the number of matching pairs and also maintains the spatial structure of optical flow distribution. Our method to achieve these two requirements consists in dividing the image into equal-sized rectangles (Figure 6), and assigning to each of them the subset of matched features whose coordinates lie within its boundaries (by convention, we use the feature coordinates on the first frame). The optical flow parameters for each rectangle can now be calculated as the average value of all of its matched features' optical flow information. If a particular rectangle has no features, its optical flow parameters are calculated as the average value of its surrounding rectangles, based on the assumption that changes in optical flow should be smooth throughout the image. The input vector $\mathbf{x}_k$ is now of dimension $2hw$, where h and w are the numbers of rectangles the image was divided into vertically and horizontally, respectively, and is generated by taking the optical flow components for each rectangle in a specific manner (i.e. starting on the top left rectangle and moving horizontally line by line).

5.3. *3D navigation*

In the 3D scenario the camera was pointing downwards, in such a way that each image captures what is below the aircraft. This configuration poses a challenge in both feature extraction and matching, due to loss in detail and sensitivity to angular motion that translates into inconsistent (and often small) overlapping areas between frames (Figure 7(a)). Assuming that the aircraft will maintain a considerable altitude and move roughly horizontally, it is reasonable to consider the ground plane as homogeneous, and therefore the entire image will share the same optical flow information. This assumption allows the optical flow information to be encoded as a single pair of parameters, here the average shift distance d and angle θ of all successful matches. The position $(x, y)_k$ and size $(h, w)_k$ of the overlapping regions in the image are also related to camera motion (see Figure 7(b)), and therefore contain information that could be useful in the inference process.

6. Experimental results

The methodology described in this paper was first evaluated in 2D environments, using data collected from a ground vehicle (Figure 8(a)) navigating in outdoor environments (both urban and off-road). In this scenario only two tasks (forward translation and rotation on the z-axis) are necessary to describe vehicle motion between frames, which translates into fewer hyperparameters, less computational

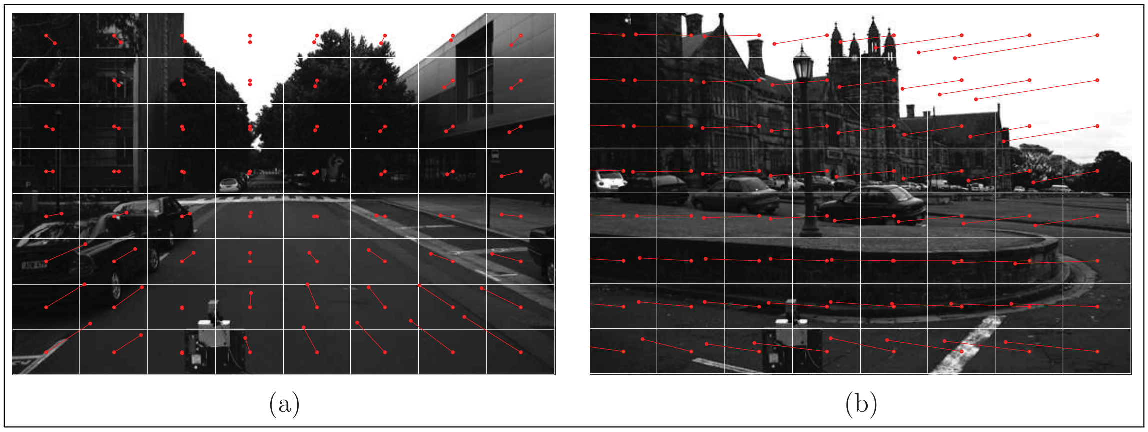


Fig. 6. Examples of optical flow parameters for translation (a) and rotation (b) motion in 2D navigation.

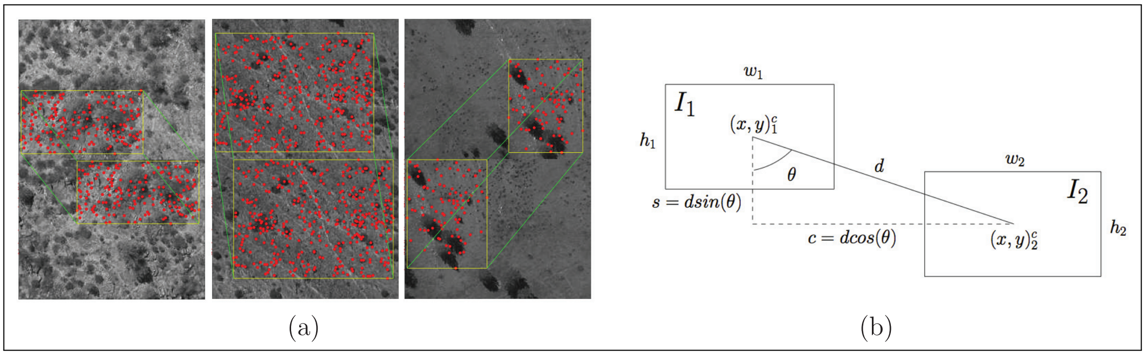


Fig. 7. Optical flow parametrization in 3D navigation: (a) examples of matching sets; (b) diagram for optical flow parameters.

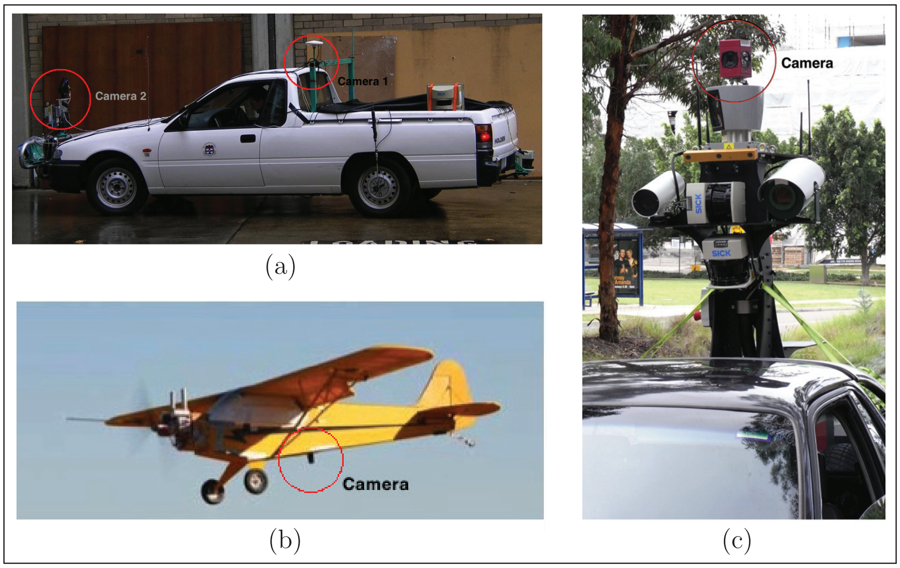


Fig. 8. Vehicles used in the experiments: (a) modified car; (b) unmanned aerial vehicle; (c) robotic platform for generalization testing.

memory requirements and faster training times. For the experiments in algorithm generalization, we used a different ground vehicle composed of a robotic platform mounted on the back of a car (Figure 8(c)). The same methodology was then extended to address 3D environments, using data collected from an unmanned aerial vehicle (UAV; Figure 8(b)), during a flight over a deserted area. The UAV is capable of moving in all six degrees of freedom, providing a test platform for any visual odometry application.

6.1. Ground experiments

For the ground vehicle tests, a conventional car (Figure 5.3) was modified to include two different cameras, a standard

SICK laser sensor and a GPS system with precision of up to 5 m, used here solely for comparison purposes. The camera captured images at a rate of roughly 5 frames per second at a 1,152 × 758 pixel resolution, which were then downsampled to 384 × 252 pixels (one third of the original resolution). The reasons for this downsample are: (1) to verify the robustness of the algorithm in low-resolution cameras (marginally better results can be obtained with higher resolution); (2) to speed up SIFT (or equivalent) feature extraction and matching processes. During data acquisition the car moved at speeds of up to 40 km/h over asphalt and grass, and interacted normally with pedestrians and other vehicles. It is also worth noting that, even though the tests were conducted outdoors, GPS information is unavailable in several areas due mostly to tree coverage and tall buildings.

The training dataset is composed of 2,000 images acquired in an urban environment. Ground-truth information was obtained based on laser data, using the iterative closest point (ICP) algorithm (Lu and Milios, 1994), and the resulting localization estimates are depicted in Figure 9(a). Because they are incremental, these estimates are by themselves subject to drift due to the accumulation of small errors over time. Even though this drift could in principle be greatly reduced by fusing the estimates with an absolute sensor (such as GPS), here we use the ICP results directly as ground-truth information. This is done in order to verify the CGP framework's ability to average over small errors by using a large training dataset to learn the underlying function directly from noisy information, and also to minimize the need for high-precision sensors during the training stage. Empirical tests show marginal improvements in localization when more precise ground-truth is used.

The testing dataset is also composed of 2,000 images, acquired using the same vehicle over a different trajectory of roughly 2 km. Localization results obtained from the same ICP algorithm used on the training dataset are presented in Figure 9(b) for comparison purposes. Similar results obtained using a geometric model (SFM), with automatic camera calibration and manual scale adjustment to account for scale recovery in a monocular configuration, are presented in Figure 10(a). As expected, both approaches are subject to drift caused by error accumulation, especially in rotation due to smaller overlapping areas between frames and higher sensitivity to imprecisions in angular motion.

Figure 10(b) shows the localization results obtained using two independent GPs, one for each task (linear and angular velocities). The first notable aspect of these results is the ability of the GP framework to recover scale up to a high degree of precision, which is a non-trivial task in monocular configurations as shown by the SFM estimates. We attribute this ability to the non-parametric structure of the model, that is capable of exploiting similarities in optical flow distribution during training and extrapolating this information to infer scale directly from a single image. However, the smaller number of training samples representing vehicle rotation, allied with the non-incorporation of cross-correlations between tasks, makes angular inference especially challenging in this approach.

Table 1. Linear and angular errors per frame for each task in ground experiments.

Method	Translational error (rmse) (10^{-2} m)	Rotational error (rmse) (10^{-2} rad)
ICP	2.92 ± 4.70	0.06 ± 0.14
SFM	9.75 ± 12.12	0.23 ± 0.16
Single GPs	5.98 ± 8.67	0.14 ± 0.22
CGPs	5.74 ± 8.18	0.07 ± 0.08
MCGP	5.12 ± 7.49	0.05 ± 0.07

The localization results obtained using the standard CGP framework (without the geometric model) are presented in Figure 10(c), where it is possible to see a substantial reduction in angular drift due to the CGP's ability to correlate between tasks, using linear motion information to further refine its angular estimates. Finally, Figure 2(b) depicts the localization results obtained using the MCGP. The calibration parameters were optimized as hyperparameters with random initial guesses, so no prior knowledge of the visual system was required. Again, scale is recovered up to a high degree of precision, and angular motion errors are even less pronounced. We attribute this improvement to the MCGP's ability to 'fine-tune' the estimates provided by the geometrical constraints, without the need to fully model the underlying phenomenon as it is the case when no geometric model is used.

A quantitative comparison of all methods described in this section is presented in Table 1, in terms of root mean square error (rmse) per frame. The ground-truth for these comparisons was obtained using ICP estimates integrated into a exactly sparse information filter (ESIF) framework (Walter et al., 2007). As expected, ICP has the lowest translational error, because distances can be measured directly from a laser scanner. Even with manual scale adjustment, the SFM approach still shows the highest translational error, and all GP-based approaches to visual odometry performed similarly in the scale recovery aspect. The rotational error, on the other hand, decreased substantially with the introduction of the GP framework over the traditional geometric model approach, and continued to decrease consistently with the incorporation of multi-tasking and the semi-parametric extension. Even though ICP has a rotational error comparable to CGP, its variance shows that this error is not spread evenly throughout the trajectory, being concentrated in only a few frames as shown in Figure 9(b). The CGP framework is able to smooth out these errors and generate more consistent results, without any large localized discrepancies.

6.1.1. Changing environments The same methodology was tested in an off-road environment, composed mostly of trees and grass terrain. The dataset is composed of 4,000

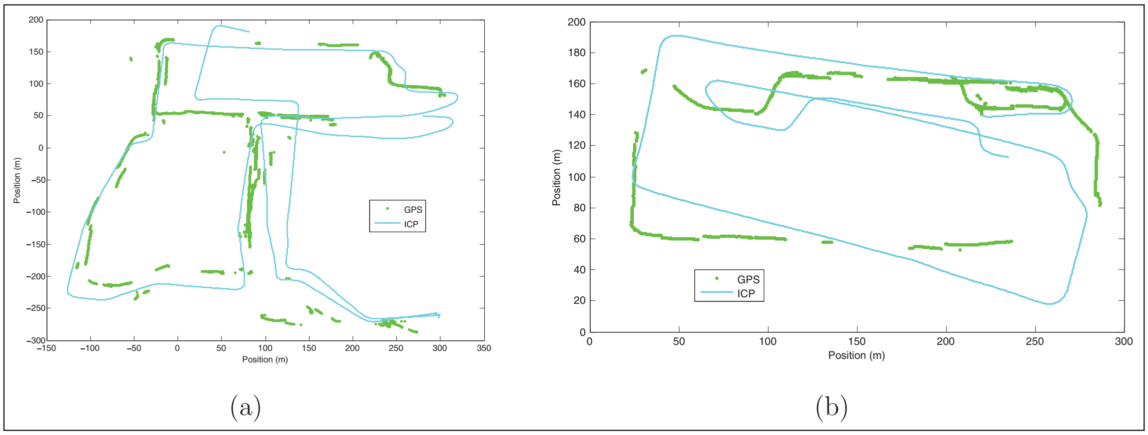


Fig. 9. ICP localization results in ground experiments: (a) training dataset; (b) testing dataset.

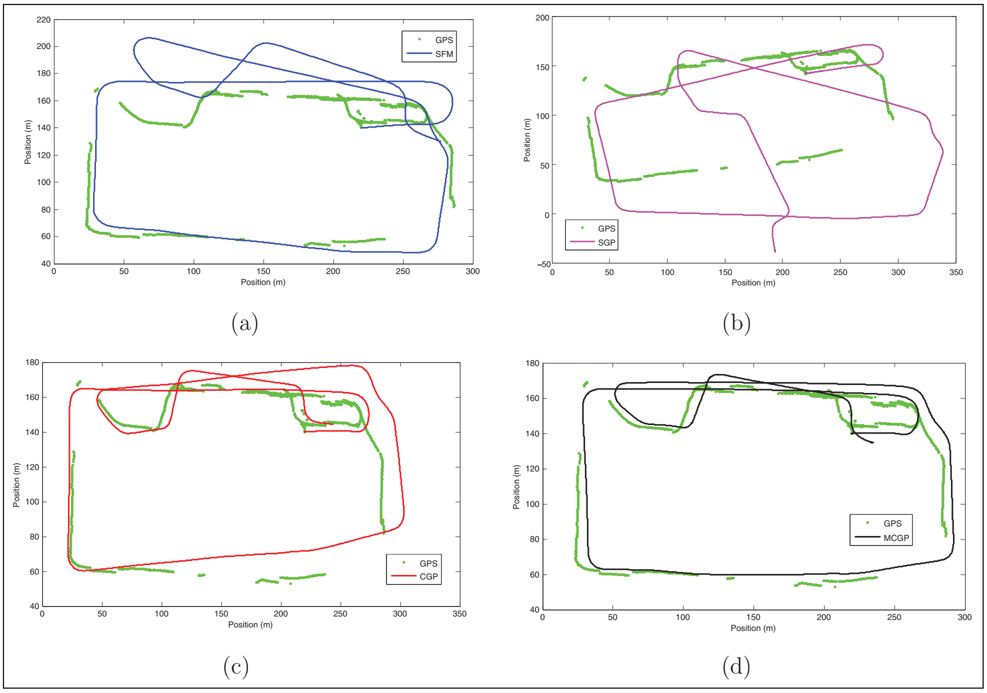


Fig. 10. Localization results in ground experiments using different methods: (a) SFM with manual scale adjustment; (b) two independent single GPs; (c) CGPs; (d) MCGP.

images obtained in a park using the same vehicle over a trajectory of roughly 3 km. The same urban training dataset was used and there was no further training conducted using information from the new environment. Our goal was to verify the MCGP framework's ability to generalize over different environments, exploring similarities in optical flow distribution that are inherent to vehicle motion and are present in any type of structure regardless of its physical nature. The localization results obtained under these conditions are presented in Figure 11, along with image examples from different parts of the trajectory.

In Figure 11 we can see that the SFM algorithm was able to recover the overall shape of the trajectory, however, as expected, there is no scale consistency (left portion of the image) and the algorithm eventually misses a turn (bottom of the image), compromising the final stages of localization. The MCGP framework, on the other hand, uses this information as initial guesses and further refines its estimates using training data, which even though from a different environment contain enough optical flow distribution examples to provide a robust model of vehicle motion. This results in scale recovery up to a high degree of precision (average translational error of $6.57 \pm 9.64 \times 10^{-2}$ m per frame), and by exploring cross-correlations between linear and angular velocities the rotational error not only decreases (average rotational error of $0.08 \pm 0.07 \times 10^{-2}$ rad per frame) but is also more evenly distributed throughout the images.

6.1.2. Changing cameras In addition to changing the environment in which the vehicle navigates, we also explored

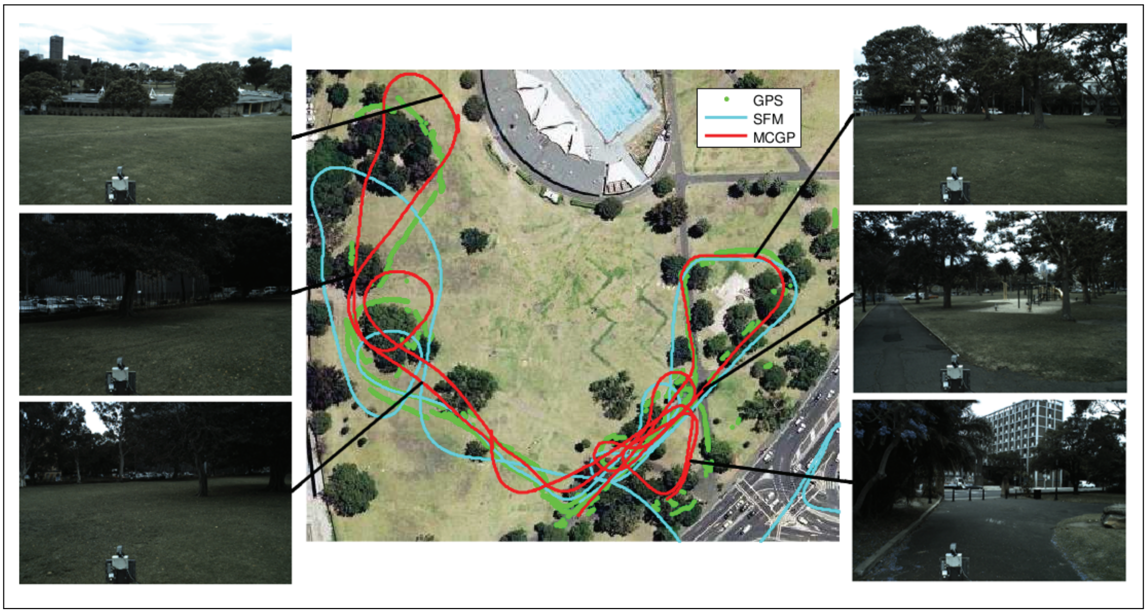


Fig. 11. Localization results in off-road ground experiments.

the effects of changing the camera in which the images are acquired. This was done in order to verify the MCGP framework's ability to deal with variations in camera parameters, as well as variations in optical flow distributions that are not caused by different structures in the environment, but rather by changes in the way camera motion is related to vehicle motion. This new camera has a lower resolution (producing images of 640 × 480 pixels, which were then also downsampled to the same 384 × 252 pixels) and was positioned in such a way that it captures the same portion of the environment as the previous camera, but from a different perspective (see Figure 12).

The same urban training dataset was used without any further training in this new configuration, and the testing dataset was composed of 2,000 images acquired in the same 2 km trajectory as the previous urban testing dataset, but from the new camera's perspective (both cameras acquired their images simultaneously). Figure 13 depicts the localization results obtained using the MCGP framework in both testing datasets. It is clear that the camera exchange had some impact on the quality of the results, most notably in respect to the linear velocity estimates. This is to be expected, since a GP's ability to recover scale in visual odometry from a monocular configuration is dependent on structure similarity between training and testing data. The inference process assumes that the environment reacts in a predictable manner in relation to vehicle motion, and is able to extrapolate scale based on ground-truth information. If the camera changes this assumption is no longer valid, since the environment will now react differently to vehicle motion due to new geometrical constraints that were not modelled during training.

However, the MCGP framework is capable of encoding the effects of camera exchange as an increase on uncertainty, to reflect discrepancies between training and testing data. As the testing data deviates from the training data, the corresponding estimates become less and less accurate, a phenomenon that is captured by an increase in the covariance matrix coefficients. So, even though the estimates obtained using different cameras were less accurate (translational error of $7.87 \pm 9.66 \times 10^{-2}$ m per frame and rotational error of $0.10 \pm 0.14 \times 10^{-2}$ rad per frame), this increase in uncertainty maintains the results equally valid in a probabilistic point of view.

6.1.3. Generalization analysis Here we explore the limits of the proposed algorithm with regards to radical changes between training and testing datasets. The first experiment is conducted using the same urban dataset both for training and testing, however during testing a different number of frames is skipped at each iteration, thus generating unique sets of optical flow distribution that were not learned during training. This configuration also generates smaller overlapping regions, because the vehicle now travels a longer distance between frames. The average errors per frame are shown in Figure 14, for both linear and angular velocities. It is possible to see that the error in linear velocity increases linearly with the number of skipped frames (and, thus, the algorithm's ability to recover scale decreases, because training and testing optical flow distributions are increasingly different). The error in angular velocity also increases monotonically with the number of skipped frames, however the rate in which this error increases has a significant jump between two and three skipped frames (we attribute this to the increasingly smaller overlapping regions between frames as more frames are skipped). When four frames are skipped at every iteration (only one for every five are used) the errors in linear velocity are around 20 times higher than in the original configuration, and the errors in angular velocities are around 15 times higher, an indication that

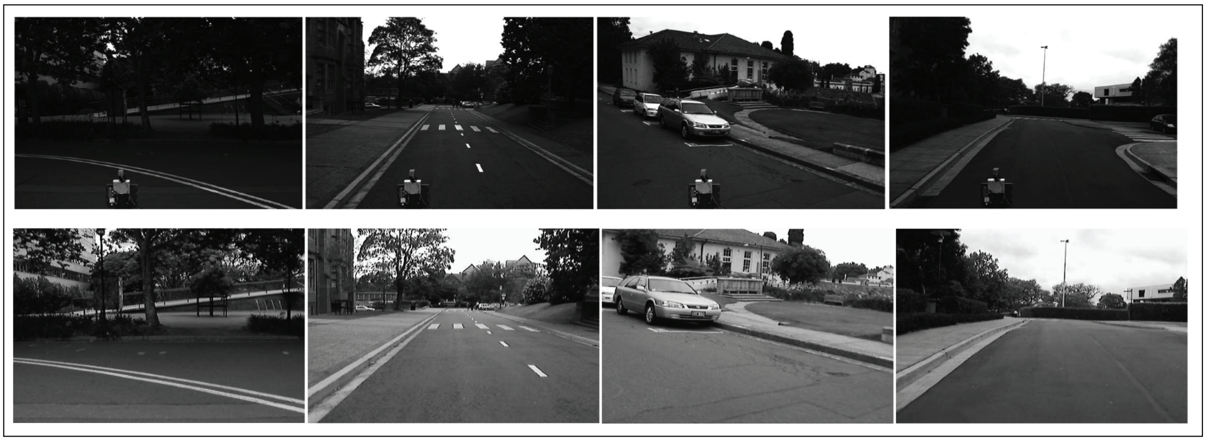


Fig. 12. Examples of images taken at the same vehicle position with different cameras. The first line corresponds to the original camera and the bottom line corresponds to the new camera (the new camera can be seen in the images captured by the original camera). The displacement between cameras is approximately 2 m horizontally and 0.5 m vertically.

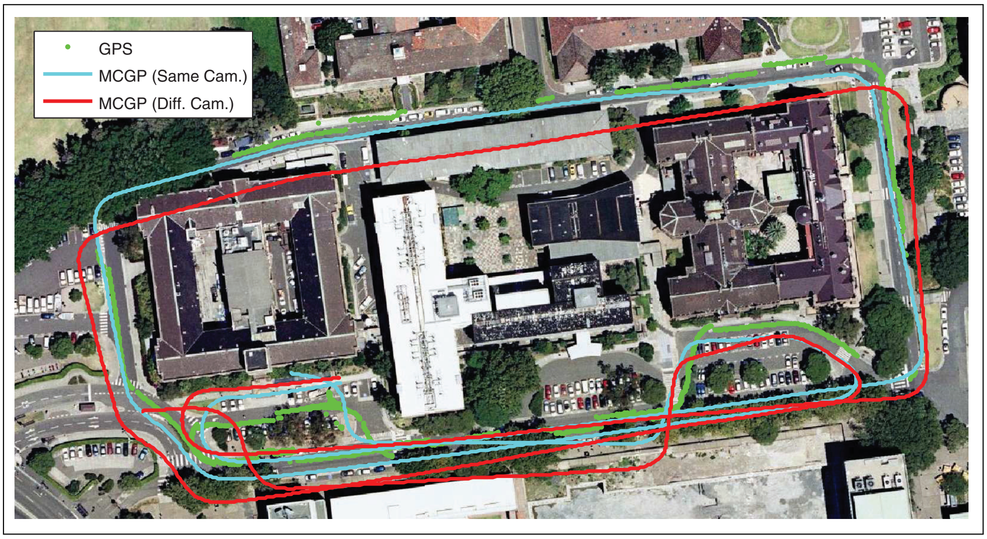


Fig. 13. Localization results in ground experiments using different cameras.

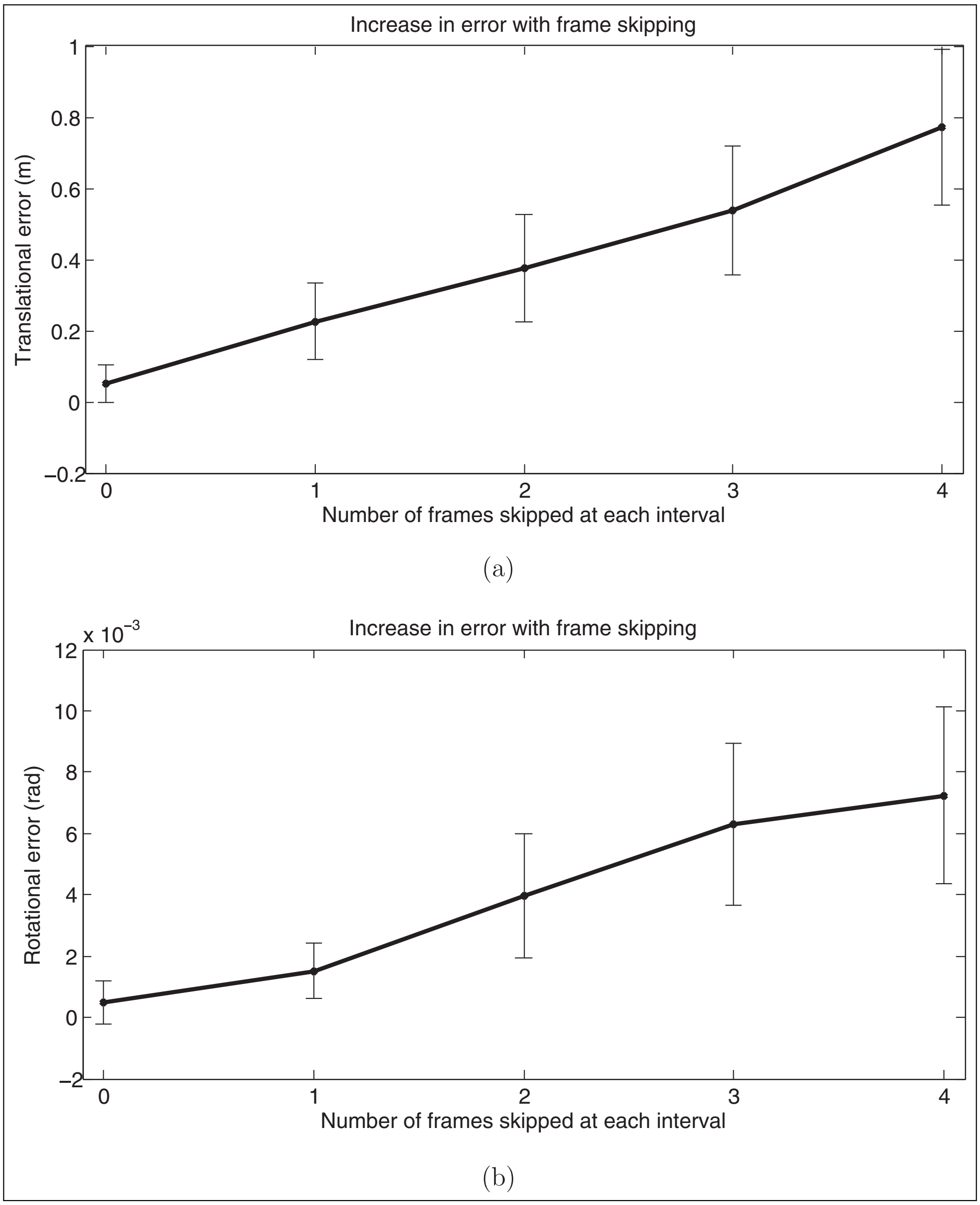
the algorithm has failed and no more useful localization information can be obtained.

The second experiment is conducted using a different robotic platform (Figure 8(b)), with different camera and motion dynamics. This new setup was used to collect data on a highly dynamic environment, composed of 14,500 images and their corresponding ground-truth information acquired from a fusion of GPS and inertial sensors. Results obtained using the proposed method on this new dataset are presented on Figure 15, where it is possible to see that MCGP was able to achieve localization up to a satisfactory degree of precision (linear and angular velocity errors were on par with those calculated so far). The next step was to use this same dataset for training and test the resulting model on the urban dataset mentioned previously.

It is natural to assume that under these conditions the proposed algorithm will not perform adequately, since the optical flow distributions available for training will differ radically from those presented during testing, due to discrepancies in the camera's intrinsic and extrinsic parameters and also due to changes in vehicle dynamics. Here we aim to evaluate how this performance decreases as less and less similarities between training and testing data are present.

The results are depicted in Figure 16, and again it is possible to see that the error in linear velocity increases linearly with the percentage of different optical flow distributions on the training dataset, whereas the error in angular velocity initially increases slowly and afterwards has a significant jump showing where the algorithm starts to fail (at approximately 50% of different optical flow distributions). This similarity between results in both experiments is to be expected, since they are all in essence dealing with the same scenario: a different set of optical flow distributions between training and testing datasets.

6.1.4. Extension to SLAM Up to this moment all vehicle motion estimates were obtained independently, based on information obtained from a single image pair. However, since the CGP framework allows the assessment of



(a)

(b)

Fig. 14. Average (a) translation and (b) rotational errors per frame according to the number of frames skipped at each interval (using MCGP and the same urban dataset for training and testing).

a full covariance matrix of uncertainties, we explore here extensions to the SLAM scenario as an attempt to improve results. More specifically, we use an ESIF (Walter et al., 2007) to track all vehicle poses during navigation and a loop-closure algorithm to recognize visited areas, using this information to retroactively decrease global uncertainty. The loop-closure process is done by matching features from the current frame with features from previous frames (downsampled by a factor of five for speed purposes, without impacting results), and an area is assumed revisited if the number of successful matches is higher than a given threshold.

The localization results obtained using this approach in both urban and off-road testing datasets are depicted in Figure 17. In the urban dataset it is possible to see how the loop-closure algorithm was capable of recognizing the second pass over the west street, aligning the trajectory and using this information to correct misalignments on the other streets. During the third pass on the west street, the vehicle was facing the opposite direction, so it was unable to match any images, resulting in residual misalignment in this area. The vehicle was also capable of recognizing its return to the starting point (upper left portion of the image). The off-road dataset presented a more challenging scenario for the ESIF framework, because the more complex trajectory and navigation in opposite directions complicated the loop-closure process. Still, the vehicle was capable of recognizing its return to the initial street and correct most of the misalignments that occurred on the left portion of the image.

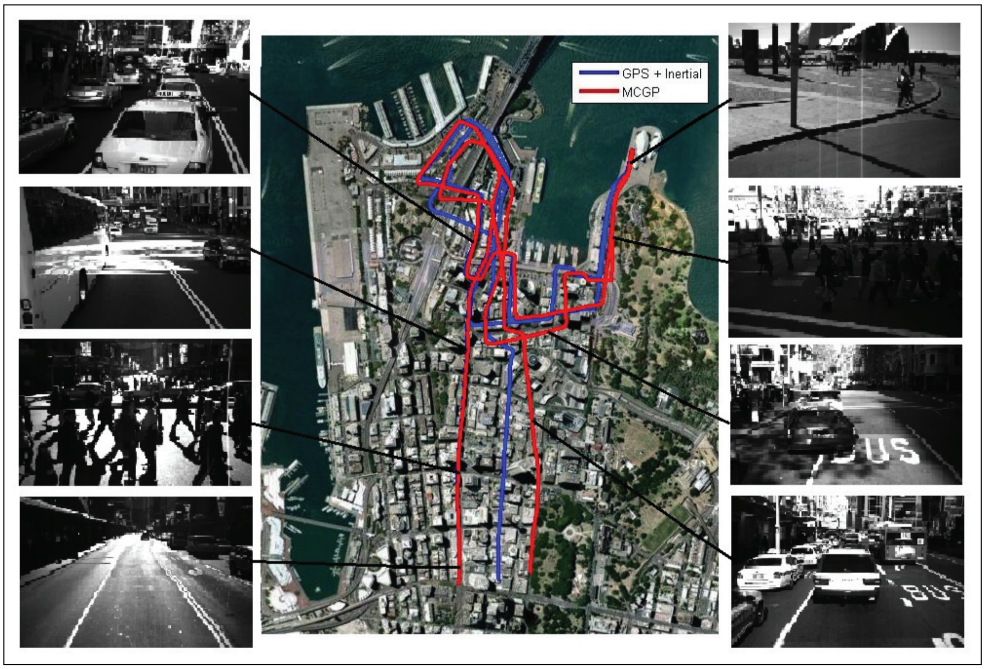


Fig. 15. Localization results obtained using the proposed method and a different robotic platform (training and testing conducted in the same conditions), along with sample images obtained during navigation.

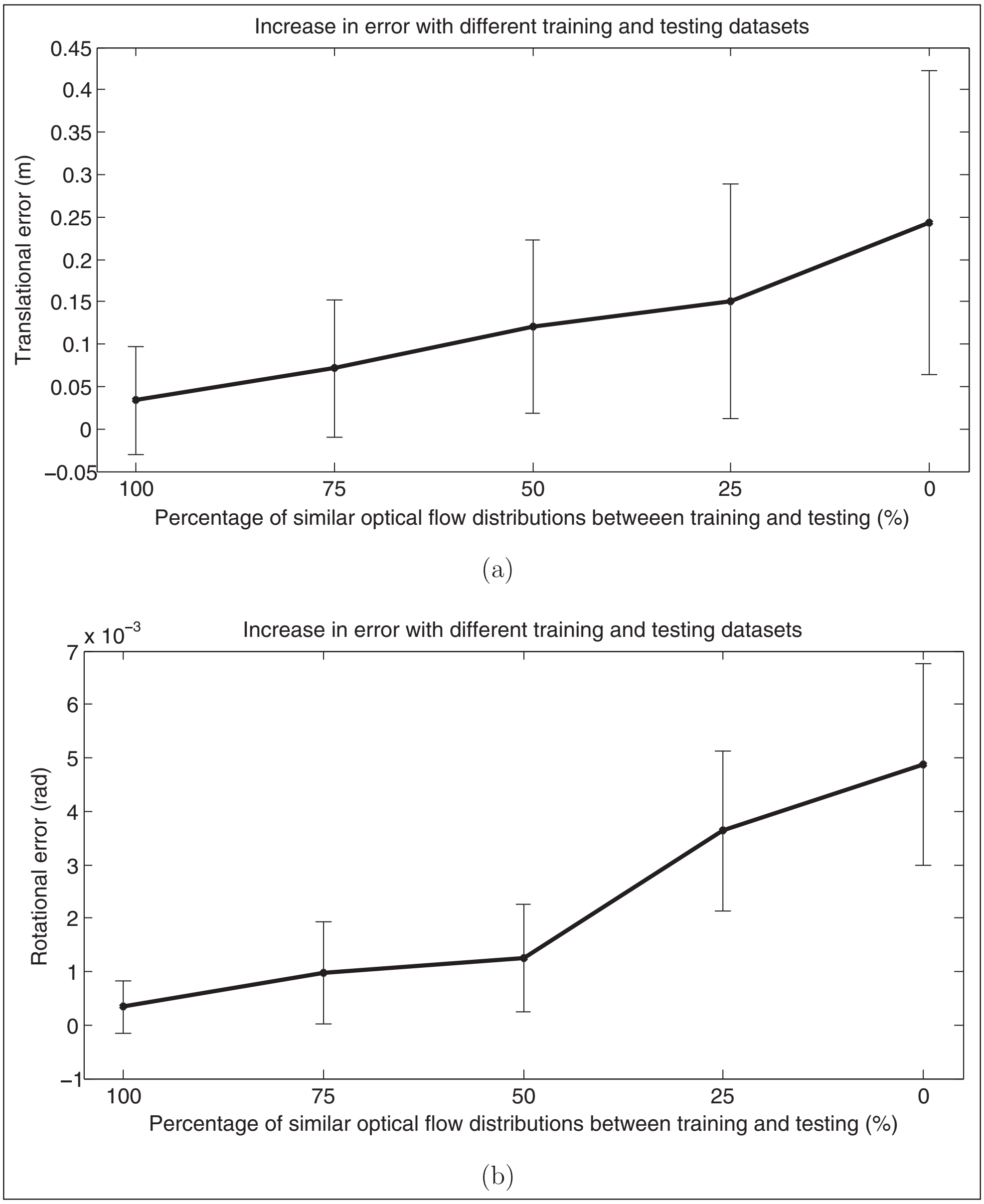
Table 2. Linear and angular errors per frame for each task in ground experiments.

Scenario	Translational error (rmse) (10^{-2} m)	Rotational error (rmse) (10^{-2} rad)
Same environment	5.12 ± 7.49	0.05 ± 0.07
Different environment	6.57 ± 9.64	0.08 ± 0.07
Different camera	7.87 ± 9.66	0.10 ± 0.14
Extension to SLAM	5.98 ± 6.54	0.04 ± 0.07

6.2. *Aerial experiments*

The visual odometry algorithm proposed in this paper was also tested using data collected from a UAV (Figure 5.3) flight over a deserted area, at a rate of 3 frames per second and an average speed of 110 km/h. The UAV was also equipped with inertial and GPS sensors that served as ground-truth information. The first 4,000 frames after aircraft stabilization were used for training, and the 2,000 following frames were used for evaluation (maintaining altitudes of 80–100 m). The SIFT algorithm failed to find any matches in around 2% of the image pairs, due to a lack of overlapping areas caused by severe angular motion. These frames were avoided during training, and during evaluation the results from the previous timestep were repeated. It is important to note that, even though constrained to forward motion, the UAV was capable of experiencing motion in all six degrees of freedom (linear velocities on the x-, y- and z-axis and angular velocities $\dot{\gamma}$, $\dot{\beta}$ and $\dot{\alpha}$ in Euler angles) due to air resistance and draft, providing a test platform for any visual odometry application. A quantitative error estimation for each one of these degrees of freedom is presented in Table 3, along with comparisons with other techniques.

Figure 18 presents the localization results obtained using only a calibrated camera model (SFM) and the MCGP framework (with first-order temporal dependency between frames). The flight trajectory was mostly horizontal, and Figure 18(a) shows that the MCGP approach was capable of recovering its overall shape, with no missing corners or changes in the plane of navigation. The absolute scale was also recovered to a high degree of precision, estimated from the training data and extrapolated to address new points in the input space. Significant changes in altitude to areas with no training data would compromise scale recovery, as this would change the correlation between image structure and vehicle motion. As expected, a combination of accumulated errors and lack of matching features generated a drift over time that could not be avoided, however the MCGP framework was able to improve significantly the results obtained using only the geometric model. In Figure 18(b) it is possible to see the cyclical changes in altitude during flight, ranging from 80 to 100 m. The high frequency of these changes constitute a challenge for the GP as a regression tool, due to the difficulty in separating what is a trend and should be modelled and what is noise and should be discarded. Interestingly, the use of temporal dependencies between tasks created a 'smooth and delay' effect as a response to sudden variations, because of the proximity constraint imposed to outputs in subsequent steps.



(a)

(b)

Fig. 16. Average (a) translational and (b) rotational errors per frame according to the percentage of similar optical flow distributions between training and testing datasets (using MCGP and different robotic platforms).

Table 3. Root mean square linear (10^{-2} m) and angular (10^{-2} rad) errors per frame for each task in aerial experiments.

Task	SFM	Single GPs	CGPs	MCGP
X	1384.10 ± 25.72	20.47 ± 0.1552	8.49 ± 0.0668	8.11 ± 0.0727
Y	453.56 ± 5.76	6.84 ± 0.0541	5.95 ± 0.0472	5.71 ± 0.0269
Z	325.50 ± 6.69	10.16 ± 0.0806	10.23 ± 0.0812	9.89 ± 0.0714
Roll	11.48 ± 0.56	0.69 ± 0.0056	0.66 ± 0.0053	0.47 ± 0.0051
Pitch	5.09 ± 0.01	0.35 ± 0.0027	0.26 ± 0.0021	0.18 ± 0.0025
Yaw	19.07 ± 0.55	0.41 ± 0.0032	0.33 ± 0.0027	0.25 ± 0.0021

7. Conclusion and future work

This paper presented a novel technique for visual odometry based on machine learning concepts. A novel multi-task GP inference method was proposed as a way to provide full covariance matrices for motion estimates, calculating both auto- and cross-dependencies between tasks. By learning the hyperparameters through a Bayesian framework it

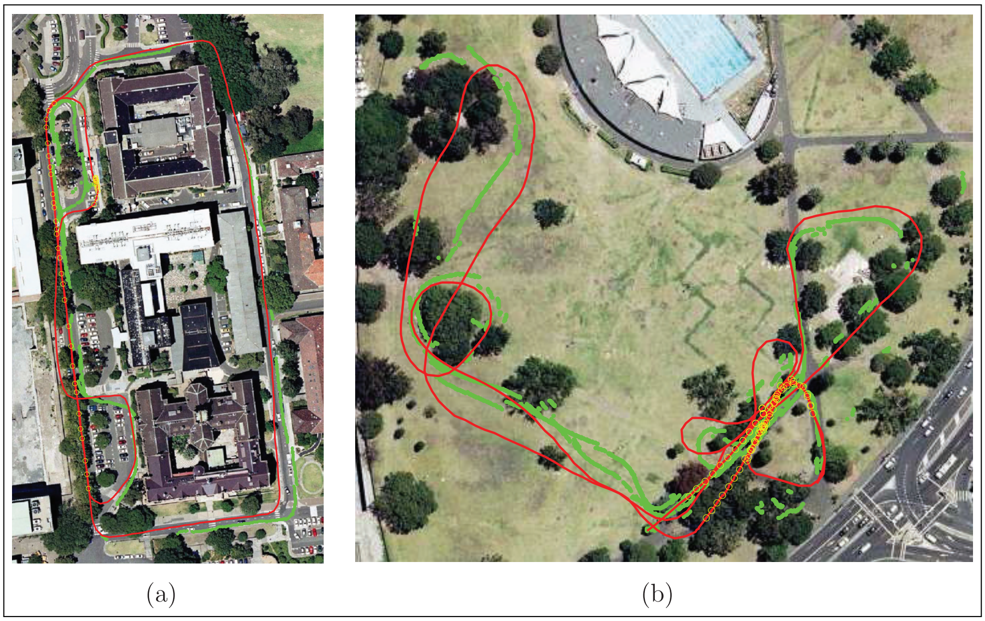


Fig. 17. Localization results obtained using the MCGP results combined with ESIF (green dots are GPS information, red lines are the localization results and yellow circles are loop-closures; color refers to the online version of this article): (a) urban dataset; (b) off-road dataset.

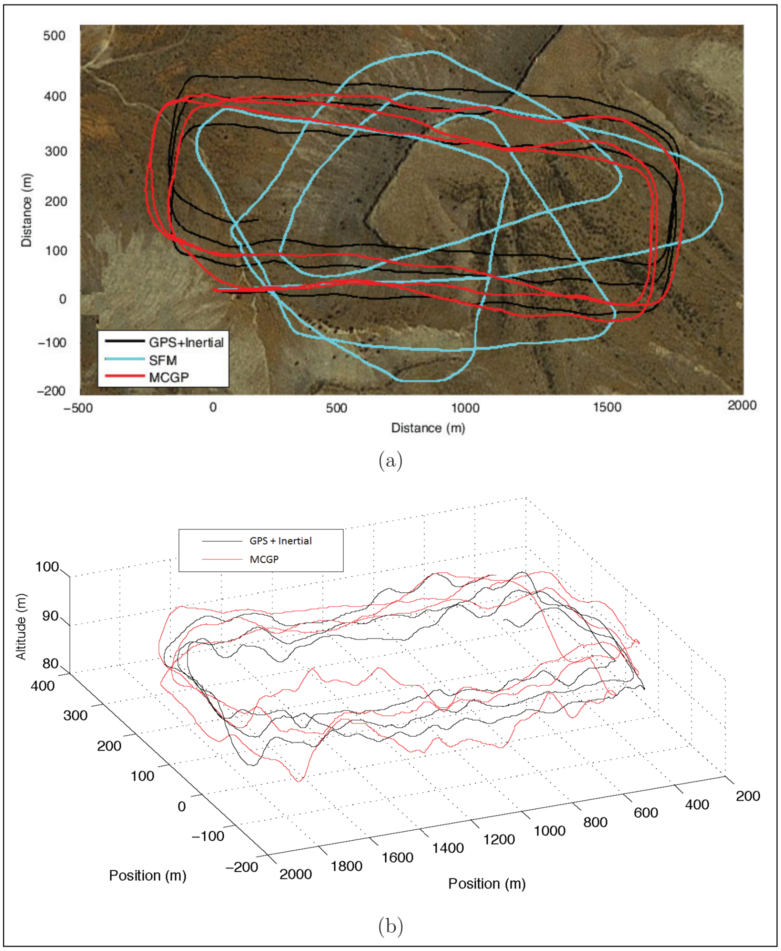


Fig. 18. Localization results obtained in aerial experiments: (a) 2D plot (top view); (b) 3D plot.

is possible to (indirectly) model the underlying characteristics of a camera, thus avoiding the calibration method necessary in other approaches. This inference method was also extended to include first-order temporal dependencies between tasks, and the traditional zero mean assumption in GP implementations was substituted by a geometric model capable of providing initial estimates that were then further refined using training data. The simultaneous optimization of both the calibration parameters and the hyperparameters eliminates the need for prior calibration of the visual system in this semi-parametric approach, and if this information is available it can be incorporated seamlessly as initial guesses. This methodology is capable of recovering scale in a monocular configuration, provided that training and testing data share a certain similarity in optical flow distribution, and the estimation of uncertainties allow the use of results in filtering and SLAM frameworks. Even though the training process may take up to a few hours, depending on the number of tasks, new inferences can be computed at a rate of 10 Hz, and thus are suitable for real-time applications. Tests were conducted in both 2D and 3D environments, using data collected from a modified car and an unmanned aerial vehicle, and the results show a significant improvement over standard visual odometry algorithms. Future work will focus on online learning, where the vehicle uses information obtained during navigation to refine its own model while eliminating redundant data points to keep the computational costs constant. The use of optical flow information for automatic detection of dynamic objects will also be explored, as means to improve the visual feature sets used during training and testing.

Notes

1. In principle, each particular training dataset may be composed of a different set of observations. However, since in the visual odometry scenario this is generally not the case (each training image has a corresponding ground-truth estimation for all degrees of freedom) we will assume from now on that all training datasets are composed of the same set of observations.
2. Dense optical flow extraction methods, such as that of Lucas and Kanade (1981), were tested and discarded due to the number of parameters to be manually determined, and also due to the large variability in performance in different environment and driving conditions.

Funding

This research was supported under the Australian Research Council's Discovery Early Career Research Award funding scheme (project number DE120103051).

References

Agrawal M and Konolige K (2007) Rough terrain visual odometry. In: *Proceedings of the International Conference on Advanced Robotics (ICAR)*.

Bay H, Tuytelaars T and Gool LV (2006) SURF: Speeded up robust features. In: *Proceedings of the ECCV*, pp. 404–417.

Bonilla EV, Chai KM and Williams CKI (2008) Multi-task gaussian process prediction. In: Platt JC, Koller D, Singer Y and Roweis S (eds.), *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, pp. 153–160.

Botelho S, Drews P, Oliveira G and Figueiredo M (2009) Visual odometry and mapping for underwater autonomous vehicles. In: *Proceedings of the 6th Latin American Robotics Symposium (LARS)*.

Boyle P and Frean M (2005) *Multiple Output Gaussian Process Regression*. Technical report, University of Wellington.

Campbell J, Sukthankar R and Nourbakhsh I (2004) Techniques for evaluating optical flow for visual odometry in extreme terrain. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

Chai KM, Klanke S, Williams C and Vijayakumar S (2008) Multi-task Gaussian process learning of robot inverse dynamics. In: *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Cheng Y, Maimone M and Matthies L (2005) Visual odometry on the Mars Exploration Rovers. In: *Proceedings of the International Conference on Systems, Man and Cybernetics*.

Corke P, Detweiler C, Dunbabin M, Hamilton M, Rus D and Vasilescu I (2007) Experiments with underwater robot localization and tracking. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Corke P, Strelow D and Singh S (2004) Omnidirectional visual odometry for a planetary rover.In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

Cressie N (1993) *Statistics for Spatial Data*. New York: Wiley.

Davison AJ (2003) Real-time simultaneous localisation and mapping with a single camera. In: *Proceedings of the 9th International Conference on Computer Vision (ICCV)*.

Dellaert F (2002) *The Expectation Maximization Algorithm*. Technical report, Georgia Institute of Technology.

Fischler MA and Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24: 381–395.

Guizilini V and Ramos F (2010) Multi-task learning of visual odometry estimators. In: *12th International Symposium on Experimental Robotics (ISER)*.

Guizilini V and Ramos F (2012) Semi-parametric models for visual odometry. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Hartley RI and Zisserman A (2004) *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press.

Hastie T, Tibshirani R and Friedman J (2001) *The Elements of Statistical Learning* (*Springer Series in Statistics*). New York: Springer.

Higdon D (2002) Space and space-time modeling using process convolutions. In: *Quantitative Methods for Current Environmental Issues*. New York: Springer, pp. 37–54.

Howard A (2008) Real-time stereo visual odometry for autonomous ground vehicles. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

Huang AS, Bachrach A, Henry P et al. (2011) Visual odometry and mapping for autonomous flight using an RGB-D camera. In: *15th International Symposium on Robotics Research (ISRR)*.

Kelly J, Saripalli S and Sukhatme G (2007) Combined visual and inertial navigation for an unmanned aerial vehicle. In: *Proceedings of the 6th International Conference on Field and Service Robotics*.

Kelly J and Sukhatme G (2007) An experimental study of aerial stereovisual odometry. In: *Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles*.

Lemaire T, Berger C, Jung IK and Lacroix S (2007) Vision-based SLAM: Stereo and monocular approaches. *International Journal of Computer Vision* 74: 343–364.

Lovegrove S, Davison AJ and Guzman JI (2011) Accurate visual odometry from a rear parking camera. In: *Proceedings of the Intelligent Vehicles Symposium*.

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60: 91–110.

Lu F and Milios E (1994) Robot pose estimation in unknown environments by matching 2D range scans. In: *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*.

Lucas BD and Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: *Proceedings of the DARPA Image Understanding Workshop*.

MacKsay DJC (2002) *Information Theory, Inference and Learning Algorithms*. Cambridge: Cambridge University Press.

Moravec H and Gennery DB (1976) *Cart Project Progress Report*. Technical report, Stanford University.

Moravec HP (1980) *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Ph.D. thesis, Stanford University.

Neal RM (1996) *Bayesian Learning for Neural Networks*. New York: Springer-Verlag.

Nister D, Naroditsky O and Bergen J (2006) Visual odometry for ground vehicle applications. *Journal of Field Robotics* 23: 3–20.

O'Callaghan S, Ramos F and Durrant-Whyte H (2009) Contextual occupancy maps using Gaussian processes. In: *Proceedings of the International Conference on Robotics and Automation*.

Rasmussen CE and Williams KI (2006) *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press.

Roberts R, Nguyen H, Krishnamurthi N and Balch T (2008) Memory-based learning for visual odometry. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Roberts R, Potthast C and Dellaert F (2009) Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Scaramuzza D, Fraundorfer F, Pollefeys M and Siegwart R (2009) Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In: *Proceedings of the International Conference on Computer Vision (ICCV)* 24: 1015–1026.

Scaramuzza D and Siegwart R (2008) Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*.

Se S, Lowe D and Little J (2001) Vision-based mobile robot localization and mapping using scale-invariant features. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*, pp. 2051–2058.

Sunderhauf N, Konolige K, Lacroix S and Protzel P (2005) Tagungsband Autonome Mobile Systeme. In: *Visual Odometry using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle*. New York: Springer-Verlag.

Tardif JP, Pavlidis Y and Daniilidis K (2008) Monocular visual odometry in urban environments using an omnidirectional camera. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pp. 2531–2538.

Tipping ME and Bishop CM (1999) Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B* 61: 611–622.

Tomasi C and Zhang J (1995) Is structure-from-motion worth pursuing? In: *Proc. 7th International Symposium on Robotics Research (ISRR)*, pp. 391–400.

Tomasi S and Tomasi C (1994) Good features to track. In: *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*.

Vasudevan S, Ramos F, Nettleton E and Durrant-Whyte H (2009) Gaussian process modeling of large scale terrain. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Walter MR, Eustice RM and Leonard JJ (2007) Exactly sparse extended information filters for feature-based SLAM. *International Journal of Robotics Research* 26: 335–359.

Williams CKI (1998) Computation with infinite neural networks. *Neural Computation* 10: 1203–1216.

Zhu ZW, Oskiper T, Naroditsky O, Samarasekera S, Sawhney HS and Kumar R (2006) An improved stereo-based visual odometry system. In: *Proceedings of the Workshop of Performance Metrics for Intelligent Systems*.