2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Visual Odometry Learning for Unmanned Aerial Vehicles

Vitor Guizilini and Fabio Ramos
Australian Centre for Field Robotics, School of Information Technologies
University of Sydney, Australia
{v.guizilini;f.ramos}@acfr.usyd.edu.au

Abstract— **This paper addresses the problem of using visual information to estimate vehicle motion (a.k.a. visual odometry) from a machine learning perspective. The vast majority of current visual odometry algorithms are heavily based on geometry, using a calibrated camera model to recover relative translation (up to scale) and rotation by tracking image features over time. Our method eliminates the need for a parametric model by jointly learning how image structure and vehicle dynamics affect camera motion. This is achieved with a Gaussian Process extension, called Coupled GP, which is trained in a supervised manner to infer the underlying function mapping optical flow to relative translation and rotation. Matched image features parameters are used as inputs and linear and angular velocities are the outputs in our non-linear multi-task regression problem. We show here that it is possible, using a single uncalibrated camera and establishing a first-order temporal dependency between frames, to jointly estimate not only a full 6 DoF motion (along with a full covariance matrix) but also relative scale, a non-trivial problem in monocular configurations. Experiments were performed with imagery collected with an unmanned aerial vehicle (UAV) flying over a deserted area at speeds of 100-120 km/h and altitudes of 80-100 m, a scenario that constitutes a challenge for traditional visual odometry estimators.**

I. 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 providing localization estimates: internal and external. Internal sensors include encoders and IMUs, which work isolated from the outside world and can perform well in small-scale experiments, but tend to drift over time due to error accumulation. External sensors include GPS, range-finders and cameras, which interact in one way or another with the environment around the vehicle and can provide both incremental and absolute localization estimates.

Of all external sensors, cameras are cheap, compact, low power, 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 can provide predictions comparable in accuracy to most commercial IMUs [1]. Also, recent increases in computational power allow real-time visual motion estimation on standard processors, which can be further integrated with other sensors for improved predictions.

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, as to establish correspondences between features in overlapping areas. If the environment is assumed static, any optical flow detected on the images will be caused by the camera's own motion that can be used to infer relative rotation and translation between frames. Most visual odometry algorithms address this problem geometrically [2], using a calibrated camera model to minimize the reprojection of 3D points triangulated from matched features.

We propose an alternative approach, where a Gaussian Process [3] is used as a regression tool to learn the underlying function that maps optical flow information directly into camera motion. Training data is obtained from an independent sensor, and a likelihood function is optimized to fit this data, where a covariance function quantifies the relationship between inputs. 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 threefold: First of all, it eliminates the need for a geometrical model, or even calibration parameters since vehicle motion is inferred directly from image features. Secondly, it naturally provides a full covariance matrix for all outputs, thus allowing posterior data fusion with other sensors. Lastly, the resulting model is capable of estimating the relative scale by exploring and learning dependencies in the image structure, a non-trivial task in monocular configurations. This approach is tested in a challenging dataset where imagery is obtained from an aerial platform flying at speeds of 100-120 km/h and altitudes of 80-100 m. The trajectory extends for around 20 km and is characterised by small and inconsistent overlapping regions between two consecutive images, lenses with very narrow field of view and significant changes in illumination.

The remainder of this paper is organized as follows: Section II provides a brief overview of visual odometry algorithms and multi-task learning methods. In section III we describe the mechanism used for optical flow extraction and parametrization. Section IV recapitulates the principles and fundamental equations behind Gaussian Processes, moves on to Coupled GPs and then introduces the temporal dependency used to increase the amount of information available for training and inference. In Section V we present and discuss results obtained using the proposed methodology, providing comparisons with a standard structure-from-motion algorithm. Finally, Section VI concludes the paper and discusses future research directions.

978-1-61284-385-8/11/$26.00 ©2011 IEEE

II. RELATED WORK

Visual information has become a viable and competitive approach for pose estimation. It has been implemented successfully in applications such as unmanned aerial vehicles [4], underwater robots [5], space exploration [6] and indoor/outdoor ground terrains [7], [1], [8]. Several modifications to the original scheme [9] have been proposed in an attempt to improve both quality and applicability of solutions: the use of omnidirectional cameras [10], [11], robust feature extraction and matching [12], [13], data fusion with other sensors [14], [7] and extension to a Simultaneous Localization and Mapping (SLAM) framework [15], [16], [17].

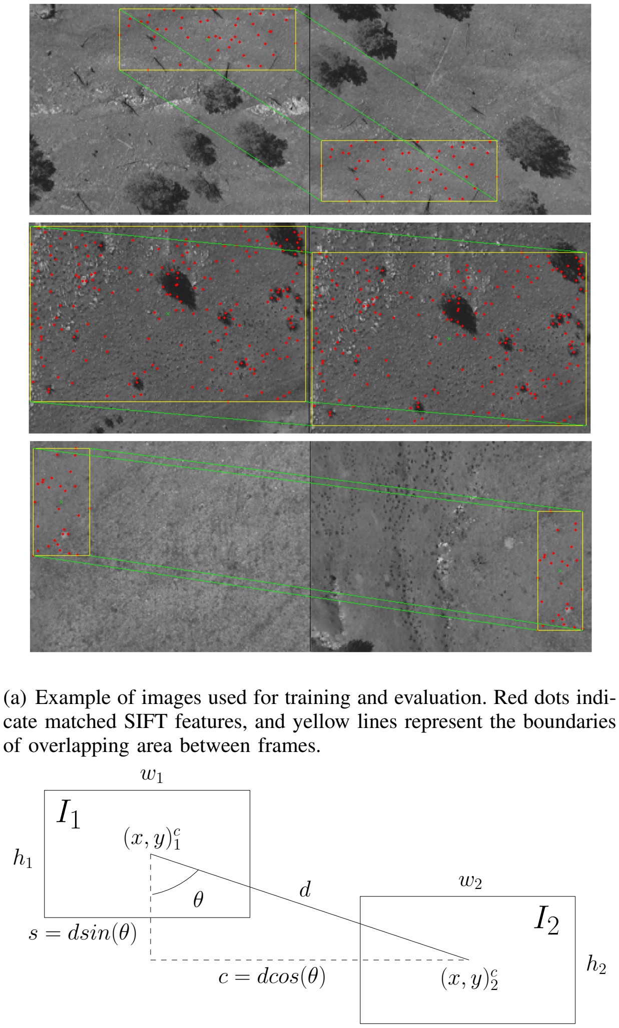
Visual odometry algorithms can be broadly divided into two categories: stereo and monocular configurations. Stereo configurations [1] use a multi-camera array (or a moving camera) to capture several images simultaneously, from different vantage points. If the baseline is known, it is possible to project detected features into the 3D space, and by tracking them over time, to estimate vehicle motion. Monocular configurations use a single camera, which is essentially a bearing-only sensor. If a sequence of images taken at different locations is provided, the baseline (here equivalent to camera translation and rotation) can be estimated, a scenario commonly known as structure-from-motion [18]. One well-known limitation of monocular vision odometry is the inability to recover absolute scale, a problem addressed in [19] for the special case of nonholonomic constraints.

Although intuitive, the machine learning framework has been scarcely used in visual odometry [20], [21]. Self-calibration methods [22] are widespread, however they still assume a known fixed camera model, and there is no guarantee that these parameters will not change over time due to vibration, mechanical stress or changes in temperature. Machine learning algorithms, on the other hand, are capable of inferring both camera model and calibration parameters, and any posterior changes on the underlying function will reflect on the uncertainty estimates. A Gaussian Process (GP) [3] is a non-parametric regression and classification tool which has been used with great success in various areas of mobile robotics such as mapping [23], terrain modeling [24] and dynamic system learning [25].

The standard GP derivation assumes a single output variable, using independent models to deal with multiple outputs when necessary. This is however not the case in visual odometry, which is essentially a multi-task problem where each output is heavily correlated due to vehicle motion constraints. Alternative derivations [26] compute a single covariance matrix containing observations from all tasks, but each inference is still conducted independently. In [27] the authors propose a method where the full covariance matrix is estimated. This method is extended here to address the full problem of 3D visual odometry with 6 degrees of freedom, and the introduction of a first-order temporal dependency between frames creates a robust and generic framework that can address more challenging scenarios.

III. OPTICAL FLOW PARAMETRIZATION

The images used in this paper were obtained from a camera installed in an aircraft pointing downwards. In this configuration, the high altitude 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 (see Fig. 1(a)). These overlapping areas are defined here as the smallest rectangle that includes all successfully matched features from one frame in relation to another.



(a) Example of images used for training and evaluation. Red dots indicate matched SIFT features, and yellow lines represent the boundaries of overlapping area between frames.

(b) Optical flow parameters extracted from two subsequent frames I_1 and I_2.

Fig. 1. Optical flow parametrization.

The SIFT (Scale Invariant Feature Transform) descriptor [28] has been shown to provide robust results under these conditions, and so was chosen here as the feature selection method. Other algorithms such as SURF (Speeded Up Robust Features) [29] could also be readily applied for speed purposes or to increase the amount of information obtained from each frame. A simple filter was implemented to remove outliers. Matched features with a shift distance and angle significantly different from their neighbours are discarded.

Assuming that the aircraft will maintain a considerable altitude and move roughly horizontally, it is reasonable to consider the ground plane as homogeneous. The optical flow information can then be encoded by a single pair of parameters, the average shift distance d and angle θ of all matched features (or their projections c and s on the x-y image axis). Also, the position $(x, y)_i^c$ and sizes $(h, w)_i$ of the overlapping areas are directly related to camera movement, and so contain information that could be useful in the inference process. These eight parameters are illustrated in Fig. 1(b), and together compose the vector

$$\mathbf{x} = \{d, \theta, s, c, x_1^c, y_1^c, x_2^c, y_2^c, h_1, w_1, h_2, w_2\} \tag{1}$$

that will be used as input for the Gaussian Process framework described in the following section.

IV. MOTION ESTIMATION

From the machine learning perspective, the estimation of vehicle motion from sensor information can be seen as a supervised regression problem: the mapping of an input $\mathbf{x}$ to outputs $\mathbf{f}(\mathbf{x}) + \epsilon$ using the training data Λ, where $\epsilon = \mathcal{N}(\mathbf{0}, \Psi)$ represents a Gaussian noise with covariance Ψ. Assuming independent noise, Σ is a diagonal matrix. For visual odometry, $\mathbf{x}_n \in \Re^D$ contains optical flow information extracted from two subsequent frames and $y_n \in \Re$ is a particular corresponding camera motion, obtained for training purposes from an independent sensor. A positive-definite kernel (from here on known as the covariance function) $k(\mathbf{x}, \mathbf{x}')$ is used to characterize the relationship between each two points in the input space, and its coefficients (the hyperparameters) are optimized as to minimize a certain cost function.

A. Gaussian Processes Overview

A Gaussian Process is a non-parametric Bayesian inference method that maintains a probabilistic distribution over an infinite number of functions. From a training dataset $\Lambda = \{\mathbf{x}, y\}_{n=1}^N$ it learns a model that represents the underlying phenomenon. This model is entirely defined by a mean $m(\mathbf{x})$ and covariance $k(\mathbf{x}, \mathbf{x}')$ functions:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2}$$

Assuming a constant zero mean function, inference for a single test point $\mathbf{x}_*$ given Λ involves the computation of the mean $\overline{f}(\mathbf{x}_*) = \overline{f}_*$ and variance $\mathcal{V}(f_*)$, calculated as

$$\overline{f}_* = k(\mathbf{x}_*, X)^T [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y} \tag{3}$$

and

$$\begin{aligned}\mathcal{V}(f_*) =& k(\mathbf{x}_*, \mathbf{x}_*) - \\ &- k(\mathbf{x}_*, X)[K(X, X) + \sigma_n^2 I]^{-1} k(\mathbf{x}_*, X),\end{aligned} \tag{4}$$

where 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 $k(\mathbf{x}, \mathbf{x}')$. The neural network covariance function, as described in [30], is used here due to its non-stationary properties and ability to model sharp transitions and non-linearities:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \arcsin\left(\frac{2\widetilde{\mathbf{x}}^T \Sigma \widetilde{\mathbf{x}'}}{\sqrt{(1 + 2\widetilde{\mathbf{x}}^T \Sigma \widetilde{\mathbf{x}})(1 + 2\widetilde{\mathbf{x}'}^T \Sigma \widetilde{\mathbf{x}'})}}\right). \tag{5}$$

In Eq. 5, σ_f is a signal variance used to scale the correlation between points and $\widetilde{\mathbf{x}} = \{1, \mathbf{x}\}$ is an augmented vector. This derivation comes from a neural network with a single hidden layer, a bias term and a number of units tending towards infinite [31]. The hidden weights are assumed to have a zero mean and a covariance Σ, composed of length-scales coefficients. The variance signal σ_f and the length-scales in Σ are the hyperparameters of the covariance function, and will be optimized during the training process.

B. Coupled GPs

In the 3D space, six parameters are necessary to uniquely describe a vehicle's pose: three for position (x, y, z) and three for orientation (γ, β, α) in row, pitch and yaw angles. Similarly, in holonomic navigation six parameters are necessary to describe vehicle motion: three for translation $(\dot{x}, \dot{y}, \dot{z})$ and three for rotation $(\dot{\gamma}, \dot{\beta}, \dot{\alpha})$; these are the outputs, or tasks, of the visual odometry algorithm. Since they are derived from the same input data (the optical flow parameters discussed previously), it is natural to assume that there are dependencies between tasks, which if modeled properly could lead to improved results.

For this reason we use an extension to the GP framework, called Coupled GP (CGP) [27], where all tasks are estimated simultaneously and a full covariance matrix representing the uncertainty of the tasks and correlations between them is obtained. First of all, the training dataset is extended to incorporate all six tasks, assuming the form $\Lambda = \{\Lambda_i\}_{i=1}^6$ where $\Lambda_i = \{\mathbf{x}_n, y_{(n,i)}\}_{n=1}^N$ and N is the total number of data points. The multi-task covariance matrix is defined as $K = K_f \otimes K_x + \Sigma_n$, where $\otimes$ denotes the Kronecker product, K_f is a 6×6 positive-definite matrix that models the amplitude of correlations between each task (a multi-task analogue to σ_f^2 in Eq. 5) and Σ_n is a diagonal matrix with noise values. K_x is a 6×6 block-matrix where K_{ij} is the standard covariance matrix between tasks i and j. A cross-covariance function (Eq. 6) is used when $i \neq j$, derived from the definition of a neural network function in which two smoothing kernels are convolved [32] to obtain a positive-definite matrix that correlates multiple outputs:

$$k_{ij}(\mathbf{x}, \mathbf{x}') = \frac{\arcsin\left(\frac{2\widetilde{\mathbf{x}}^T \Sigma \widetilde{\mathbf{x}}'}{\sqrt{(1 + 2\widetilde{\mathbf{x}}^T \Sigma \widetilde{\mathbf{x}})(1 + 2\widetilde{\mathbf{x}'}^T \Sigma \widetilde{\mathbf{x}'})}}\right)}{(|\Sigma_i||\Sigma_j|)^4 \sqrt{|\Sigma_i + \Sigma_j|}}, \tag{6}$$

where $\Sigma = \Sigma_i(\Sigma_i + \Sigma_j)^{-1}\Sigma_j$. The predictive mean vector $\overline{\mathbf{f}^*}$ and covariance $\mathcal{V}(\overline{\mathbf{f}^*})$ for a single test point $\mathbf{x}^*$ is now calculated as

$$\overline{\mathbf{f}^*} = K_s^T K^{-1} \mathbf{y} \tag{7}$$

$$\mathcal{V}(\overline{\mathbf{f}^*}) = K_{ii}(\mathbf{x}^*, \mathbf{x}^*) - K_s^T K^{-1} K_s, \tag{8}$$

where

$$K_s = \begin{bmatrix} k^f_{1,1} k_{1,1}(\mathbf{x}^*, \mathbf{x}_{1,1}) & \dots & k^f_{6,1} k_{6,1}(\mathbf{x}^*, \mathbf{x}_{6,1}) \\ \vdots & \vdots & \vdots \\ k^f_{1,1} k_{1,1}(\mathbf{x}^*, \mathbf{x}_{1,N}) & \dots & k^f_{6,1} k_{6,1}(\mathbf{x}^*, \mathbf{x}_{6,N}) \\ \vdots & \vdots & \vdots \\ k^f_{1,6} k_{1,6}(\mathbf{x}^*, \mathbf{x}_{6,1}) & \dots & k^f_{6,6} k_{6,6}(\mathbf{x}^*, \mathbf{x}_{6,1}) \\ \vdots & \vdots & \vdots \\ k^f_{1,6} k_{1,6}(\mathbf{x}^*, \mathbf{x}_{6,N}) & \dots & k^f_{6,6} k_{6,6}(\mathbf{x}^*, \mathbf{x}_{6,N}) \end{bmatrix} \tag{9}$$

and

$$\mathbf{y} = [y_{1,1}, \dots, y_{1,N}, \dots, y_{6,1}, \dots, y_{6,N}]^T. \tag{10}$$

In Eq. 9, the indexes in ”$\mathbf{x}_{t,n}$ represent respectively the task number and the data point. The definition of K_s as a multi-column matrix, containing the relationship between the test point $\mathbf{x}_*$ and the training points from all tasks, is the main contribution of Coupled GPs over traditional multi-task GPs. This allows the simultaneous estimation of all components in the mean vector $\overline{\mathbf{f}^*}$, along with a full covariance matrix $\mathcal{V}(\overline{\mathbf{f}}_*)$ containing cross-dependencies between tasks.

C. Temporal Dependency

In the previous subsection we addressed cross-dependencies between tasks, which is a natural constraint in visual odometry applications. This is however not the only one, and here we explore temporal dependencies between tasks. 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. A first-order temporal dependency implies that $\overline{\mathbf{f}^*}_t$ will be correlated to $\overline{\mathbf{f}^*}_{t-1}$, and is modeled in the GP framework by incorporating $\overline{\mathbf{f}^*}_{t-1}$ into the input vector $\mathbf{x}_t$. So, for a test point with optical flow information $\mathbf{x}^*_t$ the new augmented input vector becomes

$$\mathbf{z}^*_t = \{\mathbf{x}^*_t, \overline{\mathbf{f}^*}_{t-1}\}. \tag{11}$$

During training, the hyperparameters of a covariance function are optimized as to minimize a certain cost function, and the temporal dependencies between tasks arise naturally. The log-marginal likelihood was chosen here to be the cost function due to its ability to penalize complexity, thus avoiding over-fitting:

$$\zeta = \ln p(\mathbf{y}|X) = -\frac{1}{2}\log(|K|) - \frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - N\log(2\pi). \tag{12}$$

This iterative process, however, disturbs the traditional training methodology, because the complete training set Z is not readily available for evaluation. It is possible to use ground-truth information directly to complete Z, but this would generate a best-case scenario that is not consistent with the inference step, where estimation errors will propagate to the next iterations. This error propagation can be incorporated to the training stage by dividing Λ into two subsets, Λ^1 and Λ^2, each composed of half the training data. In the first subset, the values of $\mathbf{y}^1$ are repeated to complete Z^1 (with a shift of -1, so that $\mathbf{y}^1_{i-1}$ completes Z^1_i), and then used to evaluate Z^2 iteratively. These steps are illustrated on lines 4-11 of Algorithm 1.

Algorithm 1 Temporal Dependency Training

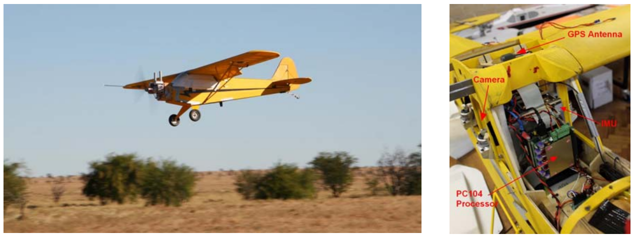
```
Input: Training Datasets $\Lambda^1$ and $\Lambda^2$
       Initial Hyperparameters $\theta$
Output: Optimized Hyperparameters $\theta$
 1: $likelihood\_new \leftarrow \infty$
 2: repeat
 3:   $likelihood\_old = likelihood\_new$
 4:   for $\mathbf{x}_i$ in $\Lambda^1$ do
 5:     $Z^1_i \leftarrow (\mathbf{x}_i, \mathbf{y}^1_{i-1})$
 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:     $Z^2_i \leftarrow (\mathbf{x}_i, \mathbf{y}_{CGP})$
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 $likelihood\_new - likelihood\_old = 0$
```

Once the evaluation process is complete, the estimated Z^2 from the second subset are used to optimize the CGP hyperparameters based on the log-marginal likelihood function defined in Eq. 12, according to the gradient-descent method (line 13 of Algorithm 1). It was determined empirically that the hyperparameters assigned as length-scales for $\overline{\mathbf{f}^*}$ should be kept from assuming too low values, since this would create a high sensibility to small errors in estimation. Once this 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.

This technique resembles the expectation-maximization (EM) algorithm, in the sense that it alternates between computing motion estimates from current hyperparameter values (the expectation step) and optimizing hyperparameters 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.

V. EXPERIMENTAL RESULTS

The visual odometry algorithm proposed in this paper was tested using data collected from a UAV flight (Fig. 2) 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 sensors and GPS that were fused to serve as ground-truth information. The first 4000 frames after aircraft estabilization were used for training, and the 2000 following frames were used for evaluation. The SIFT algorithm failed to find any matches in around 2% of the image pairs, due



(a) Unmanned Aerial Vehicle (b) Sensor payload

Fig. 2. Equipment used in the experiments.

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. We reiterate here that during evaluation the ground-truth information was used only for comparison purposes, and all results presented in this section were obtained using only images collected from a single camera.

Fig. 3 shows the CGP estimation results for each one of the tasks (red lines), along with the corresponding ground-truth information (black lines). It is clear from the proximity between these two values that the proposed method was indeed capable of learning the underlying transformation from optical flow to vehicle motion encoded on the training data, and then use the resulting model to predict estimates on new data. Linear uncertainty on the x axis is higher since this is the UAV's primary motion axis (forward), and angular tasks have a higher uncertainty in general due to a smaller number of samples in the training dataset, and also because angular motion is less constrained in this particular application.

The log-marginal likelihood values and accumulated errors for each training iteration are presented in Fig. 4. As expected, the negative likelihood values decrease steadily and the errors follow a similar trend, with occasional increases due to the multi-task nature of the optimization (certain errors might increase as others decrease).

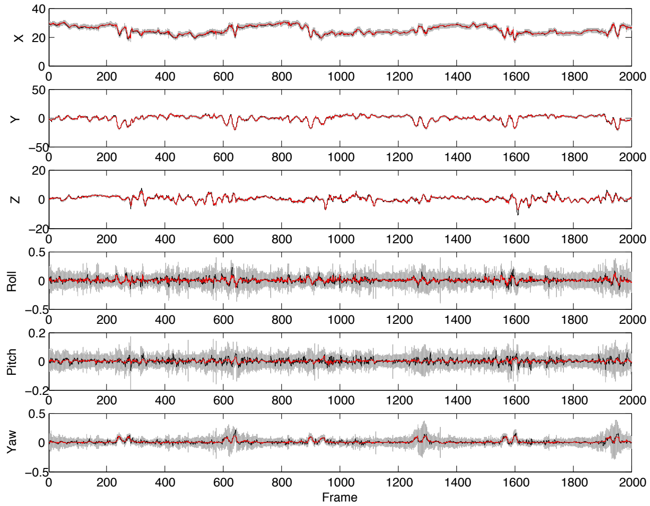
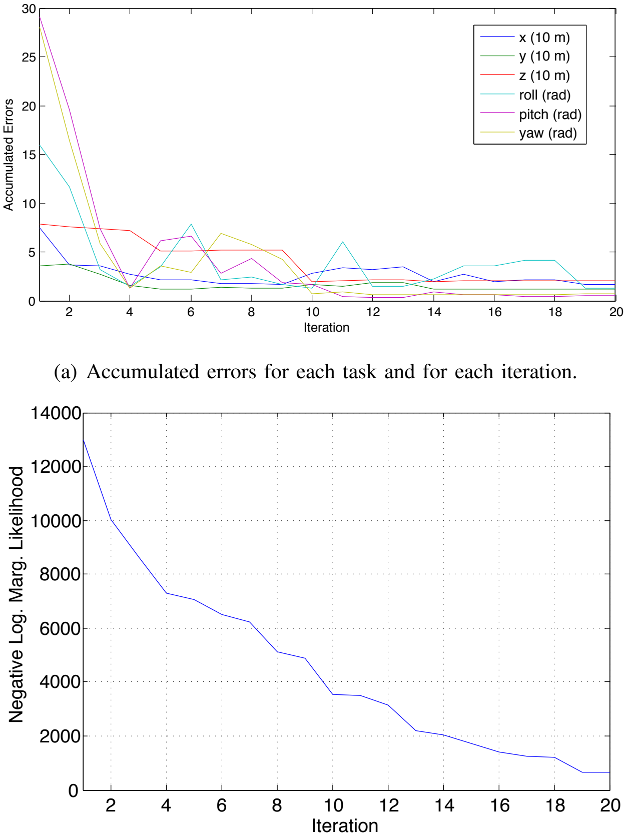


Fig. 3. CGP results for each task. Black lines represent ground-truth information, red lines represent the resulting motion estimates, and gray areas are the uncertainty within 2 standard deviations (95% confidence level).



(a) Accumulated errors for each task and for each iteration.

(b) Log-likelihood values for each iteration.

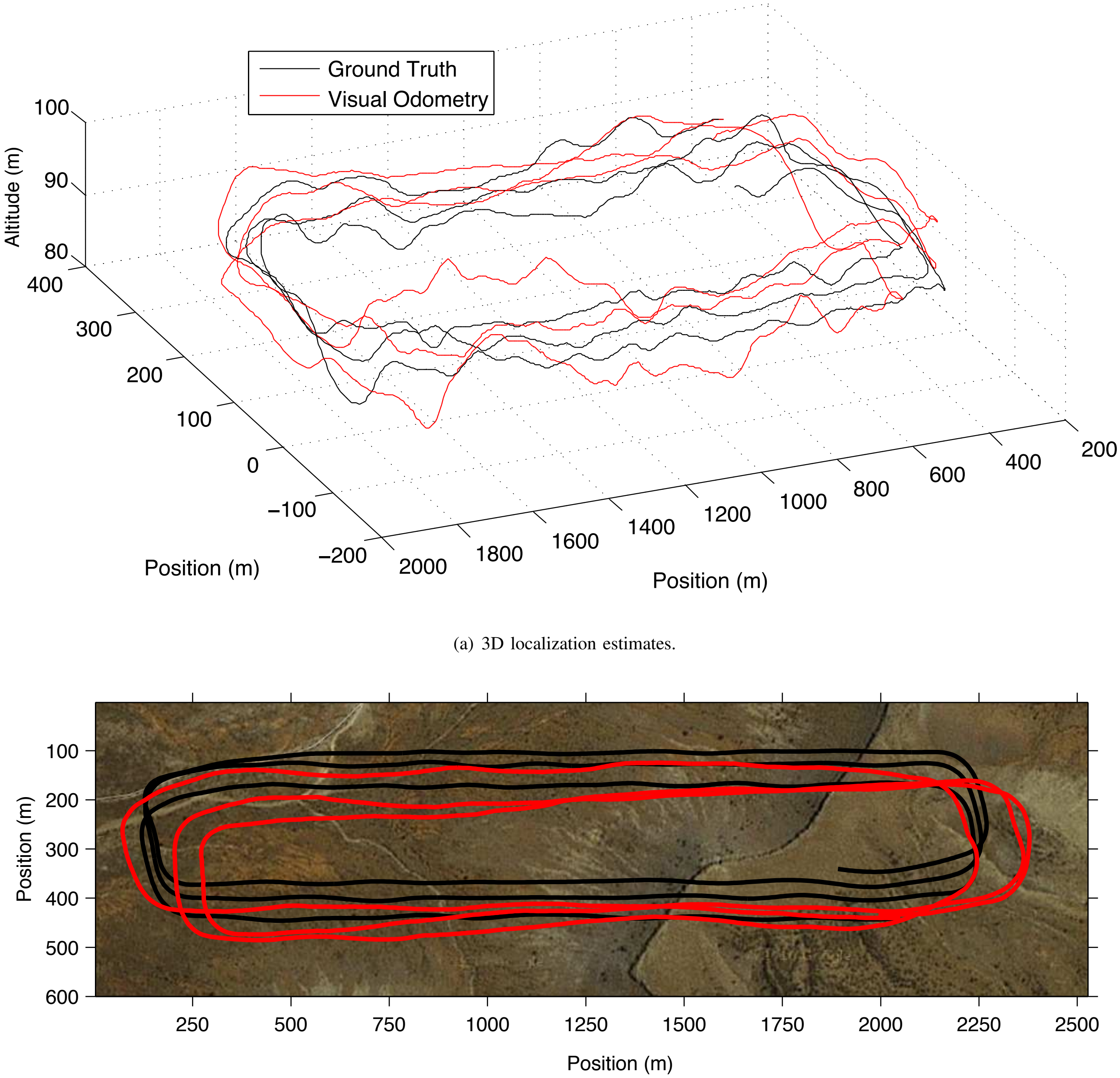
Fig. 4. Intermediary results from the training stage.

These results were then integrated to obtain a pose estimate for the UAV. Angular motion estimates are absolute and therefore can be simply added together over time, such that $(\gamma, \beta, \alpha)_{t+1} = (\gamma, \beta, \alpha)_t + (\dot{\gamma}, \dot{\beta}, \dot{\alpha})_t$. Linear motion estimates are relative to aircraft orientation, and can be projected back into the global coordinate system using the rotation matrix

$$R = \begin{bmatrix} c_\gamma c_\alpha - c_\beta s_\gamma s_\alpha & c_\alpha s_\gamma + c_\gamma c_\beta s_\alpha & s_\beta s_\alpha \\ -c_\beta c_\alpha s_\gamma - c_\gamma s_\alpha & c_\gamma c_\beta c_\alpha - s_\gamma s_\alpha & c_\alpha s_\beta \\ s_\gamma s_\beta & -c_\gamma s_\beta & c_\beta \end{bmatrix}, \quad (13)$$

such that $(x, y, z)_{t+1} = (x, y, z)_t + R_t^T(\dot{x}, \dot{y}, \dot{z})$. The final localization estimates are depicted in Fig. 5, along with ground-truth obtained from the inertial sensors and GPS. The flight trajectory was mostly horizontal, and Fig. 5(b) shows that its overall shape was maintained, with no missing corners or changes in the plane of navigation[1]. The relative scale (indicated by the linear tasks in Fig. 3) was also recovered to a high degree of precision (with an accumulated

[1] A video showing the iterative construction of this map was also submitted.



(a) 3D localization estimates.

(b) Top view of the flight trajectory (background from Googlemaps).

Fig. 5. Evaluation results with the proposed Coupled GP method.

error smaller than 2%), estimated from the training data and extrapolated to address new points in the input space. Significant changes in altitude to areas with no training samples 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. The sporadic fusion of these estimates with an absolute localization sensor, such as a GPS, or the addition of a loop-closure algorithm, would improve localization further.

Similarly, in Fig. 5(a) we can see the cyclical changes in altitude during flight, ranging from 80 to 100 m. The high frequency of these variations poses a challenge for the GP as a regression tool, because of the fine line between what is a trend and should be modeled and what is noise and should be discarded. During training, the noise hyperparameters were actively maintained at low levels to avoid an under-fitting scenario. 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 timesteps.

The choice of using Coupled GPs for multi-task estimation is an expensive one, because the volume of training data increases in direct proportion to the number of tasks. This impacts the training stage in two ways: 1) Matrix inversion has a $\mathcal{O}(N^3)$ computational cost on the number of data points; 2) Hyperparameter optimization must be performed in a higher-dimensional space, which delays convergence and increase the number of local minima. To justify this approach, the same algorithm presented in this paper was implemented using six single independent GPs (SGP), thus forcibly eliminating any cross-correlation. During training, the expectation step was still conducted simultaneously to maintain temporal dependency between all tasks, and the maximization step was conducted independently, with each task responsible for its own set of hyperparameters. The localization results are depicted in Fig. 6, where we can see the impact of joint estimation in visual odometry. Even though relative scale was still recovered (but to a lesser degree of precision, see Table. I), horizontal misalignments are much more pronounced and accumulated errors in angular tasks create several changes in the plane of navigation.

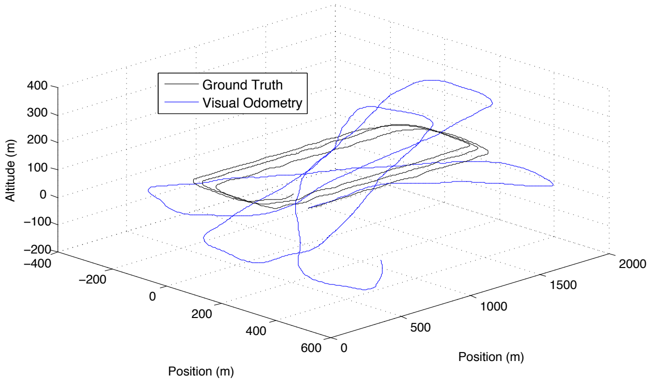


Fig. 6. 3D localization estimates obtained from the proposed method with six Single GPs.

A standard Structure-From-Motion (SFM) algorithm was also implemented and tested using the same dataset, as way to evaluate the performance of the method proposed against traditional approaches. The camera was calibrated and the geometrical model was based on the work of [2], using RANSAC to calculate the fundamental matrix and manual adjustment of scale. Results obtained using a generic urban vehicle dataset were comparable to other implementations found in the literature. The final localization estimates using the UAV dataset are shown in Fig. 7. We attribute this poor performance to three reasons: 1) Small and inconsistent overlapping areas between frames; 2) The high altitudes create a lack of depth perception in the ground plane; 3) Poor camera calibration, due to the narrow field of vision that affected the calibration process (the calibration board had to be positioned far from the camera). Further improvements to the standard SFM algorithm could lead to better results, but that was not explored in this work.

A quantitative comparison between these three methods is presented in Table I, where the root mean square error (rmse) for each task is given (all values are multiplied by 10^3) . As expected from previous localization results, the SFM approach has errors of around two orders of magnitude higher than the GP approach, and even though scale was manually adjusted the error in the forward linear axis (x) was still significantly larger than in any other task. Both GP approaches were capable of recovering scale automatically, however in the x axis CGP performed better with an error approximately 60% smaller than SGP. This improvement does not propagate to the other two linear tasks, due to the predominance of noise (see Fig. 5(a)) that was filtered out during the regression. CGP also performs better in all angular tasks, especially in pitch (improvement of 31%) and yaw (21%), both of which account for the misalignments found on Fig. 6.

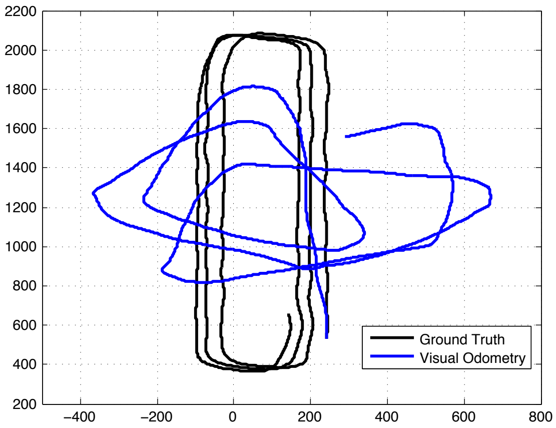


Fig. 7. 2D localization estimates with a SFM algorithm.

Task	Struct. Motion ($rmse$)	Single GP ($rmse$)	Coupled GP ($rmse$)
X	1384.10 ± 25.72	20.47 ± 0.1552	8.49 ± 0.0668
Y	453.56 ± 5.76	6.84 ± 0.0541	5.95 ± 0.0472
Z	325.50 ± 6.69	10.16 ± 0.0806	10.23 ± 0.0812
Roll	11.48 ± 0.56	0.69 ± 0.0056	0.66 ± 0.0053
Pitch	5.09 ± 0.01	0.35 ± 0.0027	0.26 ± 0.0021
Yaw	19.07 ± 0.55	0.41 ± 0.0032	0.33 ± 0.0027

TABLE I

LINEAR (10^{-3} m) AND ANGULAR (10^{-3} rad) ERRORS

VI. CONCLUSION

This paper presented a new technique for 3D motion estimation in visual odometry. Instead of relying on a calibrated camera model, we use a Gaussian Process to learn the underlying function that maps optical flow directly into vehicle motion. To account for cross-dependencies between tasks, the traditional GP implementation is extended to include multi-task estimation, using Coupled GPs, and an iterative filtering process accounts for temporal dependencies. Results obtained using data collected from a UAV flight over a trajectory of 20 km at speeds of 110 km show significant improvement over the standard visual odometry algorithm. Inference on new data can be performed at 10 Hz, a value suitable for real-time applications. As uncertainty estimates for each task are also provided, data fusion with other sensors and extension to the SLAM scenario is straightforward. Future work will focus on loop-closure algorithms and also on incorporating geometrical constraints into the GP framework itself, where the calibration parameters are learned as hyperparameters and used to provide an initial estimate that is then improved through further training.

VII. ACKNOWLEDGEMENTS

The authors thank Mitch Bryson for providing the UAV dataset used in the experiments. This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

REFERENCES

[1] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," *Int. Conference on Intelligent Robots and Systems (IROS)*, September 2008.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[3] C. E. Rasmussen and K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[4] J. Kelly and G. Sukhatme, "An experimental study of aerial stereovisual odometry," in *Proc. 6th IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

[5] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with underwater robot localization and tracking," in *Proc. Int. Conference on Robotics and Automation (ICRA)*, April 2007.

[6] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," *Int. Conference on Systems, Man and Cybernetics*, October 2005.

[7] M. Agrawal and K. Konolige, "Rough terrain visual odometry," in *Proc. Int. Conference on Advanced Robotics (ICAR)*, August 2007.

[8] J. Campbell, R. Sukthankar, and I. Nourbakhsh, "Techniques for evaluating optical flow for visual odometry in extreme terrain," in *Proc. Int. Conference on Intelligent Robots and Systems (IROS)*, 2004.

[9] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Ph.D. dissertation, Stanford University, 1980.

[10] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," *Int. Conference on Intelligent Robots and Systems (IROS)*, pp. 2531–2538, September 2008.

[11] D. Scaramuzza and R. Siegwart, "Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, October 2008.

[12] D. Nistr, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, January 2006.

[13] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel, *Visual Odometry using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle*, ser. Tagungsband Autonome Mobile Systeme. Springer Verlag, 2005.

[14] J. Kelly, S. Saripalli, and G. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," in *6th Int. Conference on Field and Service Robotics*, 2007.

[15] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *International Journal of Computer Vision*, 2007.

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

[17] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. 9th Int. Conference on Computer Vision (ICCV)*, October 2003.

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

[19] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints," in *Proc. Int. Conference on Computer Vision (ICCV)*, October 2009.

[20] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, "Memory-based learning for visual odometry," in *Proc. Int. Conference on Robotics and Automation (ICRA)*, 2008.

[21] R. Roberts, C. Potthast, and F. Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *Proc. Conference on computer Vision and Pattern Recognition*, June 2009.

[22] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *ECCV '92: Proceedings of the Second European Conference on Computer Vision*. London, UK: Springer-Verlag, 1992, pp. 321–334.

[23] S. O'Callaghan, F. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *Proc. Int. Conference on Robotics and Automation*, May 2009.

[24] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large scale terrain," in *Proc. Int. Conference on Robotics and Automation (ICRA)*, 2009.

[25] K. M. Chai, S. Klanke, C. Williams, and S. Vijayakumar, "Multitask gaussian process learning of robot inverse dynamics," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2008.

[26] P. Boyle and M. Frean, "Multiple output gaussian process regression," University of Wellington, Tech. Rep., 2005.

[27] V. Guizilini and F. Ramos, "Multi-task learning of visual odometry estimators," in *12th International Symposium on Experimental Robotics (ISER)*, 2010.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[29] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[30] C. K. I. Williams, "Computation with infinite neural networks," *Neural Computation*, 1998.

[31] R. M. Neal, *Bayesian Learning for Neural Networks*. Springer-Verlag New York Inc., 1996.

[32] D. Higdon, *Quantitative Methods for Current Environmental Issues*. Springer, 2002, ch. Space and Space-Time Modeling Using Process Convolutions, pp. 37–54.