2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Semi-parametric Models for Visual Odometry

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

Abstract— **This paper introduces a novel framework for estimating the motion of a robotic car from image information, a scenario widely known as visual odometry. Most current monocular visual odometry algorithms rely on a calibrated camera model and recover relative rotation and translation by tracking image features and applying geometrical constraints. This approach has some drawbacks: translation is recovered up to a scale, it requires camera calibration which can be tricky under certain conditions, and uncertainty estimates are not directly obtained. We propose an alternative approach that involves the use of semi-parametric statistical models as means to recover scale, infer camera parameters and provide uncertainty estimates given a training dataset. As opposed to conventional non-parametric machine learning procedures, where standard models for egomotion would be neglected, we present a novel framework in which the existing parametric models and powerful non-parametric Bayesian learning procedures are combined. We devise a multiple output Gaussian Process (GP) procedure, named Coupled GP, that uses a parametric model as the mean function and a non-stationary covariance function to map image features directly into vehicle motion. Additionally, this procedure is also able to infer joint uncertainty estimates (full covariance matrices) for rotation and translation. Experiments performed using data collected from a single camera under challenging conditions show that this technique outperforms traditional methods in trajectories of several kilometers.**

I. INTRODUCTION

This paper deals with the estimation of vehicle motion from image information, a problem commonly known in robotics as visual odometry. Accurate localization is a key aspect in most autonomous tasks, and visual systems provide several benefits that can lead to more robust and reliable results. Wheel encoders are unreliable due to slippage and terrain irregularities, inertial sensors (IMUs) suffer from velocity error accumulation, and GPS is limited to open environments. Visual information is highly descriptive, it is not restricted to any particular locomotion method, it is capable of a full 6 DoF motion estimation, and cameras are in overall inexpensive, compact and with low power consumption.

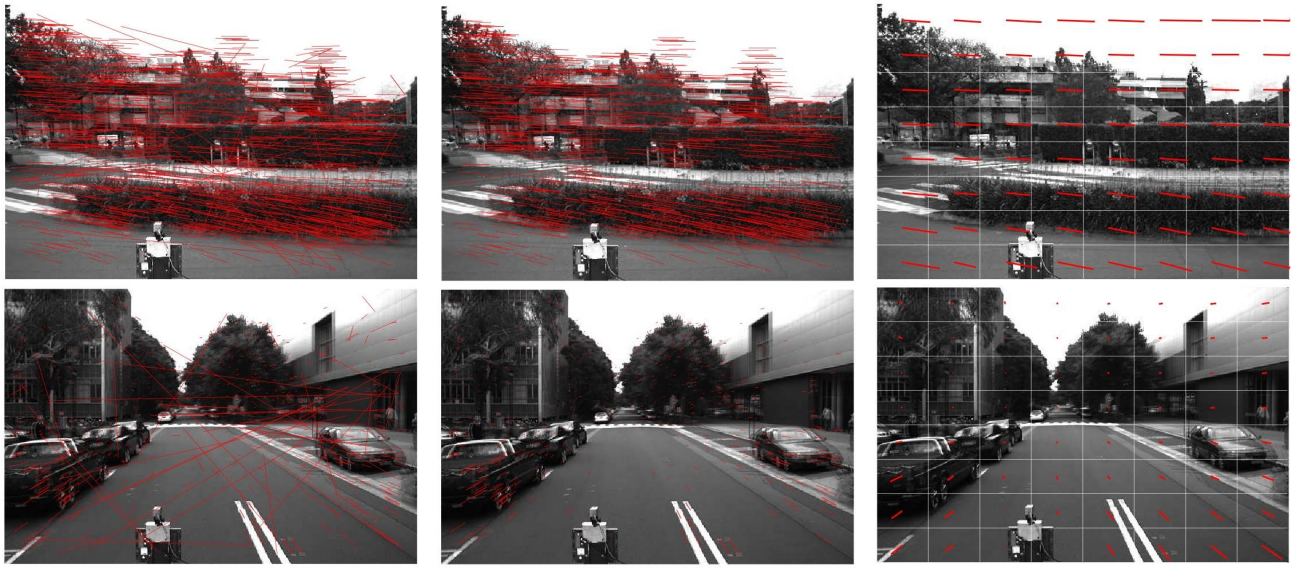
The vast majority of current visual odometry techniques address this problem geometrically [1], [2], using a calibrated camera model to calculate the camera motion hypothesis that best explains the optical flow values obtained from pairs of frames. Stereo [3], [4], [5], [6] and monocular [7], [8], [9] configurations have been successfully applied over the years in several areas, such as autonomous aircrafts [6], underwater vehicles [10], space exploration [11] and indoor/outdoor terrains [12], [7], [4], [13], [14]. Stereo configurations use a multi-camera array (or a moving camera) to capture several images simultaneously, and so are capable of recovering 3D feature locations directly from the binocular disparity between images. Monocular configurations, on the other hand, use a single camera and both feature triangulation and camera motion need to be estimated simultaneously, a scenario also referred to as "structure-from-motion" [15]. One well-known limitation of monocular visual odometry is the inability to recover absolute scale, a problem addressed in [9] for the special case of nonholonomic constraints and in [16] for a ground plane assumption.

Most current systems employ RANSAC [14] to test different camera motion hypothesis and elect the one with the highest probability of representing the optical flow values at hand. This process is usually followed by a global optimization method such as bundle adjustment [17], [18]. Self-calibration algorithms [19], [20] are also commonly used, as a way to eliminate the need for manual calibration. The incorporation of uncertainties to the final estimation leads to fusion of visual odometry data with other sensors, such as IMU or GPS [21], [12], or the extension to the SLAM framework [22].

Over the last few years, machine learning algorithms have been gaining territory in visual odometry applications as a way to eliminate the need for geometrical models and camera calibration. Machine learning techniques use training data, obtained from a different and independent sensor, to infer, in a supervised manner, the underlying function mapping optical flow to vehicle motion. One of the key benefits of this approach is the ability to infer scale directly from a monocular configuration, by exploring structure similarities between frames and how optical flow varies throughout the image. In [23] 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 in [24], where a constant pixel depth is assumed and the EM algorithm, in conjunction with an extension to PPCA, is used to learn a linear mapping between incremental motion and optical flow.

This paper is an attempt to combine both approaches into a single framework, where a geometrical model is used to obtain an initial estimate of vehicle motion that is further refined using ground-truth data during training. This semi-parametric learning procedure is the main contribution of this paper over previous works by the authors [25],

978-1-4673-1404-6/12/$31.00 ©2012 IEEE



(a) Initial SIFT matches (b) SIFT matches after RANSAC (c) Optical flow panel

Fig. 1. Visual information extracted from pairs of frames.

[26], where no prior knowledge of the camera system is assumed. The Coupled GP methodology is proposed and this geometrical model is incorporated as the mean function $m(\mathbf{x})$, usually assumed to be zero. We use the 7-point RANSAC algorithm to elect the most probable egomotion hypothesis based on sparse optical flow information, and the resulting fundamental matrix is combined with calibration parameters to recover vehicle motion up to a scale, as shown in [1]. Although these parameters can be provided manually using traditional calibration methods, here we obtain them automatically alongside with the GP hyperparameters, so initial calibration of the visual system is still not necessary.

The rest of this paper is divided as follows: Section II explains the process of extracting information from images and the optical flow parametrization into the input vector. Section III starts with an overview of Gaussian Processes, describes the Coupled GP extension and then introduces the geometrical model used and how it is incorporated into the GP framework. Section IV presents results obtained using real data in ground and aerial applications, and Section V concludes and discusses future work.

II. Optical Flow Parametrization

Our method uses sparse optical flow information obtained from consecutive pairs of frames during vehicle navigation. A histogram filter was initially applied to each frame to account for global luminosity changes. Due to its robustness and invariance properties, the initial feature extraction and matching is performed using the SIFT algorithm, as described in [27], although any other similar method could be readily applied.

Fig. I shows the three stages of optical flow parametrization. Examples of initial SIFT matches are presented in Fig 1(a), where it is possible to see a substantial number of false matches. These matches are filtered using the RANSAC algorithm, which is a probabilistic tool that elects the predominant camera motion scenario and discards matches that do not comply to this constraint. The resulting inlier sets are presented in Fig. 1(b). This step is also useful in minimising the impact of dynamic objects in the environment, since their optical flow will not be consistent with the rest of the image (assumed static).

The final stage of optical flow parametrization, depicted in Fig. 1(c), consists in dividing the image into fixed-size regions and averaging the optical flow information inside each one of these regions. Any featureless region is assumed to have the average optical flow value of its neighbour regions. This resulting optical flow panel is then reorganised into a vector $\mathbf{x}$ with dimension $2 * w * h$, where w and h are respectively the number of regions the image was divided horizontally and vertically. This vector will serve as input for the Coupled GP framework described in the next section. This procedure is necessary for two reasons: 1) Two different pairs of images will most certainly generate matching sets of different sizes, thus changing the dimension of $\mathbf{x}$ and the nature of the underlying function $f(\mathbf{x})$. 2) The coordinate in which each optical flow estimate was obtained is important because different areas react differently to camera motion. By organising the sparse optical flow information obtained from the SIFT features into a panel, we are able to fix both the dimensionality of the problem and maintain its spatial structure.

III. Motion Estimation

This paper proposes the union of a non-parametric Bayesian inference method, the Gaussian Process [28], with a parametrical geometric model defined by the camera configuration. The visual odometry problem, from the machine learning perspective, can be seen as a supervised regression problem, where an input vector $\mathbf{x} \in \Re^D$, composed of optical

flow information extracted from a pair of frames, is mapped to an output $y \in \Re$ containing the corresponding vehicle motion. A training dataset $\Lambda = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, composed of N data points obtained from a different and independent sensor, is used to optimize the parameters of a positive-definite kernel $k(\mathbf{x}, \mathbf{x}')$ that characterizes the relationship between inputs.

A. Overview of Gaussian Processes

Gaussian Processes (GPs) [28] are a non-parametric tool in the sense that they do not explicitly specify a functional model between inputs and outputs. A GP can be thought of as a Gaussian prior over the function space mapping inputs to outputs. It is characterized by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$,

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

Most traditional implementations assume $m(\mathbf{x}) = 0$ without loss of generality by scaling the data appropriately, and $k(\mathbf{x}, \mathbf{x}')$ is a positive-definite kernel (covariance function) whose coefficients are optimized to maximize a certain objective function (usually the marginal likelihood or leave-one-out cross validation). Due to its non-stationary properties and ability to model sharp transitions and non-linearities, we use here the neural network covariance function, as described in [29]:

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

where Σ is a diagonal matrix of length-scales, σ_f^2 is a signal variance used to scale the correlation between points and $\widetilde{\mathbf{x}} = \{1, \mathbf{x}\}$ is an augmented vector. 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

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

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

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 the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$.

B. Coupled GPs

In 3D navigation, six parameters (or tasks) are necessary to describe vehicle motion (the linear velocities on the x, y and z axis and the angular velocities $\dot{\gamma}$, $\dot{\beta}$ and $\dot{\alpha}$ in Euler angles). For the sake of simplicity, in this section we will focus on 2D navigation, where only two parameters (forward and angular velocities) are necessary, and the extension to a 3D scenario is achieved by the incorporation of all remaining tasks. Traditional implementations of GPs usually assume a single output, and multiple outputs are obtained using independent GP models. However, 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 the outputs which, if explored, could lead to better results. Alternative derivations [30] compute a single covariance matrix containing observations from all tasks, but each inference is still conducted independently. This paper uses a Coupled GP [25], where all tasks are inferred simultaneously and a full covariance matrix is estimated, representing the cross-correlation between tasks.

First of all, the training dataset Λ is divided into Λ_1 and Λ_2, where $\Lambda_i = \{\mathbf{x}_{(i,n)}, y_{(i,n)}\}_{n=1}^{N}$ represents the training data for task i. The new multi-task covariance matrix becomes

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

where K_f is a 2×2 positive-definite matrix, K_x is a $2N \times 2N$ covariance matrix between all the training points, Σ_n is a diagonal matrix with noise values, and $\otimes$ denotes the Kronecker product. The idea behind K_f is to model the amplitude of correlations between tasks, thus allowing dependencies to be naturally formed or discarded during training. K_x is defined as

$$K_x = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}, \tag{6}$$

where

$$K_{ij} = \begin{bmatrix} k_{ij}(\mathbf{x}_{i,1}, \mathbf{x}_{j,1}) & \dots & k_{ij}(\mathbf{x}_{i,1}, \mathbf{x}_{j,N}) \\ \vdots & \ddots & \vdots \\ k_{ij}(\mathbf{x}_{i,N}, \mathbf{x}_{j,1}) & \dots & k_{ij}(\mathbf{x}_{i,N}, \mathbf{x}_{N,N}) \end{bmatrix} \tag{7}$$

and k_{ij} indicates the covariance function utilized. When $i = j$ the auto-covariance function is used as in Eq. (2), and when $i \neq j$ a cross-covariance function is used, derived 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:

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

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 $\mathbf{x}_*$ are now calculated as

$$\overline{\mathbf{f}_*} = K_s^T K^{-1} \mathbf{y} \tag{9}$$

$$\mathcal{V}(\overline{\mathbf{f}_*}) = K_{ii}(\mathbf{x}_*, \mathbf{x}_*) - K_s^T K^{-1} K_s, \tag{10}$$

where

$$K_s = \begin{bmatrix} k_{1,1}^f k_{1,1}(\mathbf{x}_*, \mathbf{x}_{1,1}) & \dots & k_{2,1}^f k_{2,1}(\mathbf{x}_*, \mathbf{x}_{1,1}) \\ \vdots & \vdots & \vdots \\ k_{1,1}^f k_{1,1}(\mathbf{x}_*, \mathbf{x}_{1,N}) & \dots & k_{2,1}^f k_{2,1}(\mathbf{x}_*, \mathbf{x}_{1,N}) \\ k_{1,2}^f k_{1,2}(\mathbf{x}_*, \mathbf{x}_{2,1}) & \dots & k_{2,2}^f k_{2,2}(\mathbf{x}_*, \mathbf{x}_{2,1}) \\ \vdots & \vdots & \vdots \\ k_{1,2}^f k_{1,2}(\mathbf{x}_*, \mathbf{x}_{2,N}) & \dots & k_{2,2}^f k_{2,2}(\mathbf{x}_*, \mathbf{x}_{2,N}) \end{bmatrix} \tag{11}$$

and

$$\mathbf{y} = [y_{1,1}, \dots, y_{1,N}, \dots, y_{2,1}, \dots, y_{2,N}]^T. \tag{12}$$

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. Parametric Model

As stated before, most GP implementations assume $m(\mathbf{x}) = 0$, indicating no prior knowledge of the underlying function to be inferred from training data. This is however not the case in visual odometry, because it is possible to obtain an initial estimation of vehicle motion using well-established geometrical models [1]. These models are commonly used as a stand-alone solution to the structure-from-motion problem [14], [9], [17], and the main contribution of this paper is their incorporation into the GP framework, creating a semi-parametric approach to visual odometry.

The new mean vector $m(\mathbf{x})$ is obtained via triangulation, based on a calibrated camera model and a set of matched features (which is also used to estimate the optical flow that serves as input for the CGP). If image features are assumed to be static and their projections on both images are known (Fig. 2), it is possible to use this information to constrain the camera motion between frames and estimate translation and rotation.

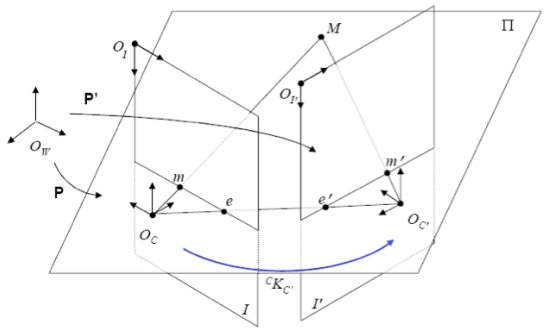


Fig. 2. Diagram of the geometrical constraints used to estimate vehicle motion from O_C to $O_{C'}$ according to a matched feature M and its projections m and m' on each image.

The first step is the calculation of the fundamental matrix, based on the inlier feature sets obtained in Sec. II. If $\mathbf{u}_{n=1}^{N}$ are the normalized pixel coordinates $(u, v, 1)^T$ of all the $N \geq 7$ inliers in a pair of frames, the fundamental matrix F is given by the optimization of

$$\mathbf{u}'^T F \mathbf{u} = 0. \tag{13}$$

Examples of epipolar lines obtained from Eq. 13 are presented in Fig. 3. The next step is the calculation of the essential matrix $E = C^T F C$, with C being the calibration matrix defined as

$$C = \begin{bmatrix} l_x & s & p_x \\ 0 & l_y & p_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

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. These are the camera intrinsic parameters, and the extrinsic parameters (translation $\mathbf{t}$ and rotation R) are extracted from E by identifying [1] the correct pair of projection matrices P_1 and P_2. If $P_1 = [I|0]$, meaning that the first frame is aligned at the center of the coordinate system, then $P_2 = [R|\mathbf{t}]$ indicates camera motion between frames. For the special case of 2D navigation we assume that there is no lateral and vertical vehicle motion, and no tilt or roll. In this scenario, the only two remaining degrees of freedom are the linear v and angular ω velocities, which together compose the mean vector[1] $m(\mathbf{x}) = \{v, \omega\}$. We define here $v = |\mathbf{t}|$ and ω as the yaw component of R.

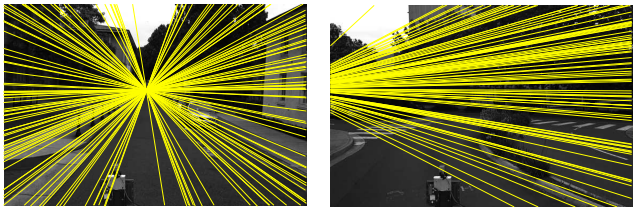


Fig. 3. Epipolar constraints during vehicle translation (left) and rotation (right).

Inference for a single test point $\mathbf{x}_*$ is now defined by Eqs. 15 and 16. By adding the mean function to $\overline{\mathbf{f}_*}$ we assure that, as the testing sample $\mathbf{x}_*$ deviates from the training dataset Λ, the outputs converge to the results provided by the geometrical model. As expected, the incorporation of $m(\mathbf{x})$ to the inference procedure does not change the estimation of $\mathcal{V}(\overline{\mathbf{f}_*})$, since the geometrical model does not provide any measure of uncertainty.

$$\overline{\mathbf{f}_*} = m(\mathbf{x}_*) + K_s^T K^{-1} \left(\mathbf{y} - m(\mathbf{x})\right), \tag{15}$$

$$\mathcal{V}(\overline{\mathbf{f}_*}) = K_{ii}(\mathbf{x}_*, \mathbf{x}_*) - K_s^T K^{-1} K_s. \tag{16}$$

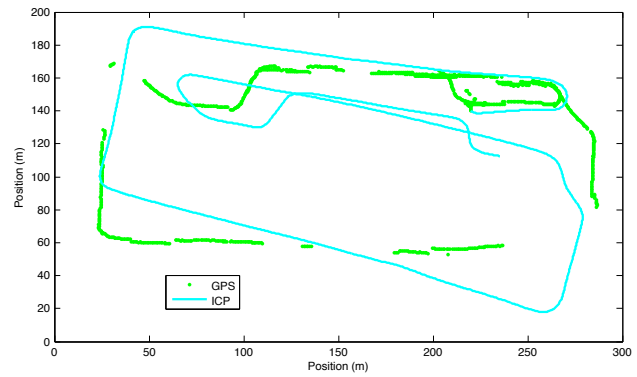
D. Parameter Optimization

During the training stage, the covariance function coefficients (the hyperparameters) are optimized according to a cost function. Due to its ability to balance between model complexity and data fit, we choose here the marginal likelihood function, shown in Eq. 17 where $\epsilon = (\mathbf{y} - m(\mathbf{x})$. In the CGP framework these hyperparameters are composed of the length-scales in Σ_1 and Σ_2, the noise values in Σ_n and the correlation amplitudes in K_f. The optimization is conducted using a combination of stochastic maximization (simulated annealing) and gradient descent algorithms to reduce the influence of initial conditions,

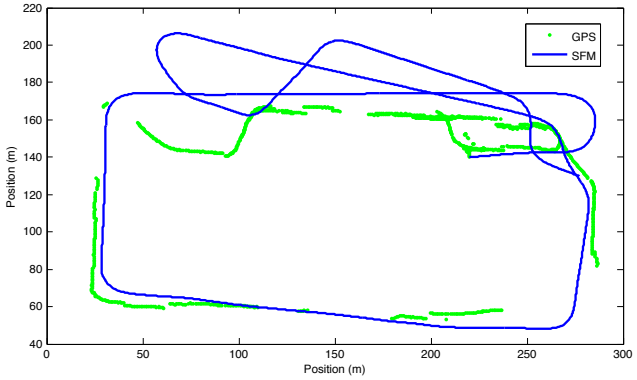
$$\zeta = \ln p(\mathbf{y}|X) = -\frac{\log(|K|)}{2} - \frac{\epsilon^T K^{-1} \epsilon}{2} - N \log(2\pi). \tag{17}$$

However, the incorporation of a geometrical model into the framework introduces a new set of parameters, the calibration

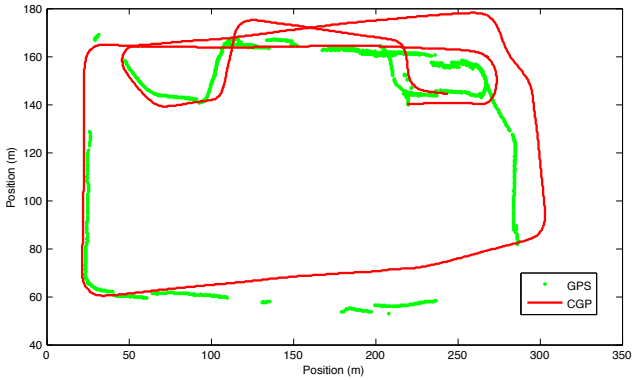
[1]For the general case of 3D navigation, the mean vector becomes $m(\mathbf{x}) = \{\dot{x}, \dot{y}, \dot{z}, \dot{\gamma}, \dot{\beta}, \dot{\alpha}\}$, where $(\dot{x}, \dot{y}, \dot{z})$ are the linear velocities in each axis and $(\dot{\gamma}, \dot{\beta}, \dot{\alpha})$ are the angular velocities in Euler angles.



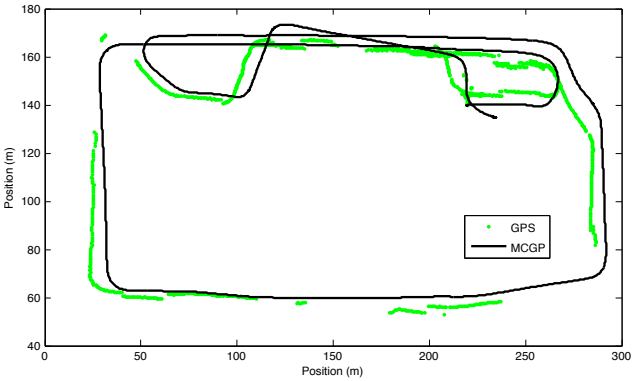
(a) Laser-based Iterative Closest Point (ICP)



(b) Structure From Motion (SFM) with manual scale adjustment



(c) Coupled GP



(d) Coupled GP + Geometrical Model

Fig. 4. Localization results obtained using different methods.

parameters present in C (focal lengths l_x and l_y, skew s and image center coordinates p_x and p_y). Although manual camera calibration could provide estimates for these parameters, we propose their incorporation into the optimization process, along with the CGP hyperparameters. The benefits of this approach are two-fold: 1) It eliminates the need for camera calibration, and if these parameters are available they can be further refined. 2) Since the geometrical model is used only as an initial estimate for the CGP inference, the actual values of these parameters may differ from the ones provided by an independent calibration.

IV. Results

The proposed methodology was first evaluated in a 2D scenario, using data collected from a ground vehicle navigating in outdoor environments (both urban and off-road). In this case only two tasks are necessary to describe vehicle motion, decreasing the number of hyperparameters, computer memory requirements and training time. The same methodology was then extended to a 3D scenario, using data collected from an unmanned aerial vehicle (UAV) flight over a deserted area. During flight, the UAV is capable of moving in all six degrees of freedom, which can be extrapolated to any generic application of visual odometry.

A. Ground Vehicles

For the ground vehicle tests, a conventional car (Fig. 5) was modified to include a camera, a laser sensor and a GPS system (used solely for comparison purposes). The camera captured images at roughly 5 frames per second at a 1152x758 pixel resolution, which were then downsampled to 384x252 pixels. 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. During data acquisition the car moved at speeds of up to 40 km/h and interacted with pedestrians and other vehicles.

Fig. 5. Car used in experiments.

(a) Urban testing dataset (b) Off-road testing dataset

Fig. 6. Localization results obtained using different methods (green dots are GPS information, red lines are visual odometry estimates and yellow circles represent loop-closures.

The training dataset is composed of 2000 images acquired in an urban environment. Ground-truth information was obtained using the Iterative Closest Point (ICP) algorithm [31] based on laser data, as depicted in Fig. 7. Because they are incremental, these estimates are by themselves subject to drift. Even though more precise results could be obtained (i.e. by fusing laser and GPS information), the CGP approach is in general capable of averaging over such errors due to a large number of samples in the training dataset. This is beneficial because it eliminates the need of high-precision sensors during the training stage.

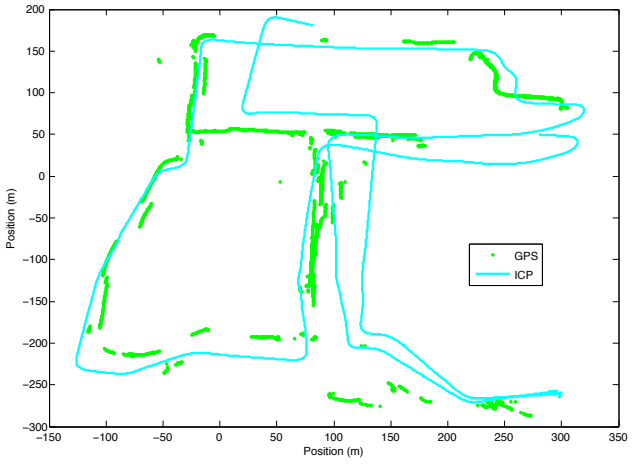


Fig. 7. Training Dataset

The urban testing dataset is also composed of 2000 images, obtained using the same vehicle over a different trajectory of roughly 2 km. For comparison purposes only, localization results obtained from the same ICP algorithm used previously on the training dataset are depicted in Fig. 4(a). Similar results obtained using the calibrated geometrical model and manual scale adjustment are presented in Fig. 4(b). As expected, both approaches suffer from error accumulation due to drift, specially in rotation because of smaller overlapping areas and higher sensitivity to angular motion.

Fig. 4(c) shows the localization results obtained using the CGP framework without the geometrical model, assuming $m(\mathbf{x}) = 0$. It is possible to see the GP's ability to recover scale, which is a non-trivial task in monocular configurations. Even though less predominant, imprecisions in angular motion estimates still constitute the main source of accumulated drift errors, mostly due to the lower number of sample curves for training and smaller overlapping areas in the image. In Fig. 4(d) the localization results obtained from the CGP framework with the geometrical model (MCGP) are presented. The calibration parameters were optimized as hyperparameters with random initial guesses. 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 geometrical model is used.

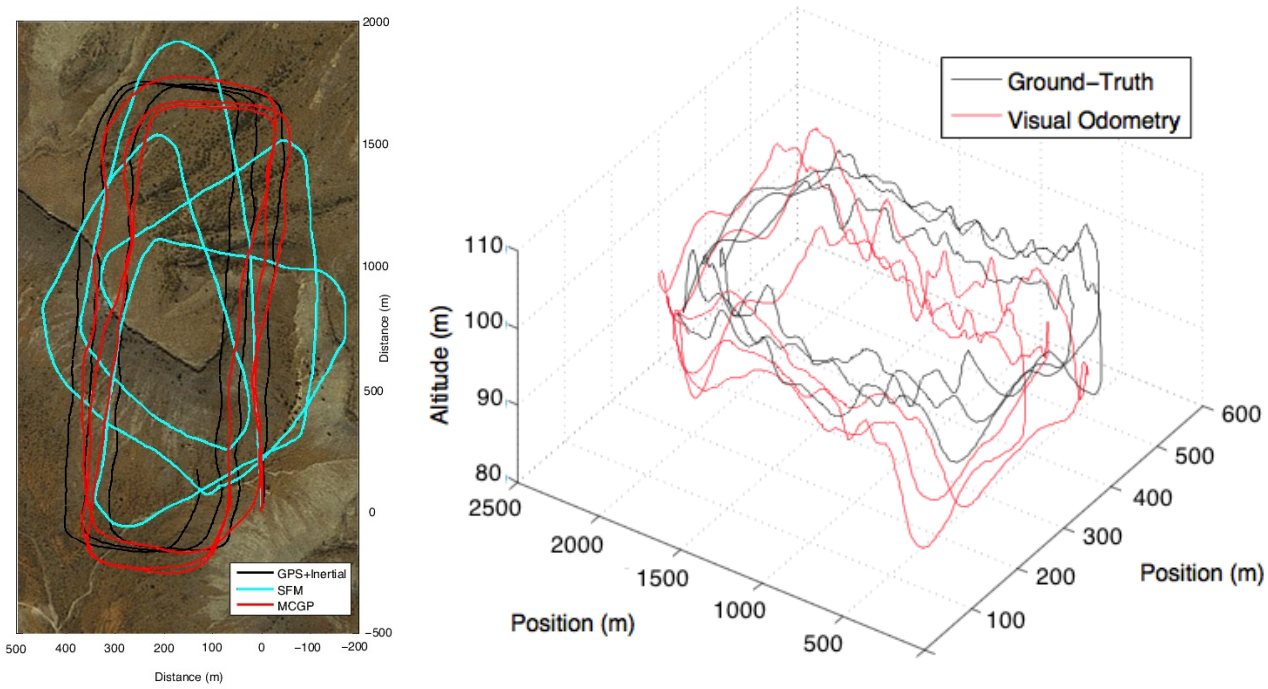


Fig. 8. 3D localization results from the UAV dataset.

Finally, the proposed methodology was incorporated into an Exactly Sparse Information Filter (ESIF) [32], taking advantage of the probabilistic nature of Gaussian Processes to explore the extension to a SLAM (Simultaneous Localization and Mapping) scenario. This was possible because the CGP framework is capable of providing a full covariance matrix, containing both auto and cross-dependencies between tasks. All vehicle position and corresponding uncertainties are tracked over time, and a loop-closure algorithm was implemented based on feature matching between frames. When a certain area is revisited the ESIF uses this information to retroactively decrease global uncertainty.

Localization results obtained using the same testing dataset are presented in Fig. 6(a), where it is possible to see loop-closures during the second pass over the left street and also when the vehicle returns to the starting point. During the third pass, the vehicle was facing the opposite direction, so it was unable to match any images, resulting in a residual misalignment in this area. The incorporation of information from other sensors, such as GPS, could further improve the results. We also tested this algorithm, using the same vehicle and without further training, in an off-road environment of roughly 3 km. This environment is composed mostly of trees with the vehicle driving over grass, and the ESIF results are depicted in Fig. 6(b). These results testify to the ability of the proposed method to generalise over different environments.

A quantitative comparison of all localization methods presented in this paper is shown in Table I, in terms of Root Mean Squared Error (rmse) per frame. The ground-truth for these comparisons was obtained using ICP estimates integrated into the ESIF framework. As expected, ICP has the lowest translational error, because distances can be measured directly from a laser scanner. The standard CGP framework performed better than the structure-from-motion approach (composed solely of the geometrical model described previously), however the semi-parametric approach outperformed both, especially on angular motion estimation, which is arguably the main source of accumulated errors. Even though small, these angular motion errors are accumulated in a few frames since the vehicle mostly drives on a straight line. The MCGP-SLAM approach was able to further decrease rotational error, by eliminating drift misalignments during loop-closure.

Method	**Trans. Error (rmse)** (10^{-2} **m**)	**Rot. Error (rmse)** (10^{-2} **rad**)
It. Closest Point	2.92 ± 4.70	0.06 ± 0.14
Struct. Motion	9.75 ± 12.12	0.23 ± 0.16
Coupled GP	5.74 ± 8.18	0.07 ± 0.08
MCGP	5.12 ± 7.49	0.05 ± 0.07
MCGP-SLAM	5.98 ± 6.54	0.04 ± 0.07

TABLE I
LINEAR AND ANGULAR ERRORS

B. Aerial Vehicles

The data used during aerial vehicle tests was collected from a UAV flying at speeds of up to 110 km/h and heights of 80-100 m. A camera pointing downwards was used to collect images at a rate of 3 frames per second, and a fusion of GPS and inertial data served as ground-truth information. This scenario is specially tricky for a calibration-based method due to the high altitudes, which create a lack of depth perception in the ground plane and require a narrow field of

vision from the camera. The lack of overlapping areas caused by severe camera motion also difficults feature matching, resulting in frame pairs with poor or non-existent optical flow.

Fig. 8 shows the localization results obtained using different methods. The black line denotes ground-truth information, where it is possible to see the UAV's overall motion pattern of elongated rectangles with a slight translation sideways. It is also possible to see that the structure-from-motion algorithm (with manual scale adjustment, cyan line) fails to correctly estimate vehicle rotation due to the difficulties listed above, generating errors that rapidly accumulate. However, these estimation errors are individually small and the MCGP uses the structure-from-motion results as initial guesses that are further refined by the CGP framework, compensating for most of the residual drift.

V. Conclusion

We presented a technique to incorporate a parametric model (the geometrical constraints from a camera) into a non-parametric algorithm (the Gaussian Process), creating a semi-parametric framework that benefits from both approaches. The simultaneous optimization of both the calibration parameters and the hyperparameters eliminates the need for prior calibration of the visual system, and if this information is available these parameters can be further refined seamlessly. This technique is capable of recovering scale in a monocular configuration, generalizing over different environments without the need of further training, and the estimation of uncertainties allow the use of results in filtering and SLAM frameworks. While the training stage may take up to a few hours, depending on the number of tasks and random hyperparameter initialization, new inferences can be computed at a rate of 10 Hz, thus being suitable for real-time applications. The results presented here challenge existing visual odometry algorithms given the magnitude of trajectories in our experiments. Future work will focus on different geometrical models, the incorporation of a geometrical model into the covariance function, and the effects of exchanging cameras after training.

References

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

[2] L. Matthies, "Dynamic stereo vision," Ph.D. dissertation, Carnegie Mellon University, 1989.

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

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

[5] Z. W. Zhu, T. Oskiper, O. Naroditsky, S. Samarasekera, H. S. Sawhney, and R. Kumar, "An improved stereo-based visual odometry system," in *Proc. Workshop of Performance Metrics for Intelligent Systems*, 2006.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

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

[15] 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.

[16] H. Wang, K. Yuan, W. Zou, and Q. Zhou, "Visual odometry based on locally planar ground assumption," in *International Conference on Information Acquisition*, 2005.

[17] 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.

[18] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ser. ICCV. Springer-Verlag, 2000.

[19] 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.

[20] J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel, "Camera self-calibration for sequential bayesian structure from motion," in *Proc. Int. Conference on Robotics and Automation*, 2009.

[21] 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.

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

[23] 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.

[24] 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.

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

[26] ——, "Visual odometry learning for unmanned aerial vehicles," in *Proc. Int. Conference on Robotics and Automation (ICRA)*, 2011.

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

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

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

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

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

[32] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based slam:," *International Journal of Robotics Research*, vol. 26, no. 335-359, 2007.