

# Computing MAP trajectories by representing, propagating and combining PDFs over groups

Paul Smith

Department of Engineering  
University of Cambridge  
Cambridge CB2 1PZ, UK  
pas1001@eng.cam.ac.uk

Tom Drummond

Department of Engineering  
University of Cambridge  
Cambridge CB2 1PZ, UK  
twd20@eng.cam.ac.uk

Kimon Roussopoulos

Pi Group Ltd  
Milton Hall, Milton  
Cambridge CB4 6WZ, UK  
kimon.roussopoulos@pigroup.co.uk

## Abstract

This paper addresses the problem of computing the trajectory of a camera from sparse positional measurements that have been obtained from visual localisation, and dense differential measurements from odometry or inertial sensors. A fast method is presented for fusing these two sources of information to obtain the maximum a posteriori estimate of the trajectory. A formalism is introduced for representing probability density functions over Euclidean transformations, and it is shown how these density functions can be propagated along the data sequence and how multiple estimates of a transformation can be combined. A three-pass algorithm is described which makes use of these results to yield the trajectory of the camera.

Simulation results are presented which are validated against a physical analogue of the vision problem, and results are then shown from sequences of approximately 1,800 frames captured from a video camera mounted on a go-kart. Several of these frames are processed using computer vision to obtain estimates of the position of the go-kart. The algorithm fuses these estimates with odometry from the entire sequence in 150 mS to obtain the trajectory of the kart.

## 1. Introduction

This paper addresses the problem of computing the trajectory of a camera by combining vision and odometry (or inertia). This problem has applications in mobile robotics and is also important for obtaining an initialisation which can then be used to solve the more general structure and motion problem.

The problem is shown in Figure 1. A camera mounted on a vehicle follows some trajectory. Odometry measurements giving estimates of the motion between each frame ( $M_i$ ) are available with known uncertainties. Some of the

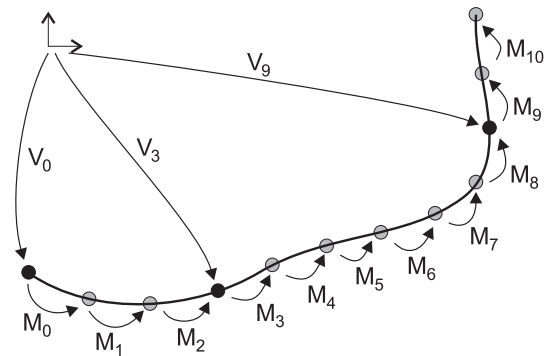


Figure 1: **An example vehicle path.** Motion information  $M_i$  is known continuously along the path, but estimates for the absolute pose  $v_i$  are only known in a few places. The task is to optimally estimate the absolute pose of each point along the trajectory.

frames have been processed using computer vision to give estimates of the pose of the camera, also with uncertainties ( $V_0$ ,  $V_3$  and  $V_9$  in the example). The goal is to obtain a maximum a posteriori (MAP) estimate for the pose of the camera at each frame in the sequence ( $E_i$ ) that maximises

$$\prod_i P(M_i|E_i) \prod_{i:\exists V_i} P(V_i|E_i) \quad (1)$$

In the problem considered in this paper, the sequences consist of approximately 1,800 frames, of which about 30 have been processed to give position estimates (there are therefore approximately 5,400 degrees of freedom).

To illustrate why this problem is not as straightforward as it might seem, consider the simple case where the odometry states that the vehicle has travelled forward in a straight line one unit per time step in the  $x$  direction for  $n$  frames. Position information from the first and last frames, however, shows that the vehicle has undergone a sideways translation in the  $y$  direction in addition to its forward motion, and is pointing in the same direction at both ends of the sequence (see Figure 2(a)). Further, assume that the position infor-

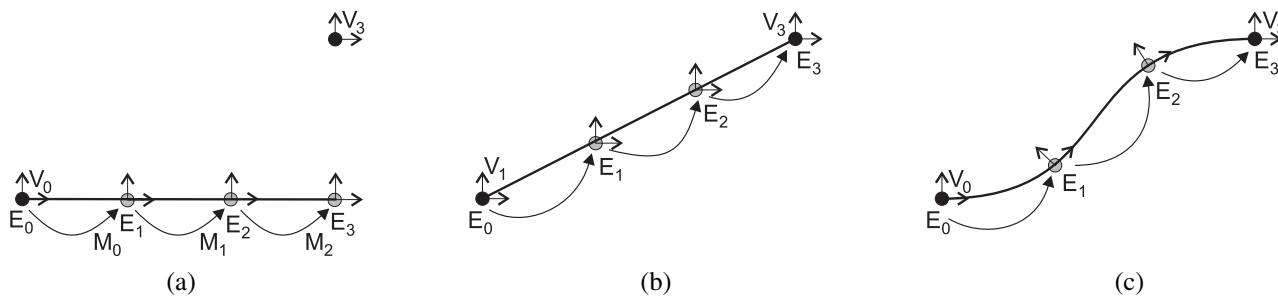


Figure 2: **An example trajectory.** a) Odometry information  $M_i$  indicates an  $x$ -translation, while the vision measurements  $V_0$  and  $V_3$  indicate that there is also some  $y$ -translation, but no rotation; b) the naïve trajectory; c) the correct trajectory.

mation is accurate, but that each frame in the odometry contains independent  $x$ ,  $y$  and  $\theta$  motion errors.

The simple solution to this problem is to optimise in  $x$ ,  $y$ ,  $\theta$  space. Since the vehicle has not rotated according to the position estimates (and this is in agreement with the odometry), the solution obtained by Kalman smoothing [9] (or any similar optimisation) shows no rotation in any of the pose estimates for the sequence. Further, the final estimate for the vehicle's motion is a straight line between the two position measurements (see Figure 2(b)). **Unfortunately this is the wrong answer.** Figure 2(c) shows the correct trajectory.

## 1.1. Background

This problem is important in both computer vision and robotics research. In computer vision it is often seen as part of the structure from motion (SFM) problem [7, 8, 11], where the task is the identification of not just the trajectory, but also the building of a three-dimensional model of the scene. These techniques tend to rely solely on vision information, and rely heavily on the ability to track features throughout the sequence, something which is not possible in every application. In robotics the same problem (although often in only two dimensions) is known as simultaneous localisation and map-building (SLAM) [3, 10], although many robotics applications are solely concerned with robot localisation [2, 6, 12, 13, 14, 15]. In robotics applications, vision is usually used as an additional information source to be used together with some other system such as odometry or inertial sensors, the fusion of the two systems providing much greater accuracy and robustness than could be achieved by either alone [15].

Robotics applications are often driven by a requirement for real-time position information (e.g. [3, 10]). This, however, is unable to provide the optimal trajectory estimate, since this can only be calculated after the event. Optimal trajectory estimation is achieved in structure from motion cases by bundle adjustment [8, 16], a time-consuming global optimisation. The case of optimal (off-line) trajectory estimation by combining multiple data sources has only recently been considered in the robotics literature

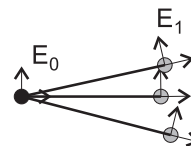


Figure 3: **Three possible poses for  $E_1$ .** Moving from  $E_0$  to  $E_1$  has uncertainties in both translation and rotation, and the final pose therefore has correlated uncertainty between translation and rotation.

[4, 13, 15], the latter two of which tackle problems very similar to the one considered in this paper. Jung and Taylor's scheme [13] is not optimal, instead largely being an interpolation scheme. Stenlow and Singh [15] consider the case of combined vision and inertial measurements, and perform Levenberg-Marquardt optimisation over an total error function to yield an optimal solution. However, they only consider isotropic covariances and do not discuss the problems of correctly propagating probability distributions.

## 1.2. Approach

The reason that the trajectory shown in Figure 2(b) is incorrect is that although the translation and rotation errors are independent at each link in the chain, the accumulated errors do not remain so. Consider the single link shown in Figure 3. If independent errors in translation and rotation are applied at the start of the link, the error in pose seen at the end of the link has a correlation between  $y$  translation and rotation.

In order to solve the problem correctly, it is necessary to formally describe poses and the distributions of these errors. This takes a little care since the manifold of poses in 3D space has non-zero Riemann curvature. Section 2 introduces a formalism for doing this, and then shows how two estimates of a pose with their uncertainties can be resolved to find a MAP estimate. Section 3 then shows how uncertainties are propagated and gives a three-pass algorithm for obtaining a trajectory.

The problem of finding a trajectory has a physical analogue of finding the minimum energy configuration of an elastically deformable object (e.g. a steel ruler). Section

4 shows simulation results validated against this analogue, and then results from sequences obtained from a camera mounted on a go-kart are shown. It is also described how this formulation copes elegantly with uncertainty in the synchronisation of the two data sources.

## 2. Formulation

In this paper, poses and motions are represented as co-ordinate frame transformations. A pose (e.g.  $E$ ,  $V$ ) is the transformation from the world coordinate frame to that of the camera, and a motion (e.g.  $M$ ) from time  $t$  to  $t + 1$  is the transformation from the coordinate frame of the camera at time  $t$  to that at  $t + 1$ . These transformations are represented by  $4 \times 4$  matrices:

$$E, V, M = \begin{bmatrix} R & \mathbf{t} \\ 000 & 1 \end{bmatrix}$$

Matrices of this type form the Lie group  $SE(3)$  and hence uncertainties in a pose or motion correspond to probability density functions over  $SE(3)$ .

Because the manifold structure of  $SE(3)$  has non-zero curvature, defining probability density functions takes a little care. One way of doing this is to define a chart. This has been done using  $\mathbf{t}$  and representing  $R$  using Euler angles [15], but this approach has difficulties because the solid angle swept out by a given change in Euler angles depends on the current pose. In this paper, we make use of the exponential map to provide a natural frame in which to define probability density functions.

The exponential map [1, 17] is a map from the tangent space of the group at the identity element to the group and, using this, the tangent space can be used to provide a chart for a region around the identity, or even the whole group. Note that elements in this tangent space are also elements of the Lie algebra of the group. A natural choice of basis for the tangent space to  $SE(3)$  (also known as the generators of the group) is given by:

$$\begin{aligned} G_1 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & G_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & G_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ G_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & G_5 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & G_6 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

These generators represent the derivatives of  $x$ ,  $y$  and  $z$  translations and  $x$ ,  $y$  and  $z$  axis rotations respectively. The exponential map then maps a linear combination of these generator matrices into a transformation matrix by

$$E = e^{\sum_i \alpha_i G_i} \quad (2)$$

for which the shorthand  $E = e^\alpha$  will be adopted in the rest of this paper.

Probability density functions over groups can be represented using the exponential map by defining a probability density function over  $\hat{\alpha} \in \mathbb{R}^6$ . The random variable  $\hat{E}$  can then be defined by  $\hat{E} = e^{\hat{\alpha}}$ .

It is often convenient to consider zero-mean distributions, in which case  $\hat{E}$  can be written, for Gaussian errors,

$$\hat{E} = e^{\alpha} E \quad \alpha \sim \mathcal{N}[0, C] \quad (3)$$

Writing the random transformation on the left of a mean transformation  $E$  is equivalent to representing the uncertainty in the target co-ordinate frame. It is also possible to represent uncertainties in the source frame by

$$\hat{E} = E e^{\alpha'} \quad \alpha' \sim \mathcal{N}[0, C'] \quad (4)$$

but  $E$  and  $e^\alpha$  do not necessarily commute, so in general  $C \neq C'$ . There is, however, a relationship between  $C$  and  $C'$ , given by the adjoint representation of  $E$ . The adjoint representation of a group is a map from the group to linear transformations on its algebra and can be defined by

$$e^{\text{Ad}(E)\epsilon} = E e^\epsilon E^{-1} \quad (5)$$

i.e.  $\text{Ad}(E)$  is the action of  $E$  on the Lie algebra by conjugation. Equation (5) can be rewritten as

$$e^{\text{Ad}(E)\epsilon} E = E e^\epsilon \quad (6)$$

hence  $\alpha$  and  $\alpha'$  from (3) and (4) are related by

$$\alpha = \text{Ad}(E)\alpha' \quad (7)$$

and  $C$  and  $C'$  are related by

$$C = \text{Ad}(E)C' \text{Ad}(E)^T \quad (8)$$

This equation will be important in Section 3 to ensure that the uncertainties are correctly represented as they are propagated.

### 2.1. Optimally combining two sources of pose information

It is also necessary to know how to combine two independent pose estimates to give the optimal (maximum *a posteriori*) estimate of the pose, and the associated covariance. This Section will consider the case of combining two transformations,  $E_1$  and  $E_2$ , with error covariances  $C_1$  and  $C_2$  respectively.

The optimal pose,  $E_{\text{opt}}$ , is given by a correction to each of  $E_1$  and  $E_2$ . Let these correction terms be  $e^{\epsilon_1}$  and  $e^{\epsilon_2}$  respectively, such that

$$E_{\text{opt}} = e^{\epsilon_1} E_1 = e^{\epsilon_2} E_2 \quad (9)$$

and, rearranging,

$$E_2 E_1^{-1} = [e^{\epsilon_2}]^{-1} [e^{\epsilon_1}] = e^{-\epsilon_2} e^{\epsilon_1} \quad (10)$$

Since (10) concerns the exponentiation of matrices, which do not commute under multiplication, the exponents in the final term cannot be simply added. However, they can be approximated as

$$e^{-\epsilon_2} e^{\epsilon_1} = e^{(\epsilon_1 - \epsilon_2) + \frac{1}{2}[\epsilon_1, \epsilon_2] + O(\epsilon_i^3)} \quad (11)$$

where  $[a, b]$  is the Lie bracket [17]. Since the  $\epsilon_i$  are small, a linear solution can initially be obtained by

$$\epsilon_1 - \epsilon_2 = \ln(E_2 E_1^{-1}) \quad (12)$$

The correction terms  $\epsilon_1$  and  $\epsilon_2$  are drawn from zero-mean Gaussian distributions with covariances  $C_1$  and  $C_2$  respectively. The MAP solution can thus be obtained by maximising the probabilities of these two variables subject to the constraint (12), giving

$$\epsilon_1 = C_1 [C_1 + C_2]^{-1} \ln(E_2 E_1^{-1}) \quad (13)$$

$$\epsilon_2 = -C_2 [C_1 + C_2]^{-1} \ln(E_2 E_1^{-1}) \quad (14)$$

However, since an approximation was involved in step (11), in practice

$$e^{\epsilon_1} E_1 \neq e^{\epsilon_2} E_2$$

although they are usually very close.

Using the estimate given by using (13) in (9) is acceptable for many applications, but if the true optimal solution is required, some iteration is needed. In this case, it is best to return to solutions in the form of (12). For some guess of the optimal pose  $E_{\text{opt}}^k$  (which may be initialised, for example, to  $E_{\text{opt}}^0 = E_1$ ), the distance to each of the two estimates can be written as

$$\mathbf{d}_1^k = \ln(E_1 [E_{\text{opt}}^k]^{-1}) \quad (15)$$

$$\mathbf{d}_2^k = \ln(E_2 [E_{\text{opt}}^k]^{-1}) \quad (16)$$

The correction to the current  $E_{\text{opt}}^k$  is given by finding the state vector  $\mathbf{d}^k$  which optimally divides the error between the two  $\mathbf{d}_i^k$ , which is given by the usual equations for combining two Gaussians,

$$\mathbf{d}^k = C_1 (C_1 + C_2)^{-1} \mathbf{d}_1^k + C_2 (C_1 + C_2)^{-1} \mathbf{d}_2^k \quad (17)$$

and then

$$E_{\text{opt}}^{k+1} = e^{\mathbf{d}^k} E_{\text{opt}}^k \quad (18)$$

The equations (15)–(18) can be iterated until convergence, which takes very few iterations. The covariance of this estimate is given by

$$C_{\text{opt}} = C_1 [C_1 + C_2]^{-1} C_2 \quad (19)$$

which is a simple rearrangement of the standard expression; in this form, most of this expression is also used in the calculation of (13) or (17).

### 3. Combining motion and pose information for optimal pose estimation

The formulation developed in Section 2 provides the means to solve the trajectory estimation problem discussed in the introduction. In this Section the solution to the simple case of Figure 2 will first be described, and then the extension to the general case.

Figure 2 shows the case where estimates for the absolute pose of the object are known only at the start and end of the path; these are  $V_0$  and  $V_3$ , with related covariance matrices  $\Phi_0$  and  $\Phi_3$ , such that

$$\hat{V}_i = e^{\mathbf{v}_i} V_i \quad \mathbf{v}_i \sim \mathcal{N}[0, \Phi_i] \quad (20)$$

In contrast, motion information  $M_i$  is available continuously, in each sample point's local co-ordinate frame. For motions, it is more convenient to treat the uncertainty symmetrically, with some before and some after the motion:

$$\hat{M}_i = e^{\epsilon_i} M_i e^{\epsilon'_i} \quad \epsilon_i, \epsilon'_i \sim \mathcal{N}[0, \Sigma_i] \quad (21)$$

As discussed in Section 1, the global estimation and optimisation of all the poses and covariances is an immense problem—there are  $4 \times 3 = 12$  degrees of freedom in the example of Figure 2 alone, and about 5,400 in the case considered in the trajectories of Section 4.2. However, the theory developed in Section 2 enables a fast and elegant solution. The pose at all the positions along the trajectory can be calculated by working along the chain of positions, steadily propagating and combining the poses and covariances.

#### 3.1. Propagating pose and covariance estimates

The pose and uncertainty at position  $\hat{E}_1$  can easily be estimated by propagating forwards from the known information about  $\hat{V}_0$ :

$$\hat{E}_1^+ = \hat{M}_0 \hat{V}_0 \quad (22)$$

where the superscript  $+$  indicates a forward-propagated estimate. Substituting with the uncertainties from (20) and (21) gives

$$\hat{E}_1^+ = e^{\epsilon_0} M_0 e^{\epsilon'_0} e^{\mathbf{v}_0} V_0 \quad (23)$$

The adjoint representation (8) can be used to correctly gather all of the uncertainty to the left-hand side, so that it can be written in the usual form  $\hat{E}_1^+ = e^{\mathbf{x}_1^+} E_1^+$ , with  $\mathbf{x}_1^+ \sim \mathcal{N}[0, C_1^+]$ . This gives

$$E_1^+ = M_0 V_0 \quad (24)$$

$$C_1^+ = \Sigma_0 + \text{Ad}(M_0) [\Sigma_0 + \Phi_0] \text{Ad}(M_0)^T \quad (25)$$

A second, and completely independent estimate for this pose,  $\hat{E}_1^-$ , is given by propagating backwards from the other

end of the chain using the inverse motions:

$$\begin{aligned} E_2^- &= M_2^{-1}V_3 \\ C_2^- &= \Sigma_2 + \text{Ad}(M_2^{-1})[\Sigma_2 + \Phi_3] \text{Ad}(M_2^{-1})^T \\ E_1^- &= M_1^{-1}M_2^{-1}V_3 \end{aligned} \quad (26)$$

$$C_1^- = \Sigma_1 + \text{Ad}(M_1^{-1})[\Sigma_1 + C_2^-] \text{Ad}(M_1^{-1})^T \quad (27)$$

These two propagations have combined all of the known information up to this point. All that remains is to optimally combine the two estimates  $\hat{E}_1^+$  and  $\hat{E}_1^-$ . This is exactly the situation described in Section 2.1, and the optimal estimate for  $\hat{E}_1$  can therefore be calculated from (15)–(19) as described earlier.

All of the other poses along the trajectory can be optimised in a similar way—the calculation of the pose  $\hat{E}_2$  is almost exactly the same case as that described above. The positions of the start and end of the chain ( $\hat{E}_0$  and  $\hat{E}_1$ ) must also be optimised. Although pose estimates  $V_i$  are already known at these points, there are also alternative estimates provided by propagating all the way from the other end of the chain. These poses can also be optimised according to Section 2.1, this time combining the propagated information with the absolute pose information.

### 3.2. The general case: including intermediate pose estimates

In a more general case, such as that shown in Figure 1, prior pose estimates may be known at any point along the trajectory, and not necessarily known at the start and end (although there must always be at least one somewhere, to give some absolute pose information). Simple extensions of the propagation scheme described above allow these cases also to be treated.

When no pose  $V_0$  is provided for the start of the chain, forward propagation cannot begin until the first pose  $V_i$  is reached. Any optimised pose estimates before this point are solely provided by backwards propagation. The converse is true at the opposite end of the chain.

When pose estimates  $V_i$  are provided elsewhere along the chain, they are simply optimised into, and included with, the propagation. In both the forward and backward propagation stages, if a sample point is reached which is also provided with a prior pose estimate, these two estimates (the propagated estimate and the prior) are optimally combined according to Section 2.1 and the optimised pose and covariances propagated onwards. There is a subtlety here when it comes to finally combine the forwards- and backwards-propagated estimates, since the vision measurement should only be included once, but this is simply dealt with as outlined in Algorithm 1, introduced in the next Section.

### 3.3. Generating a complete optimal path

The trajectory estimation algorithm developed in this paper provides a fast (linear-time) method for optimally combining continuous motion and sparse pose data, requiring three passes of the data: forward propagation, backward propagation, and an final optimisation step to combine the data from each pass, as summarised in Algorithm 1. The Combine() function referred to in the algorithm is that described in Section 2.1, equations (15)–(19). Forward and backward propagation begins only at the first pose  $V_i$  encountered.

---

#### Algorithm 1 Complete algorithm for trajectory estimation

---

##### Propagate forwards

For each frame  $i = 1 \dots N$

- Calculate  $E_i^+ = M_{i-1}E_{i-1}^+$
- Calculate  $C_i^+ = \Sigma_{i-1} + \text{Ad}(M_{i-1})[\Sigma_{i-1} + C_{i-1}^+] \text{Ad}(M_{i-1})^T$
- If also have pose  $\hat{V}_i$ , set
  - $\{E_i^+, C_i^+\} = \text{Combine}(E_i^+, C_i^+, V_i^+, \Phi_i^+)$  using (15)–(19)

##### Propagate backwards

For each frame  $i = N - 1 \dots 0$

- Calculate  $E_i^- = M_i^{-1}E_{i+1}^-$
- Calculate  $C_i^- = \Sigma_i + \text{Ad}(M_i^{-1})[\Sigma_i + C_{i+1}^-] \text{Ad}(M_i^{-1})^T$
- If also have pose  $\hat{V}_i$ , store
  - $\{E_i^{U-}, C_i^{U-}\} = \{E_i^-, C_i^-\}$  and then set
  - $\{E_i^-, C_i^-\} = \text{Combine}(E_i^-, C_i^-, V_i^-, \Phi_i^-)$

##### Optimise frames

For each frame  $i = 0 \dots N$

- If have  $\hat{E}_i^+$ ,  $\hat{E}_i^-$ , and  $\hat{V}_i$ , then
    - $\{E_i, C_i\} = \text{Combine}(E_i^{U-}, C_i^{U-}, E_i^+, C_i^+)$
    - (since  $\hat{V}_i$  has already been combined into  $\hat{E}_i^+$ )
  - If just have  $\hat{E}_i^+$  and  $\hat{E}_i^-$ , then
    - $\{E_i, C_i\} = \text{Combine}(E_i^-, C_i^-, E_i^+, C_i^+)$
  - If only have one of  $\hat{E}_i^+$  and  $\hat{E}_i^-$  (i.e. at end of chain), then
    - $\{E_i, C_i\} = \{E_i^+, C_i^+\}$  or  $\{E_i^-, C_i^-\}$  as appropriate
- 

## 4. Results

### 4.1. Simulation: steel ruler

The problem of finding the trajectory of a vehicle has a physical analogue in finding the minimum energy configuration of an elastically deformable structure (e.g. a steel ruler). A finite-element model of such structures can be constructed. The motions between the elements then describe the preferred shape of the structure (a straight line in the case of a ruler), and the (negative log) probability of the uncertainties correspond to the deformation energy at each element. External pose estimates correspond to fixing parts of the ruler at different positions in space, and dimensions

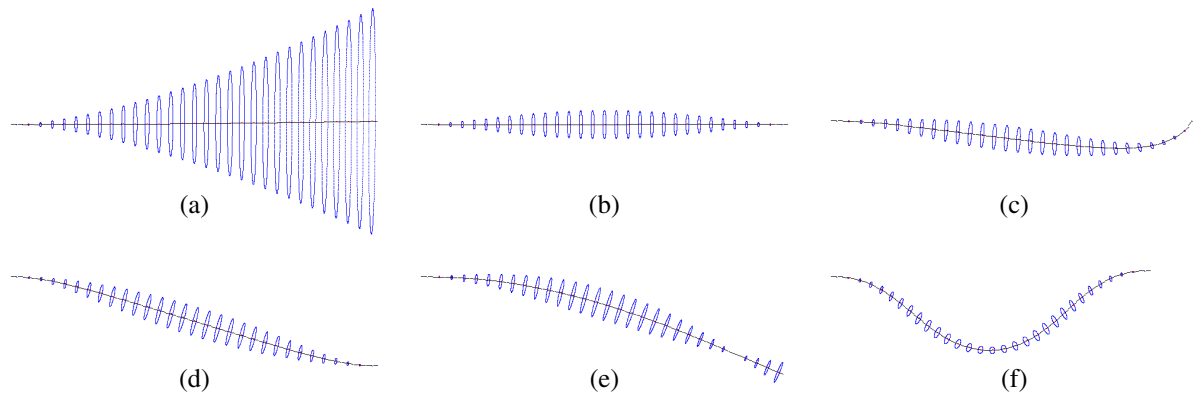


Figure 4: **Trajectory (ruler) shape and covariances** for different test configurations. a) Only one end fixed; b) both ends fixed; c) both ends fixed and one end rotated; d) both ends fixed, and one end moved vertically; e) one end fixed, and a vertical motion with no rotation applied part-way along; f) both ends fixed and moved towards each other.

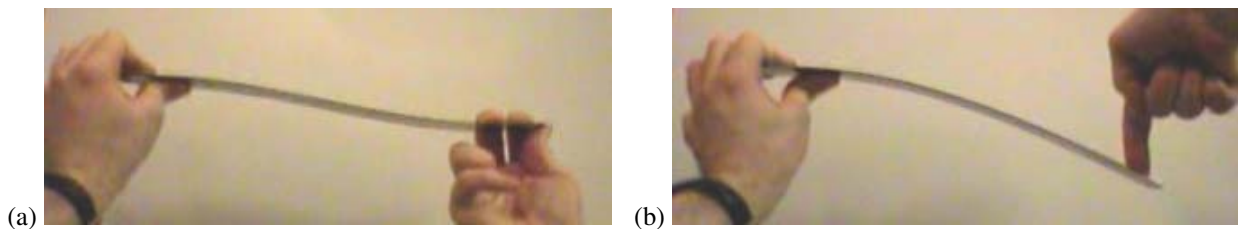


Figure 5: **Shapes adopted by a steel ruler.** a) Both ends fixed, and one end moved vertically; b) one end fixed, and a vertical motion with no rotation applied part-way along. Compare with Figures 4(d) and 4(e).

of large variance in the uncertainties of these estimates correspond to unconstrained degrees of freedom at the fixing point (e.g. a zero torque fixing such as a hinge).

For the steel ruler example, the motion is set to be unit translation in the ruler's local  $x$  direction at each element and the covariance matrix is set to be:

$$\Sigma_i = \begin{bmatrix} 0.005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.005 \end{bmatrix}$$

The effect of combining this with various different absolute pose estimates will be considered.

Figure 4(a) shows the simplest scenario, where the trajectory (ruler) is fixed at one end only. This is achieved by adding into the optimisation a single fixed pose  $V_0$  with very small associated covariances. In Figure 4(b), fixing both end points greatly reduces the uncertainty, which is now greatest in the centre of the trajectory, as expected.

The effect of forcing the trajectory into shapes contrary to that predicted by the motion is seen in Figures 4(c)-(e), showing, in order, a rotation of an end point, a vertical shift in an end-point, and a point where the position is fixed, but not the orientation (achieved by setting low variances on the translation components of the pose, but high variances on the orientation components). All three appear exactly as expected; the second two of these can be compared with tests on a real steel ruler, Figure 5.

Figure 4(f) shows a pleasing result, considering the case where the two end points are brought towards each other. In an ideal case, the symmetry of the problem means that a straight line still would result. Here, however, a very small amount of noise has been added into the motion vectors  $M_i$  to break the symmetry. Iteration is then performed over the whole chain and, as expected, it very quickly converges to a solution with a bulge to one side or the other.

## 4.2. Go-kart trajectory estimation

A significant application of the theory developed in this paper is in fusing the large amounts of data which come when trying to determine accurate trajectories for (or simply to localise) moving vehicles. A system has been developed for determining the trajectory of a go-kart over a lap of a racing circuit, using a combination of odometry data and vision. Odometry (and inertial) sensors are reliable over the very short term, but their cumulative errors are substantial over time. A visual system for pose estimation provides the ideal complement to inertial sensors since it gives highly accurate, but fragile, absolute pose estimation.

**Experimental configuration.** Odometry data was provided by wheel-speed and steering-angle sensors on the go-kart. To add to the difficulty of the problem, testing was carried out on a wet track, where the speed  $v_i$  and angle  $\theta_i$

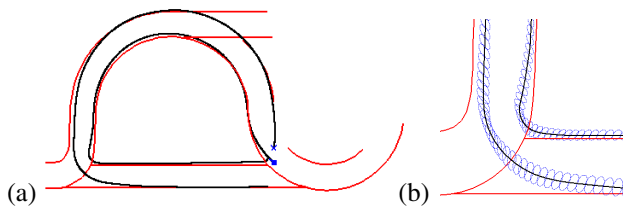


Figure 6: **Go-kart trajectories overlaid on track map, estimated from odometry alone.** a) Complete lap; b) Detail of bottom-left corner, showing 1-standard-deviation covariance ellipses. Note the apparent cutting of the corner in (b) (compare with Figure 9).

of the wheels bears only a limited relationship to the speed and direction of the kart, and there was also some amount of sideways slip  $s_i$ . The sensors provided a continuous feed allowing the motion matrix and uncertainty  $\hat{M}_i$  to be determined between sample points. The motion parameter vector and covariance matrix are given by (using the group SE(2) since the race track is planar)

$$\alpha_i = \begin{bmatrix} -v_i T & 0 & -\frac{v_i T \tan \theta_i}{L} \end{bmatrix} \quad (28)$$

$$\Sigma_i = \begin{bmatrix} \sigma_v^2 & 0 & \theta_i \sigma_v^2 \\ 0 & \sigma_s^2 & 0 \\ \theta_i \sigma_v^2 & 0 & v_i^2 \sigma_\theta^2 + \theta_i^2 \sigma_v^2 \end{bmatrix} \quad (29)$$

where  $T$  is the sampling period (20 ms),  $L$  the kart length, and  $\sigma_v$ ,  $\sigma_\theta$  and  $\sigma_s$  are the standard deviations in speed, angle and slip respectively. The first two standard deviations are estimated from the process noise in sections of constant speed and steering input, and it is assumed that  $\sigma_s = \sigma_v$ . The trajectory, at this stage based purely on the odometry data, can then be generated according to the algorithm of Table 1 given the known starting points.

The odometry alone, with suitable manual tweaking of both linear and non-linear calibration parameters, give trajectories which appear reasonable at first glance, shown overlaid on the track map in Figure 6(a). The driver performed two laps of this short loop, one following the outside of the track, and one following the inside. On closer inspection, however, it can be seen that the error is as much as one metre in places, and the fusion of another data source—vision in this case—is essential to produce an accurate trajectory. Video data is provided by a calibrated video camera mounted on the kart.

**Fusion of odometry and vision.** Figure 7 shows a view from the kart's video camera. The track has been accurately surveyed using GPS, and from this a model of the track ribbon has been created. The camera is calibrated, which enables this model to be rendered in the video frame as seen from points on the estimated trajectory. Overlaid on Figure 7(a) is the view of the model from the pose estimated for that particular video frame. It can be seen that the estimate

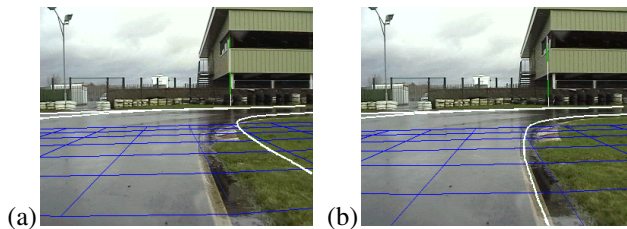


Figure 7: **Camera views from the go-kart, with track model overlaid.** a) Using pose estimated from odometry alone (see Figure 6)—note that it is in error by about 50cm; b) pose adjusted using visual tracking until model and video are aligned. This is the pose estimate  $\hat{V}_i$  added into the optimisation at this point.

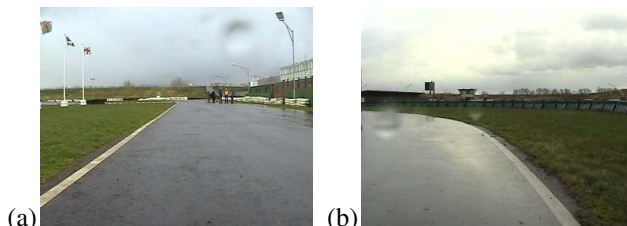


Figure 8: **Ambiguous camera views.** Sample frames from the curved and straight Sections of the test loop. Both cases provide high positional accuracy laterally, but high variance along the direction of travel.

has the camera (and the kart) about 50cm further away from the edge of the track than it really is. This pose can be used as an initialisation for a modified version of the visual tracking scheme outlined in [5], which adjusts the pose until the model edges align with edges in the image (Figure 7(b)). This gives the absolute position of the camera at this frame  $V_i$ , together with an error covariance  $\Phi_i$  which is estimated from the residuals provided by this tracking system. This data is then added to the optimisation.

The vision system can give accurate pose estimation (to within a few centimetres), but only when there are sufficient edge features to resolve all the degrees of freedom. This is the case in Figure 7, but Figure 8 shows some other video frames from the lap where there is significant pose uncertainty in the direction of travel. These can still be used in the optimisation, but will only constrain the trajectory laterally. In many cases, the vision and odometry prove to be complementary sensors, and the resultant accuracy is far higher than from either sensor alone.

**Unknown synchronisation.** The odometry and video data were gathered independently, i.e. with no synchronisation. This adds additional uncertainty into the combination of the two data sources, but this can also be accounted for in this framework. Temporal uncertainty can be handled by adding pose uncertainty into the vision measurement, where the uncertainty corresponds to motion along the local trajectory (which may be a curve). This direction is given by the vector  $\alpha_i$ , and the uncertainty can thus be expressed by

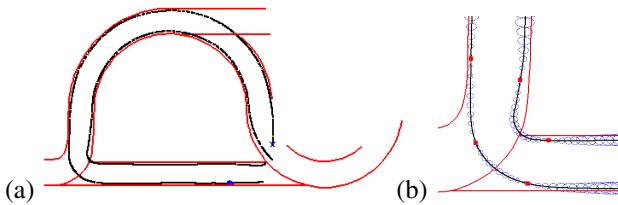


Figure 9: Go-kart trajectories overlaid on track map, using optimally combined odometry and vision (cf. Figure 6). a) Complete lap; b) Detail of bottom-left corner. The locations of the five vision measurements are marked and, since the estimated errors are so small, 50-standard-deviation standard ellipses are shown

$\alpha_i \alpha_i^T$ . The total uncertainty used for the vision measurement in the optimisation is therefore

$$C'_i = C_i + \lambda \alpha_i \alpha_i^T \quad (30)$$

where  $C_i$  is extracted from the tracking system, as described above, and  $\lambda$  is one standard deviation of the number of frames by which the synchronisation can be wrong ( $\lambda = 4$  in this case).

**Final paths.** Figure 6(b) shows detail of the bottom-left of the trajectory (corresponding to the location of the video frame of Figure 7) before fusion with video data, and it can be seen that the trajectory is clearly wrong in places, with the kart shown to be distant from, or crossing, the track edge at times. Figure 9(b) shows the effect of adding just five vision measurements, resulting in a trajectory which agrees very well with the ground truth seen through the video camera. Figure 9(a) shows the complete pair of laps after vision measurements have been applied at intervals around the whole lap. Each lap consists of approximately 1,800 data points and associated covariances, together with about 30 vision measurements. The algorithm outlined in this paper optimises each path in approximately 150 mS on a standard 1.5GHz PC.

## 5. Conclusions

This paper has shown how the exponential map can be used to represent probability distributions over a group, and how these can be correctly propagated along a trajectory. A fast algorithm is described which calculates the maximum *a posteriori* trajectory given continuous differential (odometry or inertial) data, and sparse unsynchronised positional data from visual localisation.

This algorithm provides a good initialisation for solving the structure from motion problem where some structure (in this case the white lines) is already known, and where the remaining structure in the scene is to be recovered. Future work will consider integrating the techniques developed in this paper into the full structure from motion problem.

**Acknowledgements.** This work was funded by Pi Group Ltd.

## References

- [1] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 8–15, Santa Barbara, CA, 1998.
- [2] F. Chenaview and J. L. Crowley. Position estimation for a mobile robot using vision and odometry. In *Proc. Int. Conf. Robotics and Automation*, pages 2588–2593, Nice, France, 1992.
- [3] A. J. Davidson and N. Kita. Sequential localisation and map-building for real-time computer vision and robotics. *Robotics and Autonomous Systems*, 36:171–183, 2001.
- [4] M. Deans and M. Herbert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Proc. Int. Symposium on Experimental Robotics*, pages 395–404, Waikiki, HI, USA, 2000.
- [5] T. Drummond and R. Cipolla. Application of Lie algebras to visual servoing. *Int. J. Computer Vision*, 37(1):21–41, 2000.
- [6] H. Durrant-Whyte. Where am I? A tutorial on mobile vehicle localization. *Industrial Robot*, 21(2):11–16, 1994.
- [7] O. Faugeras, Q. T. Luong, and T. Papadopoulos. *The Geometry of Multiple Images*. MIT Press, Cambridge, MA, USA, 2001.
- [8] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conf. Computer Vision*, volume I, pages 311–326, Freiburg, Germany, 1998.
- [9] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, 1974.
- [10] J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robotics and Automation*, 17(3):242–257, 2001.
- [11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2001.
- [12] P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Trans. Robotics and Automation*, 17(5):748–760, 2001.
- [13] S.-H. Jung and C. J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 732–737, Kauai, HI, USA, 2001.
- [14] S. I. Roumeliotis and G. A. Bekey. Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization. In *Proc. Int. Conf. Robotics and Automation*, pages 2985–2992, San Francisco, CA, USA, 2000.
- [15] D. Stenlow and S. Singh. Optimal motion estimation from visual and inertial measurements. In *IEEE Workshop Applications of Computer Vision*, pages 314–319, Orlando, FL, USA, 2002.
- [16] W. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Proc. Int. Workshop Vision Algorithms*, pages 298–375, Kerkyra, Greece, 1999.
- [17] V. S. Varadarajan. *Lie Groups, Lie Algebras, and their Representations*. Springer-Verlag, New York, 1984.