Train positioning using video Odometry

Problem presented by

Richard Shenton

Reliable Data Systems

ESGI100 was jointly hosted by
Smith Institute for Industrial Mathematics and System Engineering
and the University of Oxford

with additional financial support from
Engineering and Physical Sciences Research Council
European Journal of Applied Mathematics
Oxford Centre for Collaborative Applied Mathematics
Warwick Complexity Centre

Report author

Artur Gower (National University of Ireland Galway)
Dr. Robert Whittaker (University of East Anglia)
Megan Davies Wykes (University of Cambridge)
Dr. Jacqueline Christmas (University of Exeter)
Dr. Phil Browne (University of Reading)
Dr. Jan Van lent (University of West of England)
Robert Gower (University of Edinburgh)
Dr. Siddhartha Ghosh (Smith Institute)

Executive Summary

Reliable Data Systems have developed a video-based odometry system that enables trains to measure velocities and distances travelled without the need for trackside infrastructure. The Study Group was asked to investigate ways of improving the accuracy of such a system, and to suggest any improvements that might be made. The work performed in the week followed along these strands: (a). Elimination of errors in video odometery induced by pitch and height; (b) Robust calculation of (i) the train velocity and (ii) the track curvature; (c). Accurate determination of the position of a train on a track by assimilating Curvature information; (d). Determining where on UK's railway map a train journey takes place, based purely on video odometry and (e). Drawing a track map.

Version 1.0
July 24, 2014
iv+59 pages

Contributors

Artur Gower (National University of Ireland Galway)
Dr. Robert Whittaker (University of East Anglia)
Megan Davies Wykes (University of Cambridge)
Dr. Jacqueline Christmas (University of Exeter)
Dr. Phil Browne (University of Reading)
Dr. Jan Van lent (University of West of England)
Sidsel Sørensen (Technical University of Denmark)
Prof. Chris Budd (University of Bath)
Dr. David Wood (University of Warwick)
Jakub Tomczyk (University of Sydney)
Jakub Nowotarski (Wrocaw University of Technology)
Dr. Giorgi Khimshiashvili (Ilia State University)
Dr. Wasyl Kowalczuk (Centre for Industrial Applications of Mathematics and Systems Engineering)
Dr. Hiroyuki Ochiai (Kyushu University)
David O'Connor (University of Warwick)
Michał Radziwon (University of Southern Denmark)
Dr. Armindo Costa (University of Warwick)
Radu Cimpeanu (Imperial College London)
Robert Gower (University of Edinburgh)
Ana Azevedo Neves (Oxford Brookes University)
Alexander Vervuurt (University of Oxford)
Dr. Siddhartha Ghosh (Smith Institute)

Contents

1 Introduction

1.1 Background

(1.1) There is an increasing need for railways to increase capacity within the network. Installation of new signaling systems is an important step in capacity improvement. More importantly, capacity improvement require better train positioning. Conventional systems that rely on trackside infrastructure are expensive to install and maintain. For systems on the train, there may be limitations on performance in winter conditions, for example due to wheel slip for devices that measure wheel rotations. Inertial methods are expensive. There has been much work done on satellite positioning, but visibility is not consistent (for example in tunnels) and so a secondary sensor system is needed to provide coverage during those periods. The video system developed by RDS appears to overcome all these problems.

(1.2) RDS have developed a video-based odometry system that enables a train to measure distances travelled, using a forward-facing camera mounted in the cab. Such a system can report train positions via a radio data link in real time to the signalling control centre, which in turn can provide information to the train on braking points, etc. The benefits of the video system are in terms of lower costs and higher accuracy of positioning. In the longer term, there are possibilities for allowing closer separation of trains, leading to higher capacity of the rail network.

(1.3) The overview operation of the video-based system is as follows. The camera mounted in the cab images the track immediately ahead of the train, generally at a frame rate in the range 25 to 50 frames per second. Each image is “unwarped”, to provide a plan view as if viewed from directly above the track. The unwarped images from successive frames are matched by looking at pixel blocks, to build up an “optical flow” from one image to the next. This flow provides an estimate of the distance moved between frames.

(1.4) The RDS video odometer measures forward and sideways displacement. The latter is related to the track curvature. From this information it is possible to estimate the 2D movement of the train. However a key challenge in this regard is to get better estimates of the position that RDS are currently obtaining. RDS mentioned that one area for investigation is the “cant” of the track, i.e. tracks are slightly banked to improve cornering performance. Consequently, the assumption that the motion of a train is planar may not hold.

(1.5) The video odometer measures displacement from a known point. To obtain current position from the odometry, one or more “known” prior points are required. Currently it needs an additional input (e.g GPS location) for this initialization. However, GPS is not always available (and it does not distinguish between adjacent tracks reliably). As one has an estimate of track curvature for each journey, and the actual track curvature is invariant, it is possible that to use track curvature "signatures" to determine train position.

(1.6) In a previous Study-Group report Shenton et al. (2008), it was considered how errors in the camera positioning (e.g. caused by vibrations or suspension motions in the train) would affect the relative displacement between two neighbouring images recorded by the system. The work performed previously followed three strands:

1. an understanding of how deviations from the cameras calibrated position lead to errors in the trains calculated position and velocity;
2. development of models for the train suspension, designed to place bounds on these deviations; and
3. the performance of the associated image processing algorithms.

1.2 Problem statement

(1.7) Richard Shenton posed the following problems to the study group that seek to extend the existing capabilities of RDS's video odometry system, building on the efforts of the previous study group:

1. Elimination of errors in video odometery induced by pitch and height;
2. Robust calculation of (i) the train velocity and (ii) the track curvature;
3. Accurate determination of the position of a train on a track by assimilating Curvature information and
4. Determining where on UK's railway map, a particular journey takes place, based purely on video odometry and
5. Building a track map from information gathered through camera and GPS on routine train runs.

1.3 Data Available

(1.8) The following data were made available for the purposes of addressing the problems described previously:

1. Data on forward and sideways displacement captured by the odometry system from selected railway journeys. This included: (a). a return journey from Dorking to Leatherhead in the UK and (b). seven independent runs over the same 4.2km length of track from New Alresford to Ropley in Hampshire. We refer to the latter as the Mid-Hants dataset.

2. UK railway map of tracks obtained from OpenStreetMap (`www.openstreetmap.org`), including GPS coordinates and
3. Link-node diagram of the track changes.

2 Design guides

2.1 Definitions

(2.1) The geometry of a line of track can be described parametrically by its horizontal position (x, y) and height z as a function of an along-track parameter. From these functions, we may derive the local (horizontal) curvature κ and gradient angle δ. In addition track may be banked to aid cornering, which leads to a camber angle ϵ. See Figure 1.

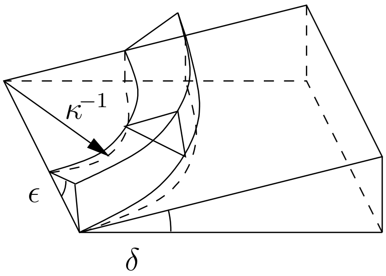


Figure 1: The geometry of the track, showing the horizontal curvature κ, the gradient δ, and the camber angle ϵ.

(2.2) We define the horizontal curvature κ to be the curvature on a 2D map projection, i.e. the rate of the change of horizontal direction per unit horizontal distance travelled. So if the parametric representation of the track is $(x(t), y(t), z(t))$, then

$$\kappa = \frac{\ddot{x}\dot{y} - \dot{x}\ddot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \,. \tag{1}$$

We define the track gradient angle as the angle made by the track centre line above or below the horizontal, so

$$\delta = \sin^{-1}\left(\frac{\dot{z}^2}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{1/2}}\right) = \tan^{-1}\left(\frac{\dot{z}^2}{(\dot{x}^2 + \dot{y}^2)^{1/2}}\right) \,. \tag{2}$$

We define the camber angle ϵ to be the deviation that the cross-section of the track makes to a horizontal line in the plane normal to the rails.

In practice the gradient of the track is small, so to good approximation, κ is the same as the rate of change of direction per unit track distance, and the camber angle is the same as the angle measured in a vertical plane.

Parameter	Symbol	Estimator	Max. value	Ref. in Safety and Limited (2011)
Gradient angle	δ	$\sin^{-1}(g_z)$	3.5×10^{-2} rad	§2.7.4
Camber angle	ϵ	$\sin^{-1}(c/d)$	1.3×10^{-1} rad	§2.5.4
Horizontal curvature	κ	R^{-1}	8.0×10^{-3} m^{-1}	§2.5.1
Vertical curvature	λ	R^{-1}	1.7×10^{-3} m^{-1}	§2.7.7
Torsion	τ	$g_c/(d\cos\epsilon)$	1.7×10^{-3} m^{-1}	§2.6.4

Table 1: Constraints on the track geometry parameters $(\kappa, \delta, \epsilon, \lambda, \tau)$ from Safety and Limited (2011), where R is the radius of curvature, $d = 1435$ mm is the track spacing, c is the cant distance, $g_z = \mathrm{d}z/\mathrm{d}s$ is the track gradient, and $g_c = \mathrm{d}c/\mathrm{d}s$ is the cant gradient.

We also introduce the vertical curvature λ and torsion τ defined by

$$\lambda = \frac{\mathrm{d}\delta}{\mathrm{d}s}, \qquad \tau = \frac{\mathrm{d}\epsilon}{\mathrm{d}s}, \tag{3}$$

where s is the distance along the track.

2.2 Constraints

(2.3) Railway Group Standard GC/RT5021 "Track System Requirements" Safety and Limited (2011) provides constraints on the maximum gradients and curvatures of the track. These can be used to derive constraints on the the track geometry parameters $(\kappa, \delta, \epsilon, \lambda, \tau)$. These constraints are shown in table 1.

2.3 Instantaneous train rotation

(2.4) The geometry of the track and the motion of the train along it causes the train to have an instantaneous angular velocity relative to the ground, which will not always be zero. This angular velocity, as measured in the frame of the train, will be added to the forward motion of the train to determine how neighbouring camera images are related.

The instantaneous angular velocity caused by the bulk train motion can be described in terms of three curvatures (α, β, γ) about three coordinate axes relative to the train. The three components of the angular velocity are then given by $(\alpha V, \beta V, \gamma V)$ where $V = \mathrm{d}s/\mathrm{d}t$ is the velocity of the train. Alternatively, the incremental rotations of the camera as the train moves forward a distance $\mathrm{d}s$ are given by $(\alpha \mathrm{d}s, \beta \mathrm{d}s, \gamma \mathrm{d}s)$. See Figure 2.

(2.5) We must now find the curvatures (α, β, γ) in terms of the track geometry parameters $(\kappa, \delta, \epsilon, \lambda, \tau)$. We do this by equating two expressions for the incremental rotation matrix $\mathcal{R}$ for the rotation that the train undergoes after travelling a distance $\mathrm{d}s$ along the track.

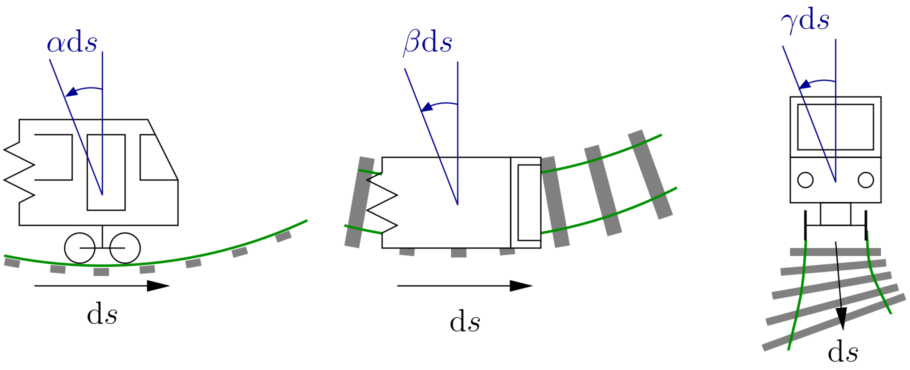


Figure 2: How the three curvatures (α, β, γ) in the frame of the train give rise to rotations of the train body (in addition to the main translation) as the train moves forwards along the track.

We use coordinates aligned with the train at its initial position, so that the x axis is forward along the track, and the z axis is normal to the plane of the track. The y axis then completes the orthogonal set, and thus lies in the plane of the track at right angles to the direction of travel.

We define the fundamental rotation matrices about the three coordinate axes:

$$\mathcal{R}_1(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}, \tag{4}$$

$$\mathcal{R}_2(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{pmatrix}, \tag{5}$$

$$\mathcal{R}_3(\phi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

In terms of the rotations in the frame of the train, we have

$$\mathcal{R} = \mathcal{R}_1(\gamma \mathrm{d}s) \cdot \mathcal{R}_2(-\alpha \mathrm{d}s) \cdot \mathcal{R}_3(\beta \mathrm{d}s). \tag{7}$$

Expanding the rotation matrices for infinitesimal $\mathrm{d}s$ and completing the matrix multiplication, we obtain

$$\mathcal{R} = \mathcal{I} + \mathcal{R}' \mathrm{d}s, \qquad \text{where} \qquad \mathcal{R}' = \begin{pmatrix} 0 & -\beta & -\alpha \\ \beta & 0 & -\gamma \\ \alpha & \gamma & 0 \end{pmatrix} \tag{8}$$

and $\mathcal{I}$ is the identity matrix. It is apparent that changing the order of the three rotations in (7) does not alter this result.

(2.6) We can also compute $\mathcal{R}$ from the track geometry, as follows. We first rotate the train to a level horizontal plane by undoing the camber ϵ, and the gradient δ.

We then apply a rotation of $\kappa \mathrm{d}s$ to account for the horizontal track curvature. Finally,we re-apply the new gradient and new camber, which are given by $\delta + \lambda \mathrm{d}s$ and $\epsilon + \tau \mathrm{d}s$ respectively. This yields

$$\mathcal{R} = \mathcal{R}_1(\epsilon + \tau \mathrm{d}s) \cdot \mathcal{R}_2(-(\delta + \lambda \mathrm{d}s)) \cdot \mathcal{R}_3(\kappa \mathrm{d}s) \cdot \mathcal{R}_2(\delta) \cdot \mathcal{R}_1(-\epsilon)\,. \tag{9}$$

Multiplying out the matrices and expanding for infinitesimal $\mathrm{d}s$ we obtain

$$\mathcal{R} = \mathcal{I} + \mathcal{R}'\mathrm{d}s \tag{10}$$

where

$$\mathcal{R}' = \begin{pmatrix} 0 & -\kappa\cos\delta\cos\epsilon + \lambda\sin\epsilon & -\kappa\cos\delta\sin\epsilon - \lambda\cos\epsilon \\ \kappa\cos\delta\cos\epsilon - \lambda\sin\epsilon & 0 & \kappa\sin\delta - \tau \\ \kappa\cos\delta + \lambda\cos\epsilon & -\kappa\sin\delta + \tau & 0 \end{pmatrix}. \tag{11}$$

Comparing equations (8) and (11), we find that

$$\begin{aligned} \alpha &= \lambda\cos\epsilon + \kappa\cos\delta\sin\epsilon\,, && (12) \\ \beta &= \kappa\cos\delta\cos\epsilon - \lambda\sin\epsilon\,, && (13) \\ \gamma &= \tau - \kappa\sin\delta\,. && (14) \end{aligned}$$

When $\delta = \epsilon = 0$ we have that $\alpha = \lambda$, $\beta = \kappa$, and $\gamma = \tau$. But when δ and/or ϵ are non-zero (i.e. the track is banked or on a gradient) then the different track curvatures have a more complicated effect on the train motion.

2.4 Constraints on α, β, γ

(2.7) Using the expressions in (12)–(14) and the constraints in table 1, we can derive constraints for α, β and γ and determine which terms dominate in the expressions.

We observe that the angles ϵ and δ are both small, and that the horizontal curvature κ is slightly larger than the the vertical curvature λ and the torsion τ. Estimating the terms, we find

- The two terms in the normal curvature α are likely to have similar magnitudes
$$\alpha \approx \lambda + \kappa\epsilon \lesssim 1.7 \times 10^{-3}\ \mathrm{m}^{-1}\,. \tag{15}$$
(So there is as much normal curvature from the interaction of cant with horizontal curvature, as comes directly from vertical curvature.)
- The in-plane curvature β is likely to be dominated by the horizontal curvature κ, and we have
$$\beta \approx \kappa \lesssim 8 \times 10^{-3}\ \mathrm{m}^{-1}\,. \tag{16}$$

- The two terms in the torsion γ are likely to be of similar magnitude, so we have
$$\gamma \approx \tau - \kappa\delta \lesssim 2.4 \times 10^{-3}\ \mathrm{m}^{-1}\,. \tag{17}$$
(So there is as much torsion from interaction of the gradient with horizontal curvature, as comes directly from the cant gradient.)

3 Elimination of distortion due to curvature motion

3.1 Overview and setup

(3.1) We now perform some similar calculations to the previous Study-Group report Shenton et al. (2008) for the case of the rotations caused by the track curvature. In all cases we adopt a coordinate system fixed relative to the train camera, with X being the distance forward along the track, Y being the distance sideways in the plain of the track, and Z being the normal distance upwards from the plane of the track. The camera on the train is located at $(X, Y, Z) = (0, 0, H_0)$ a distance H_0 above the track.

We consider a general point $\boldsymbol{X}_1 = (X_1, Y_1, Z_1)$ on the track ahead of the train, and consider the point $\boldsymbol{X}_2 = (X_2, Y_2, Z_2)$ that an object at that point will appear after a time δt when the train has moved forward a distance $\delta s = V\delta t$ along the track. These two points are then converted into observed positions on the unwarped camera image, in order to compute the offset that will be seen between at two images.

From Shenton et al. (2008), a point at (X, Y, Z) will appear on the unwarped camera image at a point (X^*, Y^*) given by
$$\begin{pmatrix} X^* \\ Y^* \end{pmatrix} = \frac{H_0}{H_0 - Z}\begin{pmatrix} X \\ Y \end{pmatrix} \approx \left(1 + \frac{Z}{H_0}\right)\begin{pmatrix} X \\ Y \end{pmatrix}, \tag{18}$$
where the approximation has come from expanding for $Z/H_0 \ll 1$ (i.e. the track level always remains much below the camera height in the region of interest). Considering the two points $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, the horizontal and vertical displacements in the unwarped camera image will be given by
$$X_2^* - X_1^* = X_2\left(1 + \frac{Z_2}{H_0}\right) - X_1\left(1 + \frac{Z_1}{H_0}\right), \tag{19}$$
$$Y_2^* - Y_1^* = Y_2\left(1 + \frac{Z_2}{H_0}\right) - Y_1\left(1 + \frac{Z_1}{H_0}\right). \tag{20}$$

For later convenience, we define the differences
$$\delta X = X_2 - X_1\,, \qquad \delta Y = Y_2 - Y_1\,, \tag{21}$$

and the mean values

$$\bar{X} = \frac{1}{2}(X_2 + X_1)\,, \qquad \bar{Y} = \frac{1}{2}(Y_2 + Y_1)\,, \tag{22}$$

of the two object positions. To leading order $\bar{X}$ and $\bar{Y}$ are equal to the equivalent quantities $\bar{X}^*$ and $\bar{Y}^*$ in the unwarped image.

3.2 Normal curvature α

(3.2) The curvature here is normal to the plane of the track. For small curvatures we can approximate the plane of the track by $Z = \frac{1}{2}\alpha X^2$. See Figure 3.

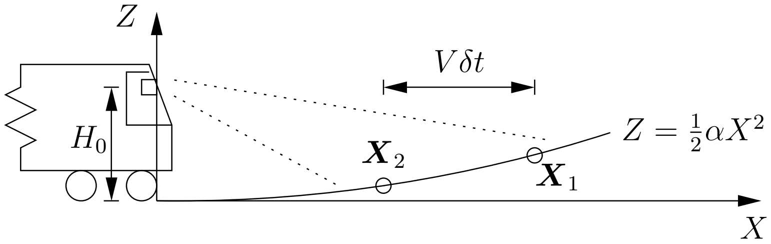


Figure 3: The geometry of the track with a normal curvature α, and the positions $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ of an object on the before and after a time interval δt.

The displacements of the object are given by

$$\delta X \equiv X_2 - X_1 = -V\delta t\,, \qquad \delta Y \equiv Y_2 - Y_1 = 0\,. \tag{23}$$

From (19) and (20), the difference in the observed positions of these points is then give by

$$\begin{aligned} X_2^* - X_1^* &= X_2\left(1 + \frac{\alpha}{2H_0}X_2^2\right) - X_1\left(1 + \frac{\alpha}{2H_0}X_1^2\right) && (24)\\ &= (X_2 - X_1) + \frac{\alpha}{2H_0}\left(X_2^3 - X_1^3\right) && (25)\\ &\approx \delta X + \frac{3\alpha\bar{X}^2}{2H_0}\delta X && (26)\\ &= -V\delta t - \frac{3\alpha\bar{X}^2}{2H_0}V\delta t\,, && (27) \end{aligned}$$

and

$$
\begin{aligned}
Y_2^* - Y_1^* &= Y_2\left(1 + \frac{\alpha}{2H_0}X_2^2\right) - Y_1\left(1 + \frac{\alpha}{2H_0}X_1^2\right) && (28)\\
&= (Y_2 - Y_1) + \frac{\alpha}{2H_0}\left(X_2^2 Y_2 - X_1^2 Y_1\right) && (29)\\
&\approx \delta Y + \frac{\alpha \overline{X}^2}{2H_0}\delta Y + \frac{2\alpha \overline{X}\,\overline{Y}}{2H_0}\delta X && (30)\\
&= -\frac{\alpha \overline{X}\,\overline{Y}}{H_0} V \delta t\,, && (31)
\end{aligned}
$$

3.3 In-plane curvature β

(3.3) The track curvature here is purely in the plane of the track, so all points on the track have $Z = 0$. Therefore the observed position (X^*, Y^*) of each point $(X, Y, 0)$ on the track is given simply by

$$
\begin{pmatrix} X^* \\ Y^* \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \qquad (32)
$$

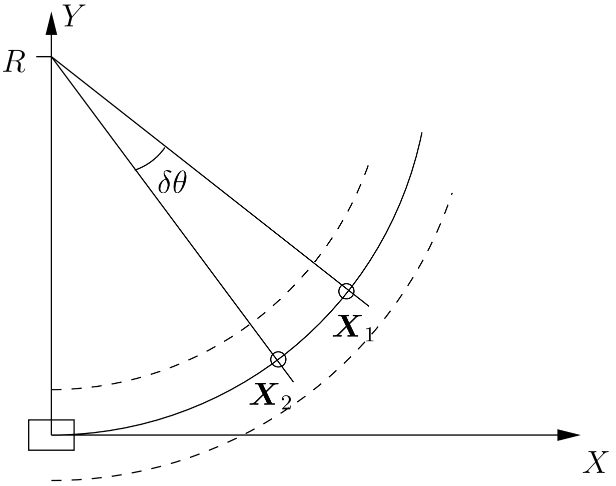


Figure 4: Coordinates (x, y) relative to the camera, and the positions $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ of an object in two neighbouring frames a time δt apart. When the train is instantaneously moving with velocity V on a track with radius of curvature R, the apparent motion of objects is on arcs of circles centred on the centre of curvature. The change $\delta\theta$ in angle is given by $R\delta\theta = V\delta t$.

If the track curvature is β, then the radius of curvature is $R = \beta^{-1}$. The path of an object relative to the train is therefore an arc of a circle centred at $(0, R, 0)$, as shown in Figure 4. In a time δt, the angle of rotation is such that the arc length along the track is $V\delta t$, i.e.

$$
R\delta\theta = V\delta t \quad \Rightarrow \quad \delta\theta = \beta V \delta t\,. \qquad (33)
$$

(3.4) Typical limiting values for the system are $\beta \leq 8 \times 10^{-3}$ m^{-1}, $V \lesssim 50$ ms^{-1}, $\delta t \approx 1/50$ s. So $\delta\theta \lesssim 0.08 \ll 1$. This bound can probably be lowered even further because faster tracks will have larger radii of curvature.

The position vectors of the two locations of the object are related to each other by a rotation of $\delta\theta$ about $(0, R, 0)$:

$$\boldsymbol{X}_2' = \mathcal{R}_3(-\delta\theta) \cdot \boldsymbol{X}_1' \,, \tag{34}$$

where

$$\boldsymbol{X}_1' = \boldsymbol{X}_1 - \begin{pmatrix} 0 \\ R \\ 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{X}_2' = \boldsymbol{X}_2 - \begin{pmatrix} 0 \\ R \\ 0 \end{pmatrix} \tag{35}$$

are the position vectors relative to the centre of the rotation, and

$$\mathcal{R}_3(-\delta\theta) = \begin{pmatrix} \cos(\delta\theta) & \sin(\delta\theta) & 0 \\ -\sin(\delta\theta) & \cos(\delta\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} . \tag{36}$$

Since $\delta\theta$ is small, we may approximate $\cos\delta\theta \approx 1$ and $\sin\delta\theta \approx \delta\theta$. Then, multiplying out the matrix–vector product, we obtain

$$\boldsymbol{X}_2 - \boldsymbol{X}_1 = -V\delta t \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \beta V \delta t \begin{pmatrix} \bar{Y} \\ -\bar{X} \\ 0 \end{pmatrix} \tag{37}$$

Hence we have

$$X_2^* - X_1^* = -V\delta t + \beta V \bar{Y} \delta t \,, \tag{38}$$

and

$$Y_2^* - Y_1^* = -\beta V \bar{X} \delta t \,. \tag{39}$$

3.4 Axial torsion γ

(3.5) The distortion here is a twist in the plane of the track about its tangent. For small torsions we can approximate the surface of the track in the vicinity of the train by $Z = \gamma XY$. See Figure 5.

(3.6) The displacements of an object on the track are given by

$$\delta X \equiv X_2 - X_1 = -V\delta t \,, \qquad \delta Y \equiv Y_2 - Y_1 = 0 \,. \tag{40}$$

From (19) and (20), the difference in the observed positions of these points is then

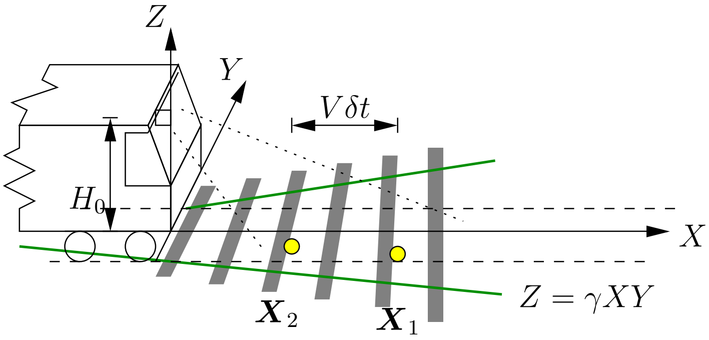


Figure 5: The geometry of the track with an axial torsion γ.

given by

$$\begin{aligned} X_2^* - X_1^* &= X_2\left(1 + \frac{\gamma}{H_0}X_2Y_2\right) - X_1\left(1 + \frac{\gamma}{H_0}X_1Y_1\right) && (41)\\ &= (X_2 - X_1) + \frac{\gamma}{H_0}\left(X_2^2Y_2 - X_1^2Y_1\right) && (42)\\ &\approx \delta X + \frac{2\gamma\overline{X}\overline{Y}}{H_0}\delta X + \frac{\gamma\overline{X}^2}{H_0}\delta Y && (43)\\ &= -V\delta t - \frac{2\gamma\overline{X}\overline{Y}}{H_0}V\delta t\,, && (44) \end{aligned}$$

and

$$\begin{aligned} Y_2^* - Y_1^* &= Y_2\left(1 + \frac{\gamma}{H_0}X_2Y_2\right) - Y_1\left(1 + \frac{\gamma}{H_0}X_1Y_1\right) && (45)\\ &= (Y_2 - Y_1) + \frac{\gamma}{H_0}\left(X_2Y_2^2 - X_1Y_1^2\right) && (46)\\ &\approx \delta Y + \frac{2\gamma\overline{X}\overline{Y}}{H_0}\delta Y + \frac{\gamma\overline{Y}^2}{H_0}\delta X && (47)\\ &= -\frac{\gamma\overline{Y}^2}{H_0}V\delta t\,, && (48) \end{aligned}$$

3.5 Estimating track curvature from the image offsets

(3.7) The basic measure of track distance covered comes from assuming that the average X^* offset is precisely $V\delta t$, and not making use of the information in the Y^* offset or any spatial variations in the offsets across the camera image.

We would like to be able to correct for any errors in the estimated velocity that arise because of the three curvature effects, and also produce an estimate for the in-plane curvature β, as this may provide useful information for the track positioning system.

(3.8) Combining the various offset expressions (assuming that the curvature-induced discrepancies will add linearly) we obtain

$$X_2^* - X_1^* = -V\delta t - \frac{3\alpha \bar{X}^2}{2H_0} V\delta t + \beta \bar{Y} V\delta t - \frac{2\gamma \bar{X}\bar{Y}}{H_0} V\delta t\,, \tag{49}$$

and

$$Y_2^* - Y_1^* = -\frac{\alpha \bar{X}\bar{Y}}{H_0} V\delta t - \beta \bar{X} V\delta t - \frac{\gamma \bar{Y}^2}{H_0} V\delta t\,. \tag{50}$$

We see that the discrepancies from the three different curvatures have different spatial dependencies within the image, so in principle it should be possible to differentiate between them and extract estimates for α, β, γ and V. In practice though, it may be difficult to reliably extract spatial variations given the level of noise in the system.

(3.9) A simple procedure is possible when the unwarped image is centred on the track, as would normally be the case. The the average value of $\bar{Y}^*$ in the unwarped image will be zero. So to good approximation, the average value of $\bar{Y}$ will be zero too. Then, taking equations (49) and (50) and averaging over the image, we obtain

$$\begin{aligned} \langle X_2^* - X_1^* \rangle &= -V\delta t - \frac{3\alpha \langle \bar{X}^{*2} \rangle}{2H_0} V\delta t\,, & (51)\\ \langle Y_2^* - Y_1^* \rangle &= -\beta \langle \bar{X} \rangle V\delta t - \frac{\gamma \langle \bar{Y}^{*2} \rangle}{H_0} V\delta t\,. & (52) \end{aligned}$$

And taking the first moment with respect to $\bar{Y}^*$ we obtain

$$\begin{aligned} \big\langle (X_2^* - X_1^*)\bar{Y}^* \big\rangle &= \beta \langle \bar{Y}^{*2} \rangle V\delta t - \frac{2\gamma \langle \bar{X} \rangle \langle \bar{Y}^{*2} \rangle}{H_0} V\delta t\,, & (53)\\ \big\langle (Y_2^* - Y_1^*)\bar{Y}^* \big\rangle &= -\frac{\alpha \langle \bar{X}^* \rangle \langle \bar{Y}^{*2} \rangle}{H_0} V\delta t\,. & (54) \end{aligned}$$

These can be regarded as four simultaneous equations for α, β, γ and V.

From §2.2 we have

$$\alpha \lesssim 1.7 \times 10^{-3}\ \mathrm{m}^{-1}\,, \qquad \beta \lesssim 8.0 \times 10^{-3}\ \mathrm{m}^{-1}\,, \qquad \gamma \lesssim 2.4 \times 10^{-3}\ \mathrm{m}^{-1}\,. \tag{55}$$

From the system design guide, we also have

$$\langle \bar{X}^* \rangle \approx 10\ \mathrm{m}\,, \qquad \langle \bar{X}^{*2} \rangle \approx 10^2\ \mathrm{m}^2\,, \qquad \langle \bar{Y}^{*2} \rangle \approx 1\ \mathrm{m}^2\,. \tag{56}$$

Comparing the ratios of the terms on the right-hand sides of (51), (52) and (53), we find:

$$\frac{3\alpha \langle \bar{X}^{*2} \rangle}{2H_0} \lesssim 8.5 \times 10^{-2}\,, \qquad \frac{\gamma \langle \bar{Y}^{*2} \rangle}{\beta \langle \bar{X} \rangle H_0} \lesssim 1.0 \times 10^{-2}\,, \qquad \frac{2\gamma \langle \bar{X}^* \rangle}{\beta H_0} \lesssim 2.0\,. \tag{57}$$

So in (51) the α term should be fairly small, in (52) the γ term should be negligible, and in (53) the two terms could be of similar magnitude.

Given these scalings and the likely levels of noise in the system, a proposed set of estimates, aimed at recovering V and β, is as follows:

- Use (54) to obtain an estimate for αV

$$\alpha V = \frac{H_0 \left\langle (Y_2^* - Y_1^*)\bar{Y}^* \right\rangle}{\langle \bar{X}^* \rangle \langle \bar{Y}^{*2} \rangle \delta t} \,. \tag{58}$$

- Use (51) and the estimate for αV to obtain an estimate for V:

$$V = -\frac{\langle X_2^* - X_1^* \rangle}{\delta t} - \frac{3\alpha V \langle \bar{X}^{*2} \rangle}{2H_0} \,. \tag{59}$$

- Finally neglect the small γ term in (52), and use the above estimate for v in order to provide an estimate for β:

$$\beta = -\frac{\langle Y_2^* - Y_1^* \rangle}{\langle \bar{X} \rangle V \delta t} \,. \tag{60}$$

Equation (53) is unused, although it could be employed to provide an estimate for the uninteresting parameter γ.

(3.10) Alternative estimates for β and V can be found by neglecting the α and γ discrepancies altogether. This method will work with camera images that do not necessarily have $\langle \bar{Y}^* \rangle = 0$.

With $\alpha = \gamma = 0$ and averaging over the image, (49) and (50) become

$$\begin{aligned} \langle X_2^* - X_1^* \rangle &= -V\delta t + \beta \langle \bar{Y}^* \rangle V \delta t \,, && (61) \\ \langle Y_2^* - Y_1^* \rangle &= -\beta \langle \bar{X}^* \rangle V \delta t \,. && (62) \end{aligned}$$

Solving these simultaneously for β and V we find

$$\begin{aligned} V &= -\frac{1}{\delta t} \left(\langle X_2^* - X_1^* \rangle + \frac{\langle \bar{Y}^* \rangle}{\langle \bar{X}^* \rangle} \langle Y_2^* - Y_1^* \rangle \right) \,, && (63) \\ \beta &= \frac{\langle Y_2^* - Y_1^* \rangle}{\langle \bar{X}^* \rangle \langle X_2^* - X_1^* \rangle + \langle \bar{Y}^* \rangle \langle Y_2^* - Y_1^* \rangle} \,. && (64) \end{aligned}$$

4 Elimination of errors from track line detection

(4.1) Trial results indicate that the dominant source of error is due to the position and orientation of the camera being shaken by the movements in the train suspension and slight bumps in the track. We have no reliable way of predicting these errors, so we will seek a way of eliminating them frame by frame.

4.1 The camera image

(4.2) The camera image is obtained by a projective transformation Shenton et al. (2008). For more informations and illustrations on how cameras work see the website cam. The starting point for assembling this transformation is shown in Figure 6. The coordinate x measures distance along the track, from some fixed point on the ground, with y being in the transverse direction. Let the camera be at position (x_D, y_D), and at a height H above the plane of the track. Three angles describe the orientation of the camera: a declination or 'pitch' θ from the horizontal, a 'yaw' angle ϕ around the vertical, and a 'roll' ψ around the axis of the camera. To rotate this configuration about the camera, we use coordinates relative to the position of the camera (X, Y, Z), where $X = x - x_D$, $Y = y - y_D$ and $Z = z - H$. To simplify the geometry we rotate the image so that the camera axis is along the $X - axis$, and so that both the vertical axis and transverse axis from the camera's perspective point respectively along the $Z-$axis and $Y-$axis. To achieve this simplification we apply three rotation matrices,

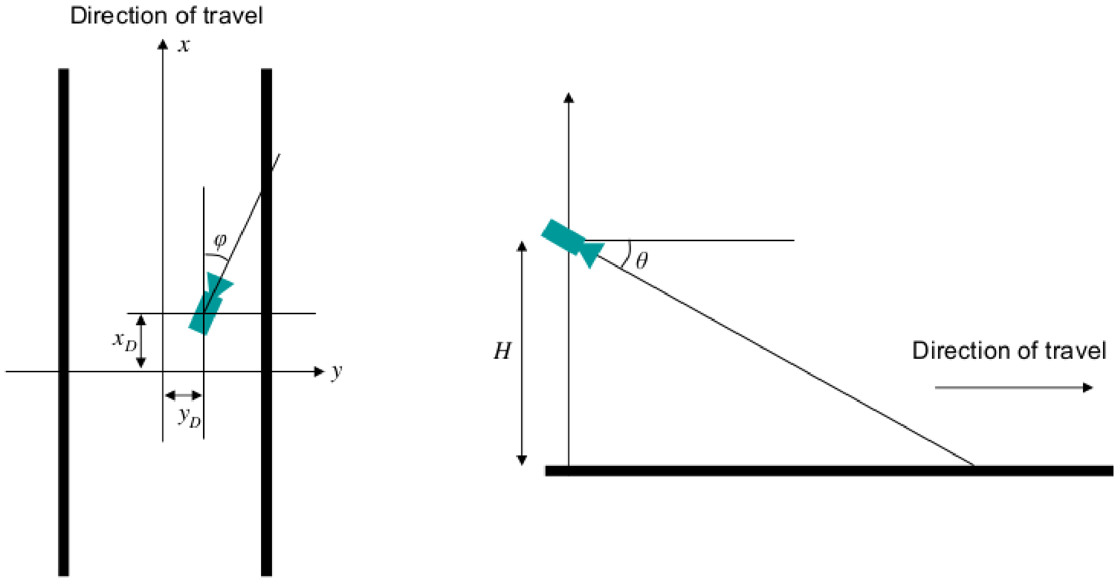


Figure 6: The Camera in its calibrated position. Note that if view from above the $y-$axis should be oriented in the opposite direction.

$$\begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} = R_1(\psi)R_2(\theta)R_3(\phi)\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{65}$$

where

$$R_3(\phi) = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{66}$$

$$R_2(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{67}$$

$$R_1(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \tag{68}$$

In the (ξ, η, ζ) coordinates the focal lens of the camera is in the $\eta \times \zeta$ plane. So to obtain the camera image we project the 3D tracks onto the focal lens. This way the coordinates (u, v) of the image on the camera's focal lens, often refered to as the camera image, become

$$\begin{pmatrix} f \\ u \\ v \end{pmatrix} = \frac{f}{\xi} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}, \tag{69}$$

where f is the focal length of the camera. For a convenient notation, we name F the function that takes the angles θ, ϕ, ψ and a point in the (X, Y, Z) system to the coordinates (u, v) on the camera image, so that

$$\begin{pmatrix} u \\ v \end{pmatrix} = F\left[\boldsymbol{\theta}, \boldsymbol{X}\right], \tag{70}$$

where $\boldsymbol{\theta} = (\psi, \theta, \phi)$ and $\boldsymbol{X} = (X, Y, Z)$. Note that F is a nonlinear function in all its arguments. If the camera is pointed inbetween the left and right rail, then the point where the camera axis intercepts the flat track is given by

$$(H \cot\theta \cos\phi, H \cot\theta \sin\phi, -H)\,. \tag{71}$$

Because the train shakes, and the camera is attached to the train, the camera image will not be exactly as described above. Small errors in the angle and position of the camera will be introduced. The camera may be initially setup with the angles θ, ϕ and ψ, but unknowingly, in any given frame, have the angles $\theta + \delta\theta$, $\phi + \delta\phi$ and $\psi + \delta\psi$, where $\delta\theta$, $\delta\phi$ and $\delta\psi$ are usually considered small disturbances. The position of the camera may also be incorrect, the camera may in fact be located at $\delta\boldsymbol{X} = (0, \delta Y, \delta Z)$ instead of $(0, 0, 0)$. We do not include an error for the X position of the camera δX, for the effect of δX on one frame of the camera image can not be perceived. That is, if we assume we do not know where the train is on the tracks. If the left rail is flat and the outer edge of this rail is described by $(X, Y_L, -H)$ for $X \in [0, D]$, the perturbed camera image of this outer edge will be

$$\begin{pmatrix} u \\ v \end{pmatrix} = F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X}\right], \quad \text{for } X \in [0, D], \tag{72}$$

for some value of $\delta\boldsymbol{\theta}$ and $\boldsymbol{X}$, where $\delta\boldsymbol{\theta} = (\delta\psi, \delta\theta, \delta\phi)$. However in general the tracks are not flat but have a small curvature. To model this we let the transverse position and height of the rail be

$$Y = Y_L + \beta\frac{X^2}{2} \quad \text{and} \quad Z = -H + \alpha\frac{X^2}{2}, \tag{73}$$

for some small parameters α and β. So the camera image could show the left track as

$$\begin{pmatrix} u \\ v \end{pmatrix} = F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X} + (0, \beta, \alpha)X^2/2\right], \quad \text{for } X \in [0, D]. \quad (74)$$

Note that the (X, Y, Z) coordinates are aligned with the tracks, so we can not have a linear contribution in X to either Y or Z of the rail's position.

4.2 Minimization method to determine track shape

(4.3) To accurately measure the rail curvature and undo the errors in the camera image introduced by the train shaking, we require that the outline of rails can be detected automatically. Without this automatic detection it is not clear how to accurately recover the curvature.

(4.4) Let the outer edge of the left rail as seen by the camera be given by the set of points $\mathcal{R}^{L_M}$. According to our model of what the camera image, the edge of this rail is described by

$$\mathcal{R}^L[\delta\boldsymbol{\theta}, \delta\boldsymbol{X}, (\alpha, \beta)] = \left\{F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X} + (0, \beta, \alpha)X^2/2\right];\ X \in [0, D]\right\}, \quad (75)$$

for some $\delta\boldsymbol{\theta}$ and $\delta\boldsymbol{X}$. This way if the train did not shake, and no errors were present, and the track was flat then $\mathcal{R}^{L_M} = \mathcal{R}^L[\mathbf{0}, \mathbf{0}, (0, 0)]$. However this is generally not the case. Similarly let $\mathcal{R}^{R_M}$ be the automatically detected outer edge of the right rail on the camera image and let $\mathcal{R}^R$ be defined analogously to $\mathcal{R}^L$.

To find out the curvature and errors we need to find $\delta\boldsymbol{\theta}$, $\delta\boldsymbol{X}$, α and β that minimize

$$\min_{\delta\boldsymbol{\theta}, \delta\boldsymbol{X}, \alpha, \beta} \|\mathcal{R}^L[\delta\boldsymbol{\theta}, \delta\boldsymbol{X}, (\alpha, \beta)] - \mathcal{R}^{L_M}\| + \|\mathcal{R}^R[\delta\boldsymbol{\theta}, \delta\boldsymbol{X}, (\alpha, \beta)] - \mathcal{R}^{R_M}\|, \quad (76)$$

where $\|\cdot\|$ denotes some measure of distance between two sets. How we choose to represent these sets, i.e. possible through a set of discrete points, and our choice for $\|\cdot\|$ will determine how effective the resulting method will be. One generally applicable norm $\|\cdot\|$ is the area between the two curves, such as the area delimited by the curves and lines that join the end points of these two curves. This norm can be applied to most any discretization of the sets. For example given the four points (u_1, v_1), (u_2, v_2), (u_3, v_3) and (u_4, v_4), the area between them is given by

$$\pm(u_1v_2 - u_2v_1 + u_2v_3 - u_3v_2 + u_3v_4 - u_4v_3 + u_4v_1 - u_1v_4),$$

where the ± should be chosen so that the area is positive. However developing the theory and a method along these lines is a bit beyond the scope of what was achieved during the study group. The method we will develop here will assume that the points chosen on $\mathcal{R}^{R_M}$ and $\mathcal{R}^{L_0}$ result from taking points evenly spaced along $X-$axis and then mapping them with some $F[\boldsymbol{\theta}', \boldsymbol{X}']$ to the (u, v) coordinate system. This assumption simplifies the resulting calculations so that we may easily demonstrate how to develop a minimization method. With this assumption we let $\mathcal{R}^{L_M}$ and $\mathcal{R}^{R_M}$ be sets of discrete points. For convenience we join all these points into one vector

$$\boldsymbol{U}^M = \{\mathcal{R}^{L_M}, \mathcal{R}^{R_M}\} = \{(u_M^1, v_M^1), (u_M^2, v_M^2), \ldots, (u_M^{2N}, v_M^{2N})\}. \tag{77}$$

where N is the number of points on each rail. We consider each (u_M^j, v_M^j) to be an element of $\boldsymbol{U}^M$. So that the transpose $(\boldsymbol{U}^M)^T$ does not affect the order within each element (u_M^j, v_M^j). Analogously the points $\mathcal{R}^L$ and $\mathcal{R}^R$ are evenly spaced on the $X-$axis. To generate them we take the even spaced points along the left rail and right rail and then map them to the camera image with

$$F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X}\right] \quad \text{and} \quad F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_R, -H) + \delta\boldsymbol{X}\right]$$

respectively. The values Y_L and Y_R are the transverse width to reach the outer edge of the left and right rail respectively. In the same form as equation (78) we line up the points from the left rail $\mathcal{R}^L$, followed by the points from the right rail $\mathcal{R}^R$ to form

$$\boldsymbol{U} = \{\mathcal{R}^L, \mathcal{R}^R\} = \{(u^1, v^1), (u^2, v^2), \ldots, (u^{2N}, v^{2N})\}. \tag{78}$$

Seeing that (u^j, v^j) and (u_0^j, v_0^j) are the result of mapping the same point from the (X, Y, Z) to the (u, v) coordinate system, it makes sense to minimize the distance between them. We therefore choose the measure distance in the following way

$$\|\boldsymbol{U} - \boldsymbol{U}^M\| = (\boldsymbol{U} - \boldsymbol{U}^M)^T(\boldsymbol{U} - \boldsymbol{U}^M) = \sum_{j=1}^{2N}(u^j - u_0^j)^2 + (v^j - v_0^j)^2 \tag{79}$$

4.3 Linearise and project

(4.5) The procedure for minimizing (76) for $\delta\boldsymbol{\theta}$, $\delta\boldsymbol{X}$, α and β would result in a system of nonlinear equations, which would likely require a more involved numerical solution. Luckily, we expect both the errors in the camera $\delta\boldsymbol{\theta}$, $\delta\boldsymbol{X}$ and the curvature parameters α, β to be much smaller than 1. For convenience let $\boldsymbol{\delta} = (\delta\boldsymbol{\theta}, \delta\boldsymbol{X}, \alpha, \beta)$, this was we can expand

$$\boldsymbol{U} \approx \boldsymbol{U}^0 + \partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta}, \tag{80}$$

where $\boldsymbol{U}^0$ is $\boldsymbol{U}$ evaluated at $\boldsymbol{\delta} = 0$. The term $\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0$ can be seen as a matrix whose elements are points in the (u, v) coordinate system. We substitute this expansion in equation (79) to obtain

$$\|\boldsymbol{U} - \boldsymbol{U}^M\| \approx (\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta} + \boldsymbol{U}^0 - \boldsymbol{U}^M)^T(\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta} + \boldsymbol{U}^0 - \boldsymbol{U}^M). \tag{81}$$

At this point one should check that $\|\boldsymbol{U}^0 - \boldsymbol{U}^M\| < 1$ as both these quantities are known a prior and the method makes this assumption. To minimize $\|\boldsymbol{U} - \boldsymbol{U}^M\|$ we need to differentiate it in the parameters $\boldsymbol{\delta}$ and then set the result to zero,

$$\begin{aligned}
\min_{\boldsymbol{\delta}} \|\boldsymbol{U} - \boldsymbol{U}^M\| &\implies \partial_{\boldsymbol{\delta}} \|\boldsymbol{U} - \boldsymbol{U}^M\| = \mathbf{0} \\
&\implies (\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T (\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0 \cdot \boldsymbol{\delta} + \boldsymbol{U}^0 - \boldsymbol{U}^M) = \mathbf{0} \\
&\implies (\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0 \cdot \boldsymbol{\delta} = (\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T (\boldsymbol{U}^M - \boldsymbol{U}^0) \\
&\implies \boldsymbol{\delta} = \left((\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0\right)^{-1} (\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T (\boldsymbol{U}^M - \boldsymbol{U}^0). \qquad (82)
\end{aligned}$$

The last step assumes that $(\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0$ is invertible, which is only possible if each parameter in $\boldsymbol{\delta}$ causes an independent change to the camera image. That is if the columns of $\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0$ are mutually independent. To this end, it can be helpful to graphically plot $\boldsymbol{U}^0 + \partial_{\delta_j} \boldsymbol{U}^0 \delta_j$, which we will call distortions, for each element of $\boldsymbol{\delta}$, where we choose δ_j to be an small arbitrary number. In Figure 7 we draw each distortion for the fixed parameters $f = 1$ m, $H = 2.5$, $\phi = \pi/50.$, $\psi = 0.$, $\theta = \pi/30$, track width 1.46 m with the rails range from $X = 5$ m to $X = 20$ m. This figure reminds us that lines remain lines under rotations and translations, which in turn leads us to an important question: how many dimensions has the space of two lines with finite length? To describe each finite 2D line we need 3 parameters, so both of these lines together live in a 6 dimensional space. If the distortions created by $\delta\theta$, $\delta\phi$, $\delta\psi$, dY, dZ are independent they will span 5 of these 6 possible dimensions. This means it is quite possible there exists a choice for θ, ψ, ϕ, H and rail positions Y_L and Y_R such that these distortions are not independent, implying that for this choice $(\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0$ would not be invertible.

(4.6) An even worse scenario would occur, for example, if one chooses a measure of distance $\|\cdot\|$ between curves, discussed in Section 4.2, such that it ignores the length of the curves. In this case the space of two finite 2D lines would have 4-dimensions and $(\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0$ would not be invertible. Though if one is only interested in recovering α and β this problem can be easily worked around.

(4.7) For the method used in this section, in all the parameter choices we explored, $(\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0$ was always invertible. The further the determinant $\det\left((\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0\right)$ the less numerical errors will be introduced when calculating the inverse of $\partial_{\boldsymbol{\delta}} \boldsymbol{U}^0)^T \partial_{\boldsymbol{\delta}} \boldsymbol{U}^0$. The value of this determinant changes with the camera position and orientation, so it is possible to choose the camera position and orientation so as to increase this determinant and therefore lower the numerical errors introduced.

4.4 Results

(4.8) With the help of Richard Shenton, we created some synthetic data which would give an extreme case. With the camera setup so that

$$f = 1 \text{ m}, \; H = 2.5 \text{ m} \; \phi = \pi/36, \; \psi = 0, \; \theta = \pi/12,$$

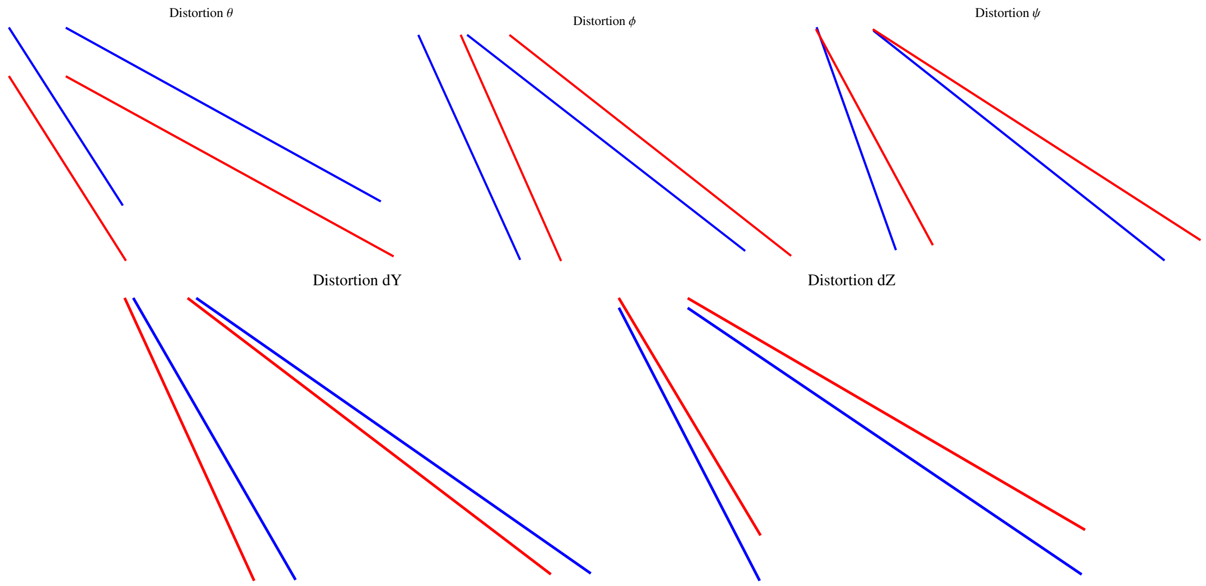


Figure 7: Distortion to the rails in the camera image due to errors in the camera angles and position. The red rails are all perfectly straight rails with no camera errors. The blue rails are the linearised disturbances to the red rail.

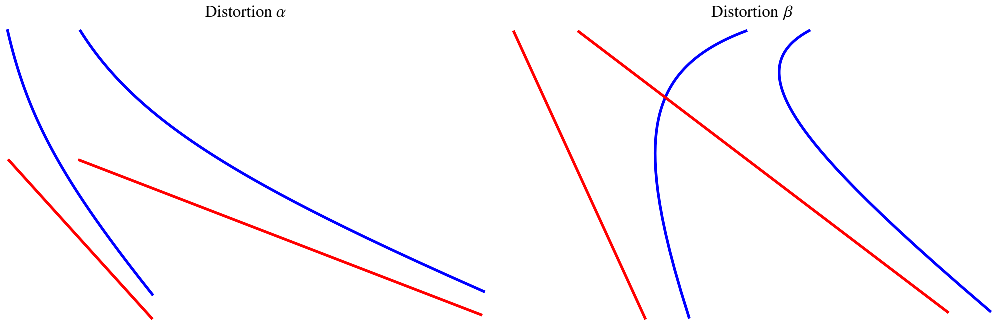


Figure 8: Distortion to the rails in the camera image due to errors in the camera angles and position. The red rails are all perfectly straight rails with no camera errors. The blue rails are the linearised disturbances to the red rail.

with a track width of 1.46 m. We assume that the camera is calibrated to point between the two tracks, so using equation (71) we find that the point where the camera's axis intercepts the flat rails is $(X, Y) = (9.3, 0.81)$ and therefore the left and right rail have $Y = 0.81 - 1.46/2$ and $Y = 0.81 + 1.46/2$ respectively. We also assume that the camera can see both the rails from $X = 5$ m to $X = 15$ m. The extreme case for each of the errors and curvature parameters used was

$$|\delta\theta| = |\delta\phi| = |\delta\psi| = 3.6^\circ, \tag{83}$$

$$|dZ| = |dY| = 0.014, \tag{84}$$

$$\alpha = 0.01 \text{ and } \beta = 0.03. \tag{85}$$

This synthetic data was generated without linearising any of the equations. Errors were simply included in the model (74), and the analogous for the right rail, to generate the synthetic data.

(4.9) The method developed in Section 4.3 was applied to this synthetic data, and to illustrate we produced a video CompareRide.gif. The video compares the observed position of the tracks in blue with the position of the tracks after using the method to estimate the curvature parameters α and β in red. Some snapshots of this video are shown in Figure 9. The video CompareRide.gif also shows the error made in estimating α and β, the estimated values α^* and β^* is compared with the real values for α and β use to generate the synthetic data in Figure 10. We believe the errors are mainly introduced due to the nonlinearity of the camera image model (74) in the parameters θ, ϕ, ψ, Y_L and H, whereas the method linearises in terms of these parameters.

5 Matching curves

(5.1) The mathematical problem is as follows:

1. Reconstruct a curve from local measurements.
2. Match a section of a curve to a given set of curves (a map of tracks).

5.1 Measurements

(5.2) The video odometry system measures,

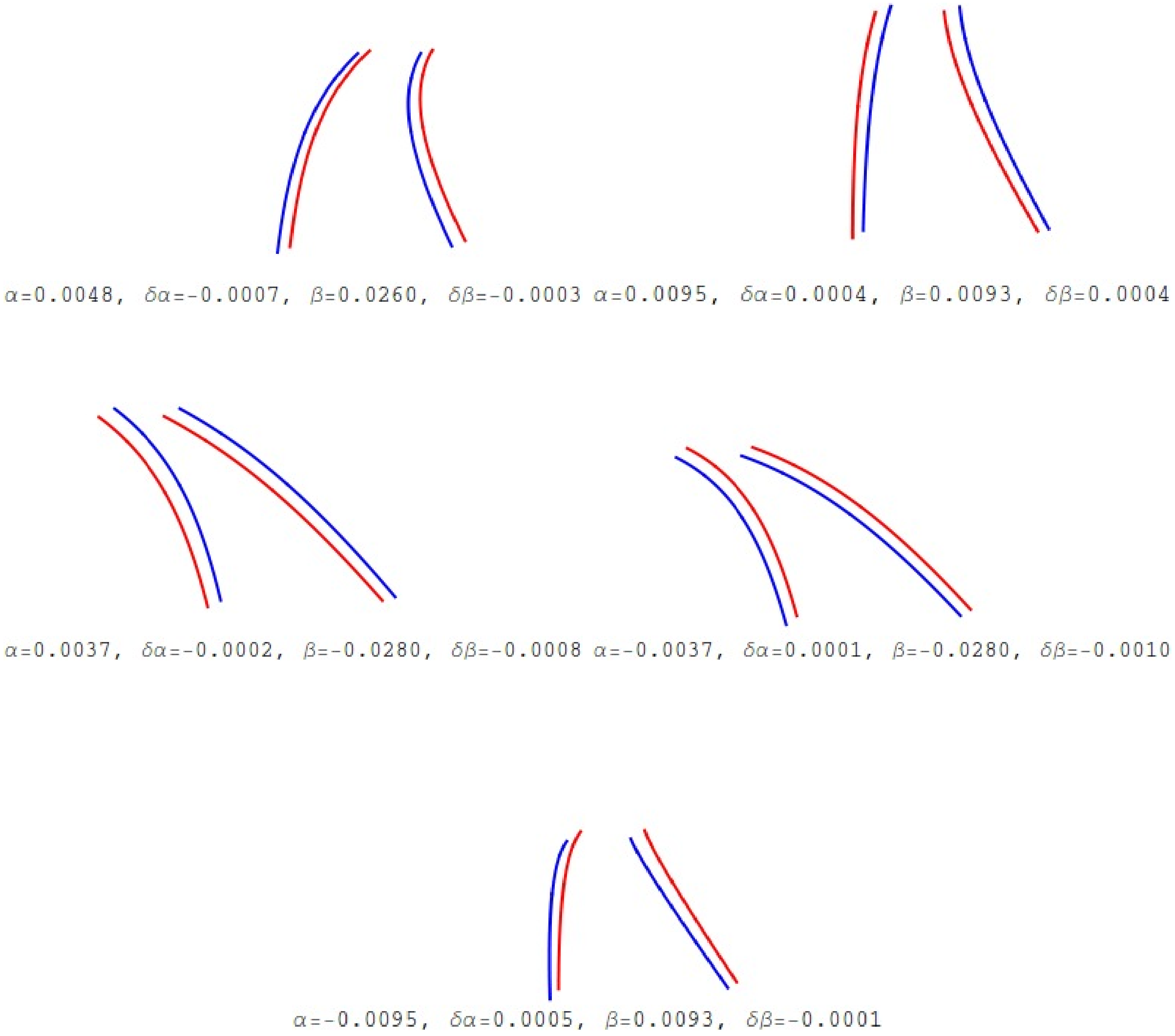
- dispacement forwards,
- displacement sideways,

both measured in pixels per frame. We start by assuming that the train moves in a flat plane (2D). Let (x, y) denote the absolute coordinate system and (ξ, η) the local coordinate system attached to the train.

(5.3) The η-axis points in the direction of travel and the ξ-axis points perpendicularly to the right. We take the train to be a point moving on the curve representing the track.

Let Δt denote the time between frames (which is constant), measured in seconds. We have $\Delta t = 1/f$, where f is the number frames per second (typical values are 25 and 60). After being converted from pixels to meters, the displacement forwards per frame gives the velocity of the train, i.e.,

$$v = \frac{\mathrm{d}\eta}{\mathrm{d}t} \approx k_\eta \mathsf{FD}. \tag{86}$$



α=0.0048, $\delta\alpha$=-0.0007, β=0.0260, $\delta\beta$=-0.0003

α=0.0095, $\delta\alpha$=0.0004, β=0.0093, $\delta\beta$=0.0004

α=0.0037, $\delta\alpha$=-0.0002, β=-0.0280, $\delta\beta$=-0.0008

α=-0.0037, $\delta\alpha$=0.0001, β=-0.0280, $\delta\beta$=-0.0010

α=-0.0095, $\delta\alpha$=0.0005, β=0.0093, $\delta\beta$=-0.0001

Figure 9: Distortion to the rails in the camera image due to errors in the camera angles and position. The red rails are all perfectly straight rails with no camera errors. The blue rails are synthetic data from the full nonlinear model.

The sideways displacement per frame can be used to calculate the curvature of the track,

$$\kappa = \frac{\mathrm{d}\theta}{\mathrm{d}s} = \frac{\mathrm{d}\theta}{\mathrm{d}t} / \frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\mathrm{d}\theta}{\mathrm{d}t} / v \tag{87}$$

The angle θ represents the rotation of the local coordinate frame (ξ, η) with respect to the absolute coordinate frame (x, y). The arc length s represents distance along the curve. The sideways displacement is turned into an angle by dividing by L which represents the distance from the camera to the point on the ground the camera is pointing at. The curvature can be written in terms of the conversion

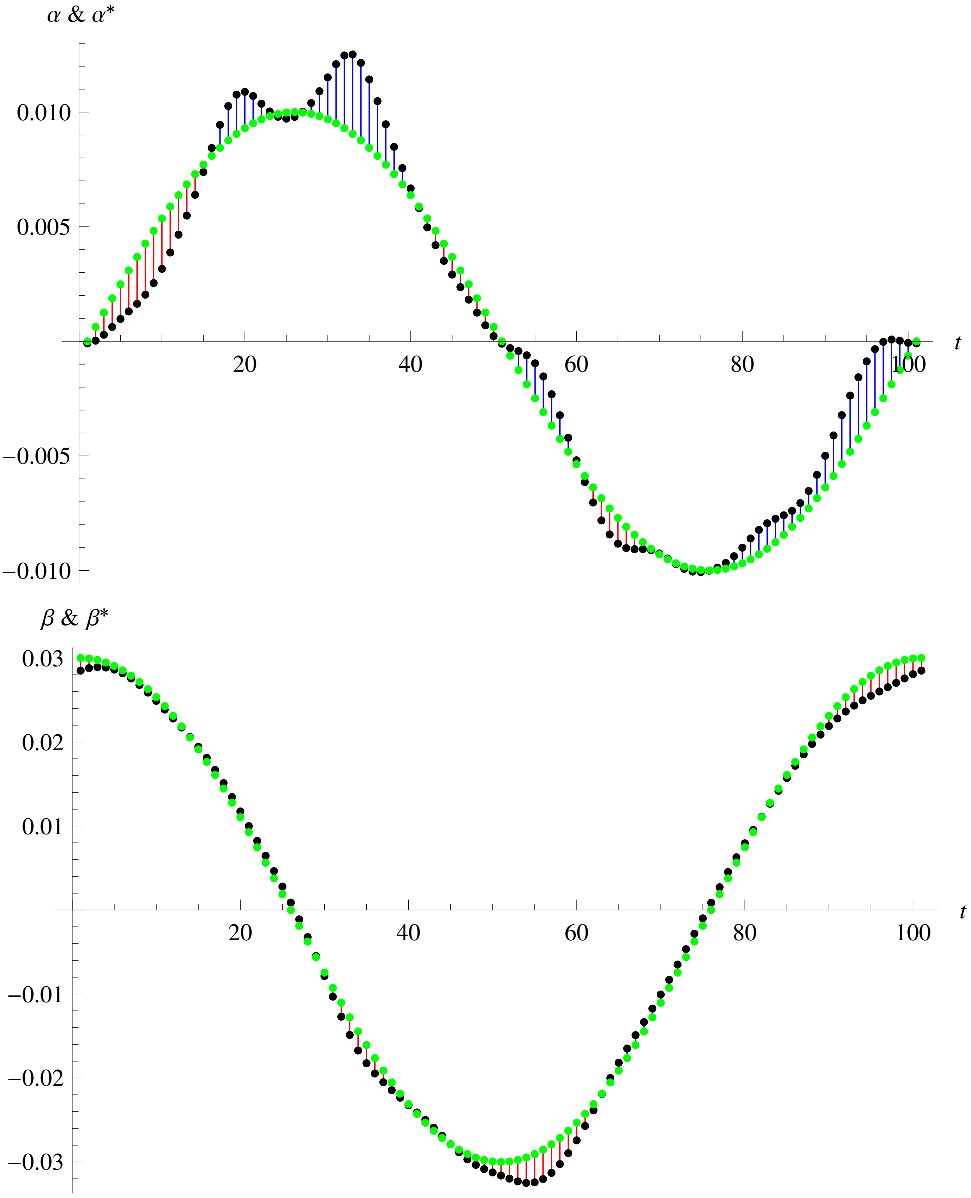


Figure 10: The green points are either the true value of α or β, while the the black points are either the estimated values α^* or β^*. The horizontal axis is the time steps of the video.

from pixels to meters for each of the directions (k_ξ, k_η).

$$\kappa \approx \frac{k_\xi}{k_\eta L}\frac{\mathsf{SD}}{\mathsf{FD}}. \tag{88}$$

The above formula assume that Δt is small enough that $\tan(\Delta\theta) \approx \Delta\theta$. Furthermore, all 3D effects (pitch, yaw, roll) are neglected. In other words it is assumed that the track is locally lying in a flat plane. We will discuss later on how the vertical position of the track can be taken into account in the curve matching process.

(5.4) By the above formulas the video odometry system directly gives the velocity v

and the curvature κ as a function of time t. A natural way to represent a curve based on local measurements is the intrinsic representation where the curvature κ is given as a function of arc length s (i.e. distance along the curve/track). Since we have $v(t)$ and $\kappa(t)$, we can find $\kappa(s)$ by integrating v.

$$\frac{\mathrm{d}s}{\mathrm{d}t} = v, \tag{89}$$

$$\frac{\Delta s}{\Delta t} \approx k_\eta \mathsf{FD}, \tag{90}$$

$$s = \int_0^t v\,\mathrm{d}t, \tag{91}$$

$$s \approx \sum_0^t k_\eta \mathsf{FD} \Delta t. \tag{92}$$

In addition to forward displacement (FD) and sideways displacement (SD) the video odometry system also provides a quality measure Q. This measure can be used to identify points on the curve where the system has failed to provide useful measurements, e.g., due to poor lighting conditions. Like the measurements FD and SD, the quality measurement Q has a lot of high frequency noise.

(5.5) It is recommended to use smoothing to get rid of this noise. This can be conveniently done by some kind of moving average process. A simple moving mean or moving median filter with a window of 2 seconds turned out to be satisfactory in experiments. It is also possible to use an exponential mean. For other options (and caveats about the moving mean see the Wikipedia – Moving Average page).

$$\tilde{Q}_i = (1 - \sigma) Q_i + \sigma \tilde{Q}_{i-1}. \tag{93}$$

If $\sigma = 0$ there is no smoothing, if $\sigma = 1$ then Q_i is ignored. Choose σ such that $\sigma^{Wf} = \sigma^{\frac{W}{\Delta t}} = \frac{1}{2}$ where W is the smoothing window, e.g., 1 or 2 seconds. The quality measure Q is a number between 0 and 255. The initial value can be set to $\tilde{Q}_0 = 255$. To clean up the data negative forward displacements ($\mathsf{FD} < 0$) were discarded, along with points that had the (smoothed) quality measure below a certain threshold (when $\tilde{Q} < 0.5 \times Q_{\max}$). Note that Δt is no longer a constant when measurements are ignored.

5.2 Curve reconstruction

(5.6) Angle Formulation

From $\kappa(s)$ we can reconstruct the coordinate-arc length representation $(x(s), y(s))$ of the curve by using the formulas

$$\frac{\mathrm{d}\theta}{\mathrm{d}s} = \kappa, \quad \frac{\mathrm{d}x}{\mathrm{d}s} = \cos(\theta), \quad \frac{\mathrm{d}y}{\mathrm{d}s} = \sin(\theta). \tag{94}$$

By direct integration this gives

$$\theta = \theta_0 + \int_0^s \kappa \, \mathrm{d}s \qquad = \theta_0 + \int_0^t \kappa \, v \, \mathrm{d}t, \tag{95}$$

$$x = x_0 + \int_0^s \cos(\theta) \, \mathrm{d}s = x_0 + \int_0^t \cos(\theta) \, v \, \mathrm{d}t, \tag{96}$$

$$y = y_0 + \int_0^s \sin(\theta) \, \mathrm{d}s \ = y_0 + \int_0^t \sin(\theta) \, v \, \mathrm{d}t. \tag{97}$$

Using our video odometry measurements this becomes

$$\frac{\Delta\theta}{\Delta t} \approx \frac{k_\xi}{L}\mathsf{SD}, \tag{98}$$

$$\frac{\Delta x}{\Delta t} \approx k_\eta \mathsf{FD} \cos(\theta), \tag{99}$$

$$\frac{\Delta y}{\Delta t} \approx k_\eta \mathsf{FD} \sin(\theta), \tag{100}$$

or, using integration notation

$$\theta \approx \theta_0 + \sum_0^t \frac{k_\xi}{L} SD \, \Delta t, \tag{101}$$

$$x \approx x_0 + \sum_0^t k_\eta \cos(\theta) FD \, \Delta t, \tag{102}$$

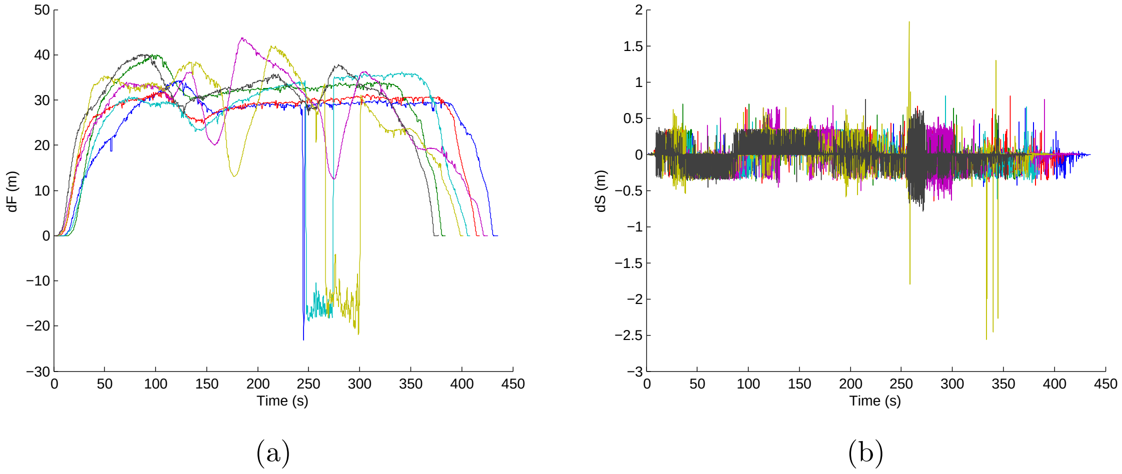
$$y \approx y_0 + \sum_0^t k_\eta \sin(\theta) FD \, \Delta t. \tag{103}$$

Note that the initial position (x_0, y_0) can be interpreted as a translation of the reconstructed curve. The initial angle θ_0 represents the orientation of the curve and the forward pixels to meters conversion faction k_η can be interpreted as a scaling of the curve. All of these parameters can be incorporated in the curve matching process discussed later on.

The coefficient $K = \frac{k_\xi}{L}$, however, determines the “curviness” of the curve. For $K = 0$ we would get a straight line and as K increases the curve folds in on itself more and more. In other words, changing K affects the curve in a nonlinear way. Using K for calibration based on (x, y) information would require a nonlinear optimisation process. It is suggested that this parameter should be determined as accurately as possible from the parameters of the camera system. The parameter K scales the angle θ, so calibration based on θ information (e.g., from compass measurements) may be easier.

Note that θ, x and y are found by integrating measured values. Since integration is a smoothing process, this suggests that no extra smoothing of the data may be necessary. This was born out by our preliminary experiments.

Accumulation of errors, i.e., dead-reckoning errors, are unaffected by smoothing. They are an inherent part of any system reconstructing a curve from only local



(a) (b)

Figure 11: The (a) forward and (b) sideways displacement against time from the seven runs.

measurements. Such errors could be addressed by calibration using other (globarl) measurements (e.g., total distance travelled; initial and final positions and orientations; positions and orientations along the way; e.g., form GPS or compass readings). This will be discussed further in the curve matching section.

(5.7) Purely Cartesian Formulation Since we have assumed that Δt is small enough so that SD and therefore $\Delta\theta$ are small ($\tan(\Delta\theta) \approx \Delta\theta$), we can remove the need for trigonometric functions in the curve reconstruction. For small $\Delta\theta$ we have

$$\cos(\theta + \Delta\theta) \approx \cos(\theta) - \Delta\theta \sin(\theta), \tag{104}$$

$$\sin(\theta + \Delta\theta) \approx \sin(\theta) + \Delta\theta \cos(\theta). \tag{105}$$

With $p \approx \cos(\theta)$ and $q \approx \sin(\theta)$ we get p and q by integrating SD and x and y by integrating FD

$$\begin{aligned} \frac{\Delta p}{\Delta t} &\approx -\frac{k_\xi}{L}\mathsf{SD}\, q, & \frac{\Delta x}{\Delta t} &\approx k_\eta \mathsf{FD}\, p, \\ \frac{\Delta q}{\Delta t} &\approx \frac{k_\xi}{L}\mathsf{SD}\, p, & \frac{\Delta y}{\Delta t} &\approx k_\eta \mathsf{FD}\, q. \end{aligned} \tag{106}$$

The vector $(p, q) \approx (\cos(\theta), \sin(\theta))$ now represents the instantaneous orientation of the (ξ, η) coordinate frame attached to the train. Note that the condition $p^2 + q^2 = 1$ will not be preserved exactly.

(5.8) An Example of Curve Reconstruction

The data used was the Mid Hants dataset, which contained seven runs over the same section of track, shown in figure 11.

The data shown in figure 11 has been smoothed using a moving average filter over 25 frames, which is equivalent to one second of data. However, as discussed earlier in this report, further smoothing is required. For example, the dips in two of the runs from positive to negative displacement are physically unrealistic as

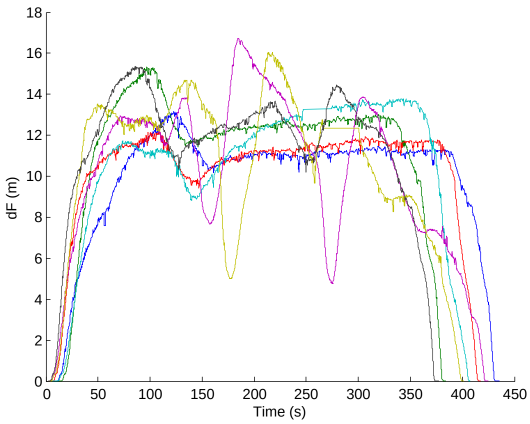


Figure 12: The forward displacement against time from the seven runs.

this implies the train has accelerations that are improbably large. These points were ignored and the quality measure was used to remove some points, leaving the data shown in figure 12.

The curvature of the track is calculated from the forward and sideways displacements and the distance of the camera from the focus point on the track as described in equation 88. The distance of the camera from the focus point on the track was approximated to be $L \approx 6$m. To calculate the route from the mean $\kappa(s)$ the curvature must be calculated against distance, rather than against time. The distance is calculated by integrating FD with time. This was done using the trapezium rule (cumtrapz in MatLab).

As the velocity profile for each of the seven runs is different they will have recorded the data over a different distance vector. Therefore they had to be mapped onto the same distance vector in order to calculate the mean curvature at a certain distance. This was done using linear interpolation and the results are shown in figure 13. The average of the seven runs is shown in red. It was assumed that the start point of each of the runs was identical, such that the curvature with distance down the track would be common to all runs. If this was not known to be true, one run could be taken as a basis and all other runs compared to that using cross correlation, to correct for changes in starting position of the train.

The curvature was then integrated with distance to plot the track of the train from the curvature results. The routes from data taken on each of the seven runs and calculated from the mean curvature data (dashed line) are shown in figure 14.

Future suggestions would include pre-analysing the data using ideas from earlier in this report and improving the matching of historical data (cross correlating to improve the average $\kappa(s)$).

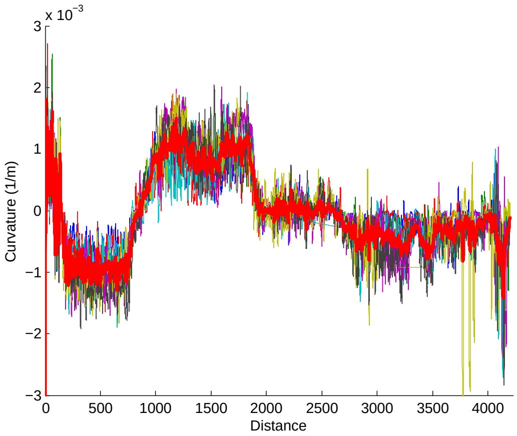


Figure 13: The curvature κ against distance for the seven runs. The average curvature is shown in red.

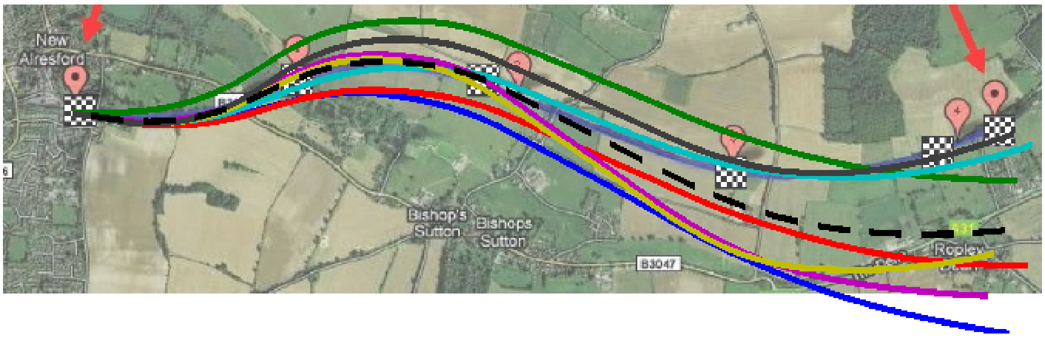


Figure 14: The curves generated using approximate calibration from each of the seven runs (solid lines) and mean curvature data (dashed line) against the true route of the train from a map provided.

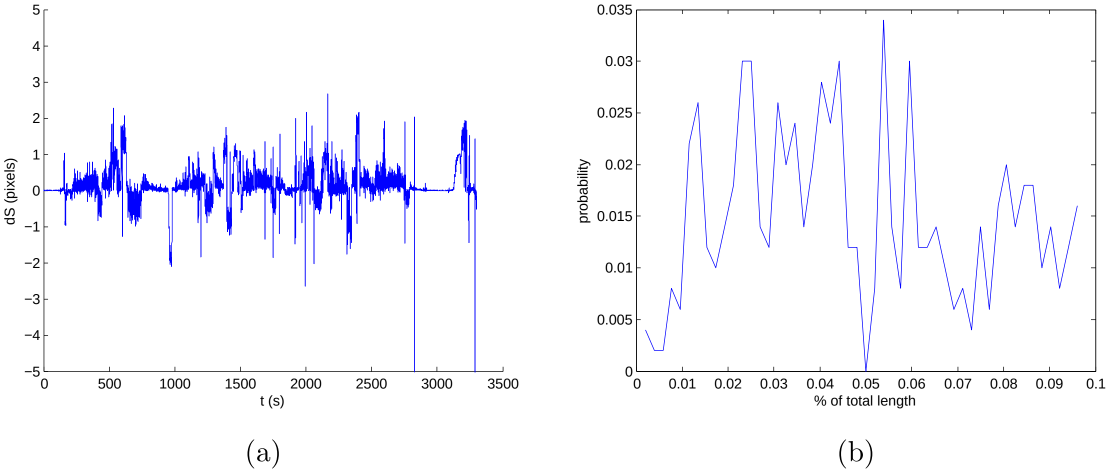


Figure 15: The (a) sideways displacement from the camera data and (b) The length required to match a sample to the signal.

5.3 Curve Matching

(5.9) Our examination of curvature was on a couple of different scales. Firstly, curvature itself occurs on two different scales. When the train changes track the signal from the curvature spikes with a much higher magnitude than when the track curves. The spikes in curvature when the train changes track was used in particle filter description of the train network.

(5.10) We were asked about the possibility of calculating the position of a train from curvature data. The position of a train on a particular stretch of track cannot be determined from an instantaneous measure of curvature (even with perfect data) because curvature is piecewise linear (i.e. the curvature is constant on any particular curve and not unique to a point on the track). Therefore it could be identified that the train was on a particular curve, but not where on that curve it was. However, a longer measurement could potentially identify the position of the train.

The signal of sideways displacement data for one journey is shown in figure 15a. If a sample of some length is taken from this signal we could then try and calculate where in the signal it was taken from. We will refer to the sample as $f(i)$ and the signal $h(i)$. The initial method used to calculate this was the mean squared error,

$$\epsilon(i_0) = \frac{1}{N} \sum_{i_0}^{i_0+N} \left(f(i - i_0) - h(i)\right)^2 , \tag{107}$$

where N is the number of data points in the sample. The mean squared error is ϵ is calculated as a function of starting position i_0 along the historical signal. The best guess for where along the track the sample came from is where ϵ is a minimum.

This can be adapted to include information from the quality measure Q. This is achieved simply by normalising by Q,

$$\epsilon(i_0) = \frac{1}{N} \sum_{i_0}^{i_0+N} \frac{Q}{Q_{\max}} \left(f(i - i_0) - h(i)\right)^2 . \tag{108}$$

This method is very slow to run however. A more efficient method is to use cross correlation (which is equivalent to 107 but makes use of fast fourier transforms to speed things up). The position with the maximum correlation between the signals is the location of the sample in the historical signal.

There are some possible problems with this method. The GPS is sufficient to tell where in the country the train is. However, it cannot differentiate between adjacent tracks. It seems likely that adjacent tracks will have similar curvature and therefore this method would also be unable to differentiate between adjacent tracks. Added to this the piecewise linear nature of the curvature along a particular track means that some stretches will be straight (or have constant curvature) and measurements will have to continue for a long time before the position can be deduced.

One thing that was tried was calculating the required length of a sample for its location to be correctly (and uniquely) identified. This was calculated for many random starting points in the signal in figure 15a. The results are shown in 15b which shows length of signal (in % of the length of the original signal) against fraction of starting points that require that length to uniquely identify the starting point.

(5.11) Representation

The $\kappa(s)$ representation of a train journey is very useful for detecting and locating the distinctive signature of track changes. Track changes appear as significant peaks in κ over a short distance. For example, 2 track changes in one direction, followed by 2 track changes back, on an otherwise straight track may look like

By converting the map representation of the track to $\kappa(s)$ representation, the $\kappa(s)$ representation of the journey could be used for matching as well. We posit however that this is neither the most natural nor the most useful approach. Since errors in the measurement FD can manifest themselves as a distortion of the s-axis, this means that some kind of elastic matching or dynamic warping should be used.

The $(x(s), y(s))$ representation can be used directly for matching. This requires the map representation of the tracks to be converted into this form as well. It will be convenient to use an equidistant sampling in s, for both the measured segment and the reference curve. The parameter Δs should be chosen carefully to balance accuracy and computational cost (storage and processing time).

The measured journey will have a constant Δt (except when measurements are discarded), but Δs will be variable. We propose a resampling using a constant

Δs. A simple procedure like linear interpolation should be sufficient, but more advanced interpolation methods or function approximation methods could be investigated.

The resampling can be done for the $\kappa(s)$ representation, before $(x(s), y(s))$ reconstruction or it can be done directly in the (x, y) space.

The former may be recommended, especially if the piecewise linear nature of the $\kappa(s)$ representation is exploited. This is not as natural for the map information however, since the tracks are already given by a series of (x, y) points.

For a series of (x, y) points (either the reconstructed journey or a map representation of a track), the resampling can be done by piecewise linear interpolation (or again by some more sophisticated interpolation or approximation, e.g., splines or using the $\kappa(s)$ representation).

(5.12) Comparing Two Curves

If we have two curves represented by equidistant (x, y) samples, then we can define a distance metric d between them by

$$d^2 = \sum (\delta x^2 + \delta y^2)\, \Delta s, \tag{109}$$

where δx and δy represent the difference between corresponding points on the two curves. We assume for now that the two curves have the same number of points. We use the Euclidean version (exponent 2) since this will allow efficient calculation for the matching of subsegments Let $\delta x_i = x_i - \hat{x}_i$ and $\delta y_i = y_i - \hat{y}_i$, where $\hat{\cdot}$ denotes the reference curve.

(5.13) Curve Segments

Suppose a reference curve is given by equidistant samples $(\hat{x}_i, \hat{y}_i)$, $i = 0, \ldots, n-1$ and we want to match and locate a given curve segment (x_j, y_j), $j = 0, \ldots, m-1$. The curve segent is equidistantly sampled with the same Δs, but it is shorter than the reference curve, i.e., there are fewer samples ($m \leq n$). We can find whether and where the curve segment matches the reference curve by finding the value i for which our previous distance metric is minimised. Define

$$d_i^2 = \sum_{j=0}^{m-1} \left((x_j - \hat{x}_{i+j})^2 + (y_j - \hat{y}_{i+j})^2 \right) \Delta s, \tag{110}$$

and

$$d = \min_i d_i. \tag{111}$$

If implemented directly, this will take $O(mn)$ operations, where n and m are the number of samples in the reference curve and the curve segment respectively. This is acceptable if m is small. We can perform the calculation of the d_i values more efficiently using FFT, however, by rewriting the calculation as a cross-correlation. We get

$$
\begin{aligned}
d_i^2 &= \sum_j \big((x_j - \hat{x}_{i+j})^2 + (y_j - \hat{y}_{i+j})^2\big)\ \Delta s, && (112)\\
&= \sum_j (x_j^2 + y_j^2)\,\Delta s + \sum_j (\hat{x}_{i+j}^2 + \hat{y}_{i+j}^2)\,\Delta s + 2\sum_j x_j \hat{x}_{i+j}\,\Delta s + 2\sum_j y_j \hat{y}_{i+j}\,\Delta s. && (113)
\end{aligned}
$$

The first term does not depend on i and can be calculated in $O(m)$ operations for the given curve segment.

The second term can be pre-calculated for each reference curve in $O(n)$ operations.

$$\sum_{j=0}^{m-1} \hat{x}_{i+j}^2 = \sum_{k=i}^{i+m-1} \hat{x}_k^2 = \sum_{k=0}^{i+m-1} \hat{x}_k^2 - \sum_{k=0}^{i-1} \hat{x}_k^2 = X_{i+m-1} - X_{i-1}, \tag{114}$$

where $X_i = \sum_{k=0}^{i} \hat{x}_k^2$ are cumulative sums of squares.

The last terms are two cross-correlations which can be calculated in $O(n\log(n))$ operations using FFT. To do this, pad the reference with m zeros to a signal of length $n+m$ and pad the segment with n zeros to a signal of length $n+m$. We then have

$$c_i = \sum_{j=0}^{m} x_j \hat{x}_{i+j} = \sum_{j=0}^{n+m} x_j \hat{x}_{i+j}. \tag{115}$$

By the convolution theorem the cross-correlation signal can be calculated by componentwise multiplication in the Fourier domain (the padding is necessary to handle the periodic boundaries)

$$c = x * \hat{x} = \mathsf{IDFT}\big(\mathsf{DFT}(x) \cdot \mathsf{DFT}(\mathrm{conj}(\hat{x}))\big). \tag{116}$$

The transformations can be calculated in $O((n+m)\log(n+m))$ using FFT and the componentwise multiplication and conjugation take $O(n)$ operations. Note that $O((n+m)\log(n+m)) = O(n\log(n))$ since $m \leq n$. The transformations for the reference curves can be pre-calculated. The x and y signals can be combined as real and imaginary parts of a complex signal so that a single complex FFT suffices for both cross-correlation calculations.

It may be possible to get extra efficiency gains by updating the calculations for the curve segments of increasing number of samples, i.e., as the train moves along.

(5.14) Dealing with Translation, Scaling and Rotation

As was remarked in the section on curve reconstruction, it is possible to account for uncertainties in the position, scale and orientation of the curve segment matched. To keep the notation simple, we will asssume in this section that the curve and the reference curve have the same number of samples.

We consider here curves in (x, y) space. The parameter k_η determines directly the scale of the reconstructed curve. The initial position (x_0, y_0) determines the location of the reconstructed curve. The initial angle θ_0 (or the vector (p_0, q_0)) determines the orientation of the reconstructed curve. If these parameters are known or determined from other measurements (e.g., GPS, compass), then this can help the matching process. It is, however, possible to perform matching without knowing some or all of these parameters. Specifically, we can find the values of the parameters that give the best match. The parameter $K = k_\xi/L$ determines the "curviness" of the reconstructed curve and its effect is not as easy to discount.

It may be useful to have bounds for K if its value is not known exactly.

In what follows we show how each of the parameters determining position (x_0, y_0), scale (k_η) and orientation (θ_0) can be taken out of the matching process.

(5.15) Position

The position parameters (x_0, y_0) can be taken out of the matching process by translating the reconstructed curve and the reference curve so that they have the same mean position (e.g., $(0, 0)$). We have points (x_i, y_i), $i = 0, \ldots, n-1$ on the reconstructed curve and points $(\hat{x}_i, \hat{y}_i)$, $i = 0, \ldots, n-1$ on the reference curve. If we do not know the position of the reconstructed curve (i.e., we used arbitrary (x_0, y_0)), then we can find the translation (t_x, t_y) that minimises the distance metric between the curves. Define

$$d^2(t_x, t_y) = \sum_i \left((x_i + t_x - \hat{x}_i)^2 + (y_i + t_x - \hat{y}_i)^2\right) \Delta s. \tag{117}$$

The new distance metric is

$$d = \min_{(t_x, t_y)} d(t_x, t_y). \tag{118}$$

Expanding the brackets gives

$$\begin{aligned} d^2(t_x, t_y) = &\sum_i \left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right) \Delta s - \qquad (119)\\ &2\sum_i \left((x_i - \hat{x}_i)t_x + (y_i - \hat{y}_i)t_y\right) \Delta s + \qquad (120)\\ &\sum_i \left(t_x^2 + t_y^2\right) \Delta s. \qquad (121) \end{aligned}$$

The first term is the curve distance metric without any translation and does not depend on (t_x, t_y). The other terms can be separated into terms that depend on either t_x or t_y. For t_x we have

$$-2t_x \sum_i (x_i - \hat{x}_i)\, \Delta s + t_x^2 \sum_i \Delta s. \tag{122}$$

This is minimised for

$$t_x = \frac{\sum_i (x_i - \hat{x}_i)\, \Delta s}{\sum_i \Delta s} = \overline{x} - \overline{\hat{x}}, \tag{123}$$

where the mean x coordinates are defined as

$$\overline{x} = \frac{\sum_i x_i\, \Delta s}{\sum_i \Delta s}. \tag{124}$$

Note that $\sum_i \Delta s$ is the total length of the curve (assumed to be the same for the reconstruction and the reference for simplicity). Analogously, we find

$$t_y = \frac{\sum_i (y_i - \hat{y}_i)\, \Delta s}{\sum_i \Delta s} = \overline{y} - \overline{\hat{y}}. \tag{125}$$

We can therefore find the translation (t_x, t_y) that gives the best match with very little extra calculation. The mean position can be pre-calculated for each reference curve. The minimal curve distance metric can be obtained directly by translating both curves so that they have the same mean position (e.g., $(0, 0)$ or the mean position of the reference curve). The position of the reconstructed curve that gives the best match (smallest distance metric) could be compared to information from other sources (e.g., GPS for initial, final or intermediate positions of the train). Note that if $\overline{x} = \overline{\hat{x}}$ and $\overline{y} = \overline{\hat{y}}$, we have

$$d^2(t_x, t_y) = d^2(0, 0) + (t_x^2 + t_y^2). \tag{126}$$

(5.16) Scale

We assume that the scale of the reference curve is correct. If we do not know the scale of the reconstructed curve, then we can find the scale factor that minimises the curve distance metric. Note that we can also use other information such as known positions along the journey (e.g., GPS) or the total distance travelled to calibrate the k_η parameter, which determines scale.

Define

$$d^2(\lambda) = \sum_i \left((\lambda x_i - \hat{x}_i)^2 + (\lambda y_i - \hat{y}_i)^2 \right) \Delta s. \tag{127}$$

The new distance metric is

$$d = \min_\lambda d(\lambda). \tag{128}$$

Expanding the brackets gives

$$d^2(\lambda) = A\lambda^2 - 2B\lambda + C, \tag{129}$$

where

$$A = \sum_i \left(x_i^2 + y_i^2 \right) \Delta s, \quad B = \sum_i \left(x_i \hat{x}_i + y_i \hat{y}_i \right) \Delta s, \quad C = \sum_i \left(\hat{x}_i^2 + \hat{y}_i^2 \right) \Delta s. \tag{130}$$

The minimum is reached for $\lambda = B/A$ and the minimum value is given by $d^2 = -B^2/A + C$.

Note that the quantities A, B and C were already used in the efficient calculation based on cross-correlation.

For $\lambda = 1$ (unscaled distance metric) we have

$$d^2(1) = A - 2B + C. \tag{131}$$

We can use this to eliminate B in the distance metric with scale, which gives

$$d^2(\lambda) = A\lambda^2 - (A + C - d^2(1))\lambda + C. \tag{132}$$

The optimal scale factor is now

$$\lambda = \frac{A + C - d(1)}{A} \tag{133}$$

and the optimal distance metric is given by

$$d^2 = -(A + C - d(1))^2/A + C. \tag{134}$$

This shows that one unscaled distance metric calculation can be used to find the scaling factor that gives the best match. The only extra calculations are for A and C. The quantity A only depends on the reconstructed curve and the quantity C only depends on the reference curve.

Alternative Notations:

$$A = (\mathbf{x}, \mathbf{x}) + (\mathbf{y}, \mathbf{y}) = \overline{\mathbf{x}^2} + \overline{\mathbf{y}^2}, \qquad B = (\mathbf{x}, \hat{\mathbf{x}}) + (\mathbf{y}, \hat{\mathbf{y}}), \qquad C = (\hat{\mathbf{x}}, \hat{\mathbf{x}}) + (\hat{\mathbf{y}}, \hat{\mathbf{y}}) = \overline{\hat{\mathbf{x}}^2} + \overline{\hat{\mathbf{y}}^2} \tag{135}$$

$$A = \|X\|_F = tr(X^T X), \qquad B = \mathrm{tr}(\hat{X}^T X), \qquad C = \|\hat{X}\|_F = \mathrm{tr}(\hat{X}^T \hat{X}) \tag{136}$$

(5.17) Orientation

The orientation that minimises the distance metric can be found by reformulating the problem as an Orthogonal Procrustes Problem Schnemann (1966); Kabsch (1976)

It is convenient to use matrix notation to formulate this problem. Let X and Y denote the matrices containing the coordinates of the curves, i.e.,

$$X = \begin{bmatrix} x_0 & \cdots & x_{n-1} \\ y_0 & \cdots & y_{n-1} \end{bmatrix}, \qquad \hat{X} = \begin{bmatrix} \hat{x}_0 & \cdots & \hat{x}_{n-1} \\ \hat{y}_0 & \cdots & \hat{y}_{n-1} \end{bmatrix}. \tag{137}$$

The Constrained Orthogonal Procrustes Problem aims to find a orthogonal matrix Q ($Q^T Q = I$), with $\det(Q) = 1$ (to exclude reflections, so that Q is in fact a rotation matrix), which minimises

$$d^2(Q) = \|\hat{X} - QX\|_F^2. \tag{138}$$

The subscript F denotes the Frobenius matrix norm, which means this is the same as the curve distance metric we used before, but now between the reference curve and the curve rotated by the matrix Q.

The solution can be found by singular value decomposition Schnemann (1966); Kabsch (1976). Compute the SVD of the 2×2 matrix $X\hat{X}^T = U\Sigma V^T$. The rotation matrix is given by

$$Q = V\Sigma' U^T, \qquad \Sigma' = \begin{bmatrix} 1 & \\ & s \end{bmatrix}, \qquad s = \det(VU^T) = \det(U)\det(V). \tag{139}$$

The angle θ can be found from

$$Q = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{140}$$

5.4 3D Effects

(5.18) It could be suggested that the *cant* of the track does not have a great influence on the results.

We suggest that it may be possible to take into account the incline and decline of the track by “flattening” the 3D track information.

Suppose the map data for a track is given as a set of points (x_i, y_i, z_i), $i = 0, \ldots, N-1$. The goal is to turn this into a reference curve $(\hat{x}_i, \hat{y}_i)$, $i = 0, \ldots, n-1$ such that distances and curvatures along the curve are (approximately) preserved. For convenience the reference curve representation should also be equidistantly sampled (see curve matching section). This could be achieved by approximating distance s and curvatures κ along the curve using 3D formulas. (Or equivalently FD and SD, e.g., for $v = 1$.) The “flat” representation could then be obtained using the techniques described in the curve reconstruction section. Torsion could also be calculated, but it is ignored in the “flattening” process. The system could check that the values for torsion are small.

6 Mapping a train journey to a map

(6.1) Part of the problem is to determine where on the railway map of the UK a particular train journey is taking place, based purely on the train odometry extracted from the video. There are three considerations to take into account:

1. the state of the railway map, as extracted from OpenStreetMap (`www.openstreetmap.org`), discussed in section 6.1,
2. the features of the train’s journey that may allow it to be uniquely matched to the railway map, discussed in section 6.2, and
3. possible methods for performing the match, discussed in section 6.5.

6.1 The railway map

(6.2) The railway track is defined as a series of 42 636 segments, each made up of a number of waypoints (minimum 1, maximum 555, mean 9). A waypoint is defined by its location, given in latitude and longitude, and may appear in more than one segment. Waypoints that appear in more than one segment may represent branches in the line, but they may just link non-branching segments together. Figure 16 shows the current railway map, as defined in the `GBTrackMap.csv` file provided by RDS.

(6.3) A single journey is assumed to consist of a start point and an end point and the track between these two points is visited only once during the journey. We would like, therefore, to match a journey to a plausible stretch of track containing no repeated sections and no branches. Given a randomly-selected waypoint, we may construct a single track by stitching together the segments from that point to either end of the track, randomly selecting branches to follow along the way. The end of the track is defined as any end-point in a segment that is not linked to any other segments. Figure 17 shows one possible track extracted in this way. Its position is shown in red in Figure 16 and each segment is coloured differently from its neighbours to highlight how the track is constructed. 1 000 random tracks were constructed in this way and Figure 18 shows histograms of the number of segments and the number of waypoints in each track.

(6.4) Analysis of the railway map highlights a number of potential issues:

- There are 311 segments that are not connected to any other segment. It is likely that only a small number, if any, of these are be correct. Figure 19 shows the area around King's Cross and St. Pancras stations in London; note that King's Cross is disconnected from the rest of the network because some of the segments (shown in red) are not connected to any other segment.
- Junctions in the map are directionless, that is, they do not record whether a train is physically able to take a particular branch. Figure 20 shows a section of track south of Exeter. There are a number of branches in the track, but there is no direction information, so nothing to determine whether a train is able to turn the sharp corner at, for example, the junction indicated by the blue arrow.
- Any waypoint associated with more than one segment may have more than one set of feature values associated with it. For example, the point indicated by the blue arrow in Figure 20 will have different curvature values associated with it depending on which route the train takes through the junction.
- A single station may be represented by multiple waypoints (for example, the two stations shown in Figure 19).

(6.5) It would also be very helpful for the map to include the locations of bridges

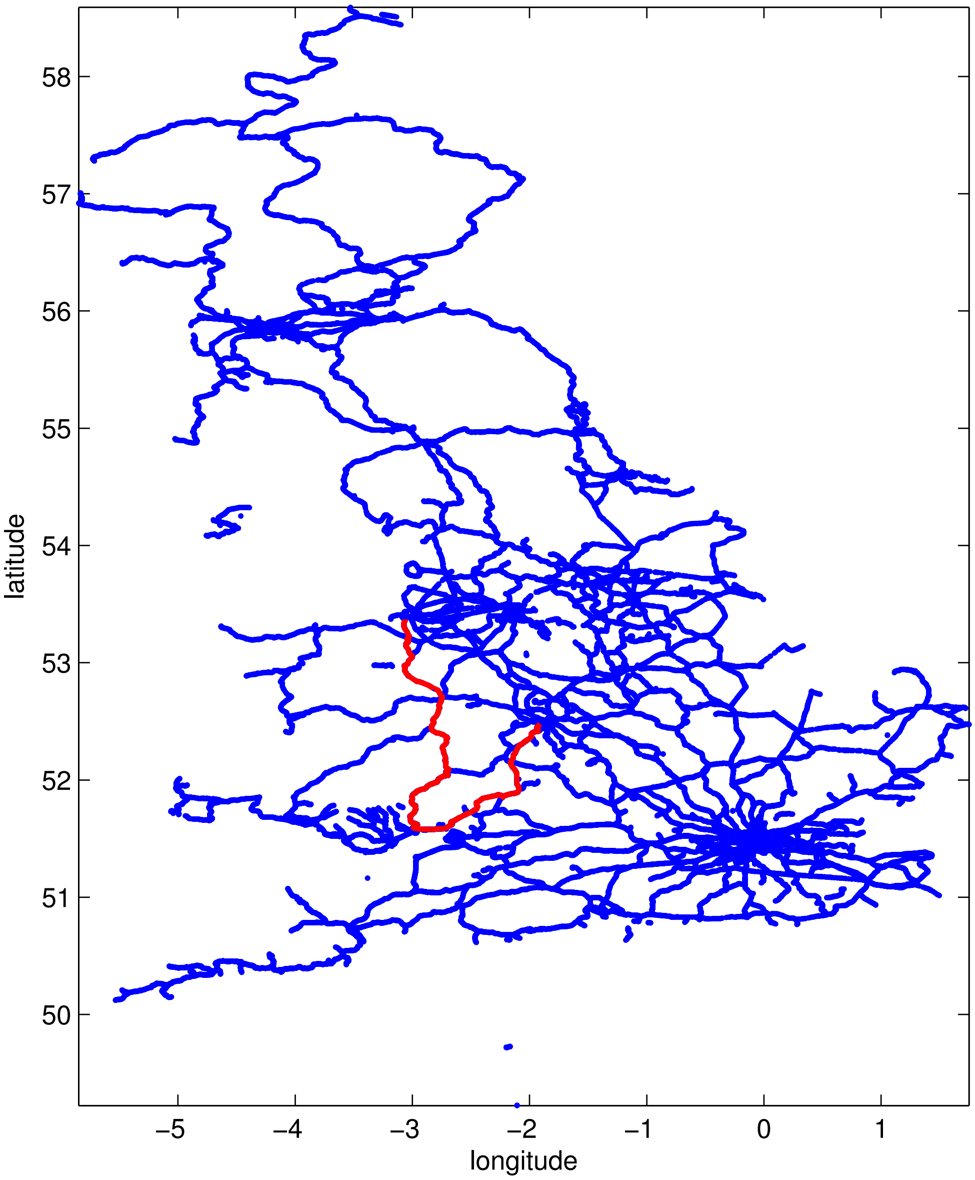


Figure 16: The railway map, as provided in the `GPTrackMap.csv` file provided by RDS and originally extracted from `www.openstreetmap.org`. The red section is the track segment shown in Figure 17.

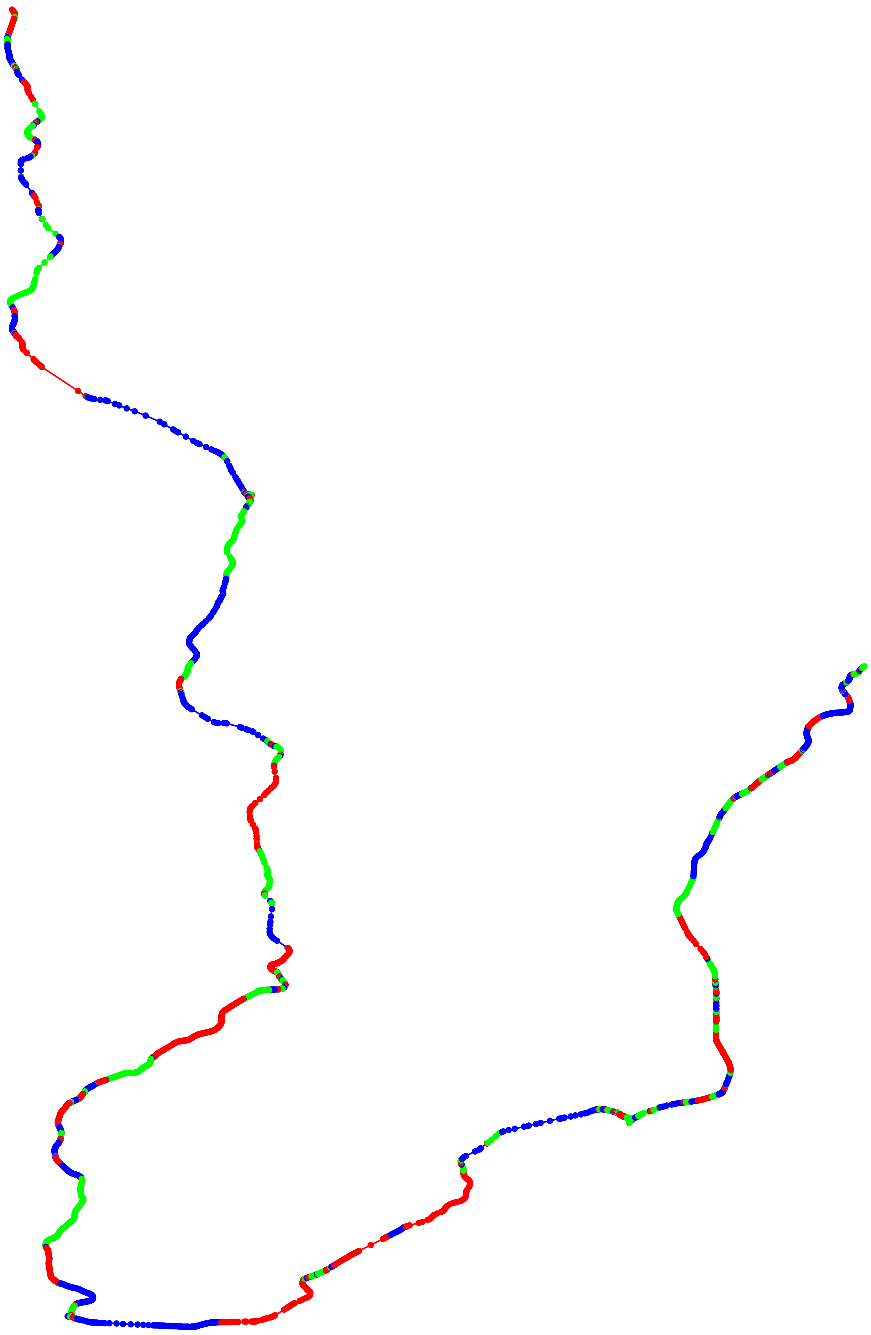
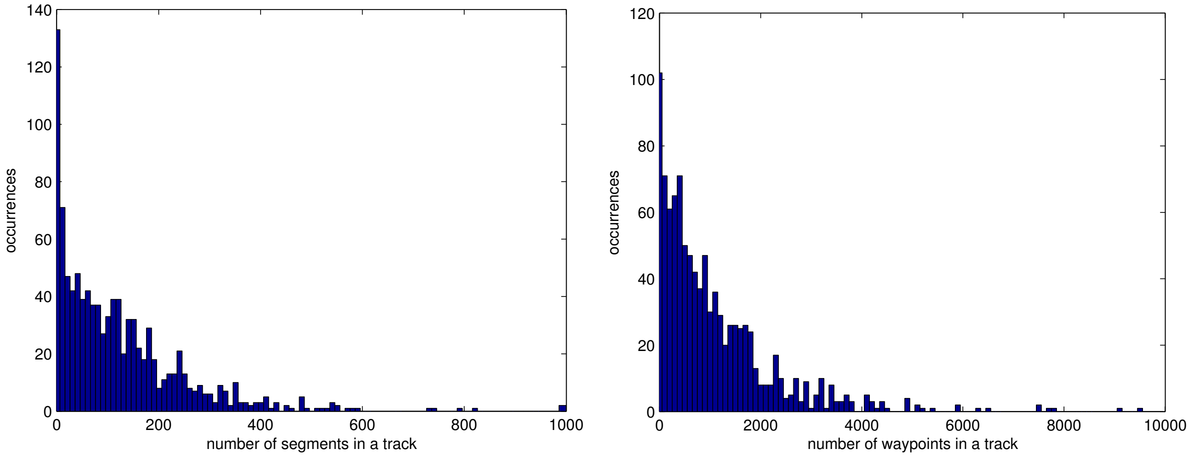


Figure 17: Example track showing the segmentation. This is the track shown in red in figure 16.



(a) segments per track

(b) waypoints per track

Figure 18: Histograms of (a) the number of segments and (b) the number of waypoints in 1 000 randomly-extracted tracks.

and tunnels, which would increase the number of features available to enable the localisation of a train journey to a particular section of track.

6.2 Train journey features

(6.6) A particular train journey is currently recorded as a series of forwards and sideways displacements, and a quality measure, at the frame rate of the video (in the range 25 to 65 frames per second). The quality measure is an integer in the range 0 to 225, with higher numbers representing higher confidence in the displacement values. The displacements are recorded in pixels, with pixel sizes estimated to be 19.1mm in the forward direction and 16.7mm in the sideways direction. Figure 21 shows the forwards and sideways displacements in green, plotted over the quality measure which is shown in grey. The continuous section of low quality towards the centre of these plots shows where the track goes through a tunnel.

6.3 Noise

(6.7) We can see from Figure 21 that the data are noisy. The effect of this noise is shown in Figure 22, which shows (in blue) the path allegedly taken by the train if we just ignore the noise and assume that the displacements are true. Applying a crude, 1 000-observation moving average to smooth out the outliers results in the equally unlikely path shown in red.

(6.8) A common way to deal with the noise and estimate the underlying true displacement values is to use a Kalman Filter (or a Kalman Filter/Smoother if we are able to see into the "future"). This assumes that the noise follows a Gaussian

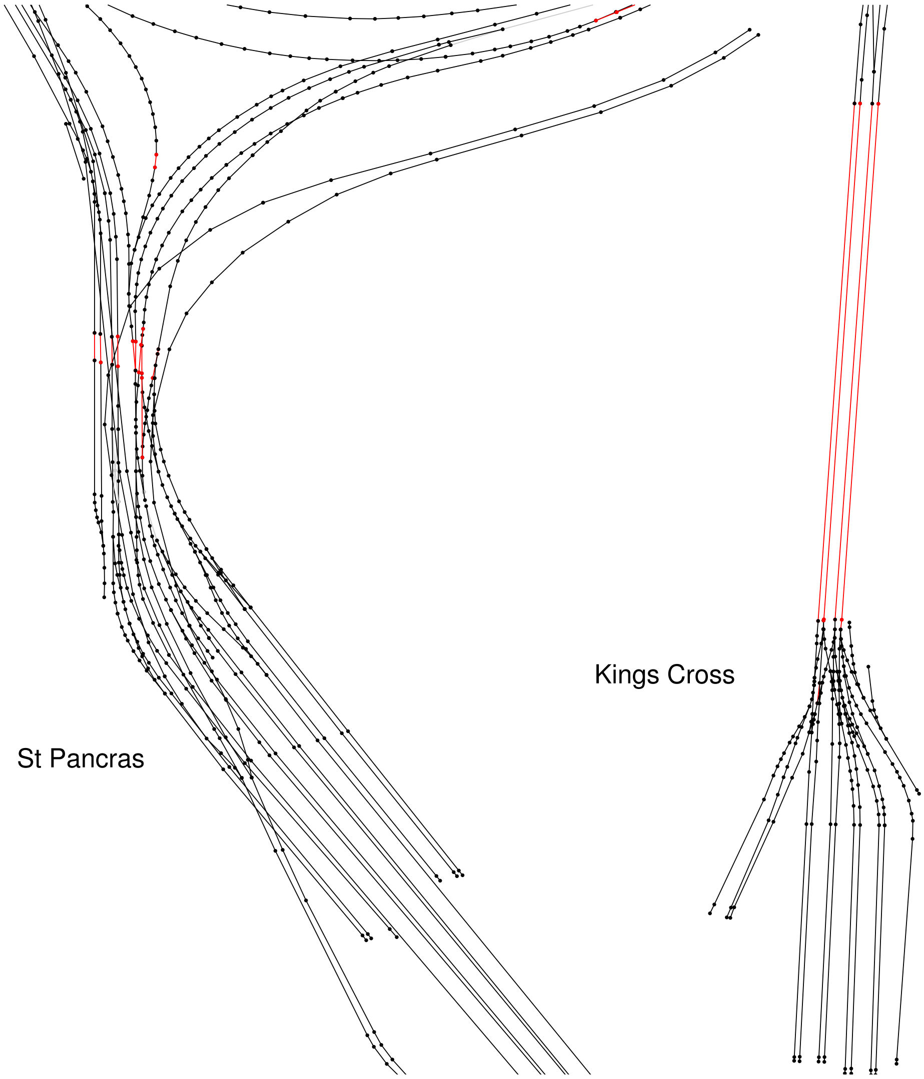


Figure 19: The railway map of the area around King's Cross and St. Pancras stations in London. The red sections are segments that are not linked to any other segments; note that this includes the track connecting Kings Cross station to the rest of the network.

(Normal) distribution. However, the Gaussian is known to be badly affected by outliers and, ignoring the tunnel section, we can see that the forwards displacements in particular are subject to occasional, but rather large excursions from the range of values that seem likely given the adjacent data, which we would consider to be outliers.

(6.9) Gaussian models, such as the Kalman Filter, are known to be particularly sensitive to outliers. Figure 23 demonstrates the effect of a small number of outliers on the estimation of the parameters of a Gaussian distribution. Plot (a) shows a histogram of 30 samples taken from a Gaussian distribution with mean zero and

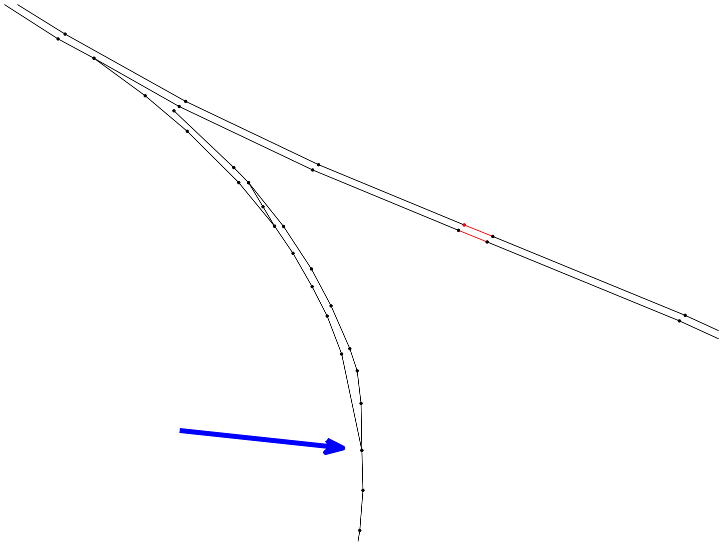


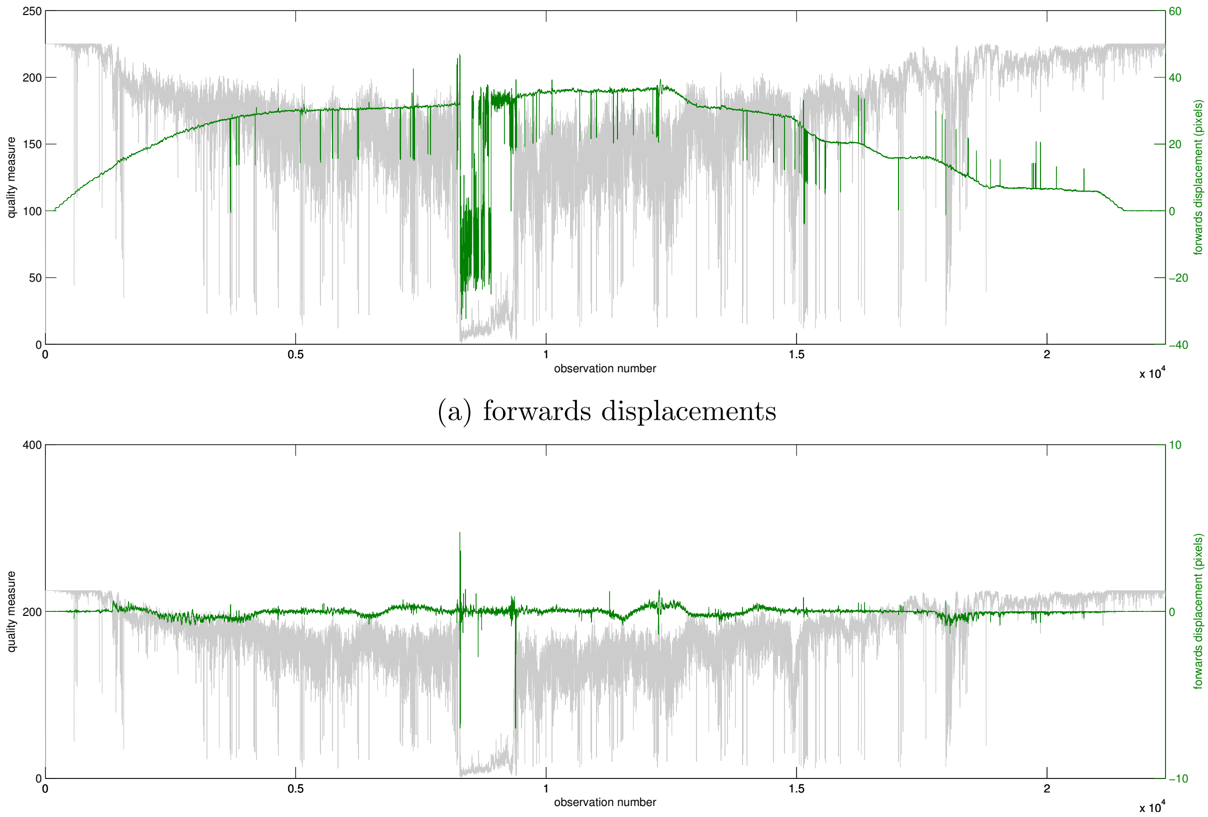
Figure 20: A section of the railway map south of Exeter. There are a number of branches in the track, but there is no direction information, so nothing to determine whether a train is able to turn the sharp corner at, for example, the junction indicated by the blue arrow.

standard deviation of 0.5. Overlaid in green is the Gaussian distribution estimated from these data. Plot (b) shows the same, but with the addition of three outliers. Notice how both the mean (location of the centre) and the variance (width) have been substantially adversely affected. There are many different ways of dealing with outliers. Following Lange et al. (1989), we could replace the Gaussian noise model with one based on the Student-t distribution. This enables the outliers to be absorbed into the noise without a significant impact on the shape of the distribution. This effect is shown in Figure 23, where the red plots show the Student-t equivalents of the green Gaussian plots.

(6.10) A type of Kalman Filter (or Kalman Filter/Smoother) that assumes the noise to be distributed according to a Student-t distribution, rather than Gaussian, has been developed (journal paper in preparation), but a small amount of work will be required to apply it to this particular problem.

6.4 Extracting features

(6.11) In order to match a journey to a section of track, we need to extract some features from it to make a unique "fingerprint" through which the matching may be performed. We do not know in an absolute sense where a journey is (we are not allowed to use GPS); the features we choose to try and match to the railway map can only be extracted from the video data. Curvature of the track is the most obvious choice (see sections 5 to 5.3) as it can be inferred from the recorded displacements. It has been recognised that pitch and roll of the locomotive may contribute to erroneous displacement and curvature measurements, but it may



(a) forwards displacements

(b) sideways displacements

Figure 21: A particular train journey is recorded as a series of forwards and sideways displacements (shown in green), and a quality measure (in grey). These plots show the (a) forwards displacement and (b) sideways displacements for a train journey between Leartherhead and Dorking.

also be the case that yaw makes a contribution, given that the video camera is placed forward of the train's front wheels. As was pointed out in section 5.3, it is unlikely that curvature on its own will be able to distinguish between adjacent, parallel tracks.

(6.12) Other potential features that might be used in addition to curvature are the locations of

- points, identified by a particular pattern of displacement noise,
- speed limits, identified by a particular pattern of forward displacements,
- tunnels, identified by darkness in the video, or the switching of the video camera to night vision,
- bridges, identified by significant structure passing over the train, and
- (possibly) stations and signals, identified by periods when the train is stationary.

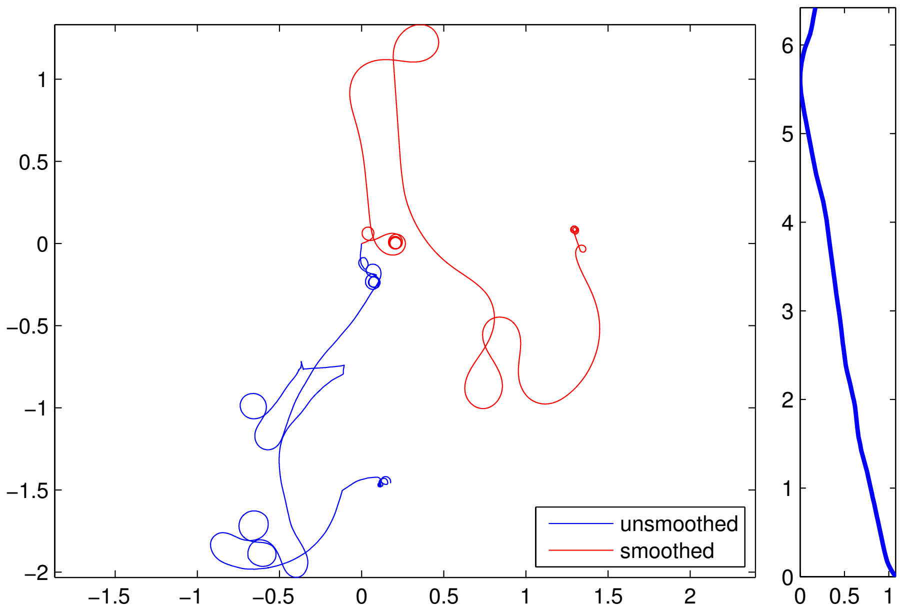


Figure 22: The train journey from Leatherhead to Dorking, as calculated by dead reckoning based on the original forwards and sideways displacements (shown in blue) and by crudely smoothed displacements (in red). The crude smoothing is achieved by applying a 1 000-observation moving average. The true track is shown on the right. The distances are shown in kilometres.

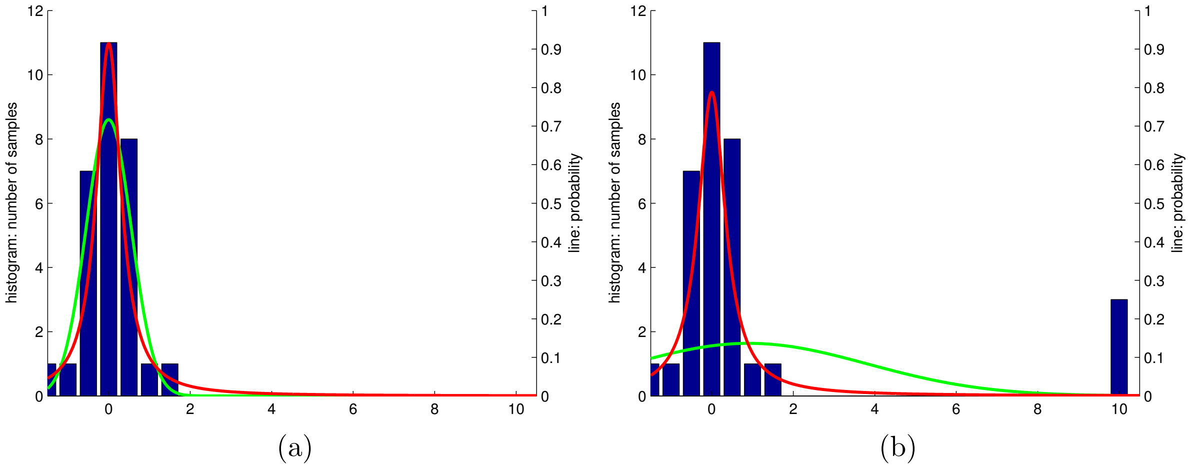


Figure 23: Demonstration of the effect on the maximum likelihood solution for Gaussian and Student-t distributions of a small number of outliers. Plot (a) shows a histogram of 30 samples taken from a Gaussian distribution with mean zero and standard deviation of 0.5. Overlaid in green is the Gaussian distribution estimated from these data, and in red is the Student-t distribution similarly estimated. Plot (b) shows the same, but with the addition of three outliers. Notice how both the mean and the variance have been adversely affected in the Gaussian case, but the Student-t distribution is largely unaffected.

but these would be dependent on knowing the true locations of these physical objects on the railway map.

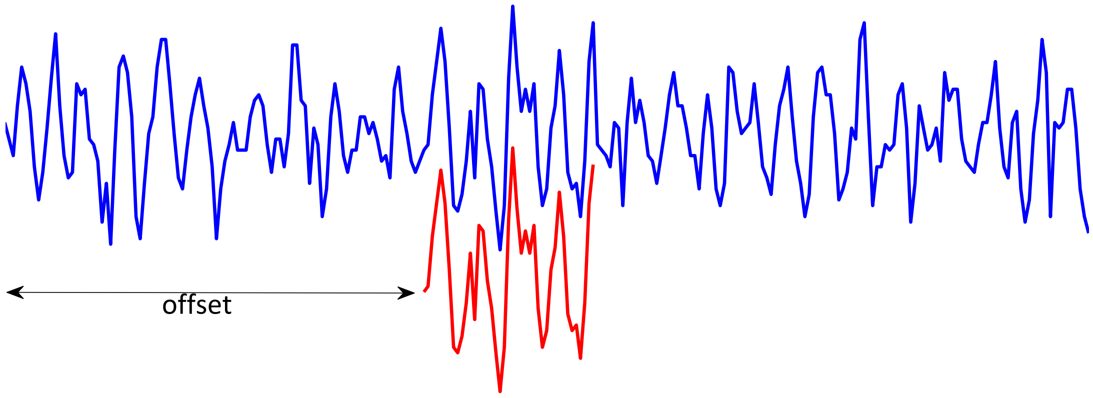


Figure 24: Two example one-dimensional datasets that are matched when the marked offset is applied to the shorter dataset. We might imagine that the red plot represents the features of a train journey and the blue plot a section of track.

6.5 Matching a journey to the map

(6.13) Given a set of journey features and the same set of features extracted from a section of track, we would like a method of matching the journey to a subset of that track. Given a pair of one-dimensional sets of time-series data, for example, cross-correlation (referred to in section 5.3) calculates the time offset that maximises the correlation between the two traces by finding the x that maximises

$$\int S_1(t+x)S_2(t)\ dt \tag{141}$$

where $S_1(\cdot)$ and $S_2(\cdot)$ are the two traces and t is the timestamp index. This is quickly and conveniently calculated in one step using the Fast Fourier Transform (FFT). If $\mathcal{S}_1$ and $\mathcal{S}_2$ are the FFTs of S_1 and S_2 respectively, and the asterisk denotes complex conjugation, then the location of the maximum of

$$\mathcal{S}_1\mathcal{S}_2^* \tag{142}$$

enables the calculation of the offset x. Figure 24 shows two example one-dimensional datasets that are matched when the marked offset is applied to the shorter dataset. We might imagine that the red plot represents the features of a train journey and the blue plot a section of track.

(6.14) In this example the match is straightforward because the short section is an exact copy of a portion of the longer dataset. The method will still work with a certain amount of noise in the measurements of the features (i.e. noise in the y-axis), but it is unlikely to work if the journey features are compressed or dilated (or made up of a number of sections with different compressions and dilations) in time compared with the track. In this case we would like to generalise the expression for cross-correlation in (141) as follows Belmont and Hotchkiss (1997); Belmont et al. (1997); Belmont and Jardon (2000):

$$\int S_1(t + \mathrm{f}(t))S_2(t)\ dt \tag{143}$$

where $\mathrm{f}(t)$ is some function of t that allows for localised compression and dilation. The function needs to be constrained; allowing any function will enable any journey to be matched to any section of track. Uchida and Sakoe (2005) provide a survey of other *elastic matching* techniques. Muller (2007) describes *dynamic time warping* and Thanawin et al. (2013) provide a recent paper on dynamic time warping for big time-series data.

7 Dynamic train tracking

7.1 The forward model

(7.1) We implemented the Sequential Importance Resampling (SIR) filter of Gordon et al. (1993). To do so, a forward model of the train, f, moving in relation to the link-node network of the rail system was created.

(7.2) The state vector, $x \in \mathbb{R}^4$, was defined as follows:

$$x = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} = \begin{bmatrix} \text{Train line index} \\ \text{Distance along train line } x^1 \\ \text{Distance moved since last assimilation step} \\ \text{Current track curvature} \end{bmatrix}. \tag{144}$$

(7.3) In order to apply a particle filter, it is necessary to have a forward dynamical model of the system. To do so, we created an artificial rail network similar to that given in the example of a train approaching a station. The layout of the track was as shown in Figure 25.

(7.4) The network was split into sections with $\mathcal{O}(1)$ length. When a train is within a small distance of the junction, it will change lines randomly with probability 0.5. A database of piecewise linear tracks was created to store the curvature of the tracks, which is looked up depending on the train position at the given iteration. At each iteration, the train will move a random $\mathcal{O}(1)$ distance along the network. The main lines were numbered 1–4 and the link tracks were numbered 5–10.

7.2 Observations for particle filter

(7.5) The data available for the particle filter are the *df* and *ds* measurements coming from the video camera. We do not use them directly in a frame-by-frame basis.

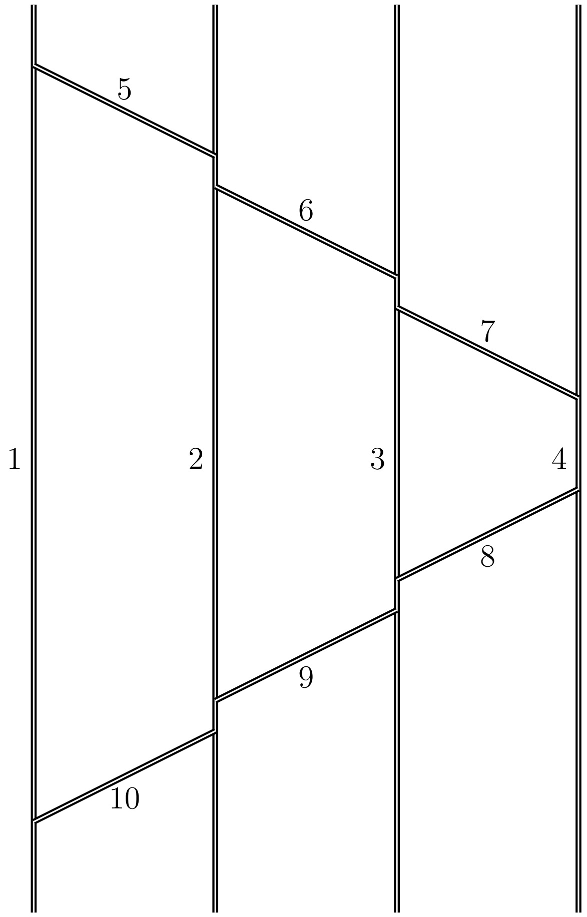


Figure 25: Artificial rail network

Instead, we group together multiple frames so that we have $\bar{df} = \mathcal{O}(1)$, i.e. we compute the sum of K frames of information so that,

$$y = \begin{bmatrix} \bar{df} \\ \bar{ds} \end{bmatrix} = \sum_{j=1}^{K} \begin{bmatrix} df \\ ds \end{bmatrix}. \tag{145}$$

(7.6) To use the SIR filter, we must know the structure of the errors in observations. For simplicity we assume that these errors are Gaussian with covariance matrix R. i.e.

$$y = \begin{bmatrix} \bar{df}_{\text{true}} \\ \bar{ds}_{\text{true}} \end{bmatrix} + R^{\frac{1}{2}}\varepsilon \tag{146}$$

where $\varepsilon \sim \mathcal{N}(0, I)$. Hence the errors in the observations y are $\mathcal{N}(0, R)$.

(7.7) This Gaussian choice was made for simplicity and is not necessary for the method. If another distribution is more suitable, it can be used provided the equation to compute the likelihood of each particle is adjusted accordingly.

(7.8) With this observation structure, we must define the observation operator H, which will map the state vector into observation space. In this case, H is given by

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{147}$$

7.3 The assimilation method

(7.9) We assume that we have a prior distribution for where the train is. Typically this will be anchored at the start of the journey in a station, or the track matching methods discussed in Section 6.5 could be used to obtain this.

(7.10) We will propagate forward in time an ensemble of particles, each of which represents the same train. The initial ensemble is created by generating multiple copies of a particle at the known initial point.

(7.11) The SIR filter algorithm is used to assimilate the data, a pseudocode for which is given below.

```
• Create initial ensemble $\{x_i,\ i = 1, \dots, N\}$.
• Run the following steps.
    1. Apply the forward model $x = f(x)$
    2. Read in the data $y$
```

3. Compute the weights of each particle

$$w_i = \exp(-\tfrac{1}{2}(y - H(x_i))^T R^{-1}(y - H(x_i))) \tag{148}$$

4. Resample the particles based on the weights w_i.
5. Repeat steps.

(7.12) Resampling is done by implementing stochastic universal sampling Baker (1987).

7.4 The octave (matlab) code

(7.13) The octave code is designed to run a twin experiment. Firstly, it is run with the logical flag `gen_data` set to `true`. This will simulate a train running through the network and output files containing simulated observation data at each timestep.

(7.14) Secondly, the code is run again with `gen_data` set to `false`. This will use the simulated data in the particle filter to constrain the particles to follow the same path as created the original data.

7.5 Results

(7.15)

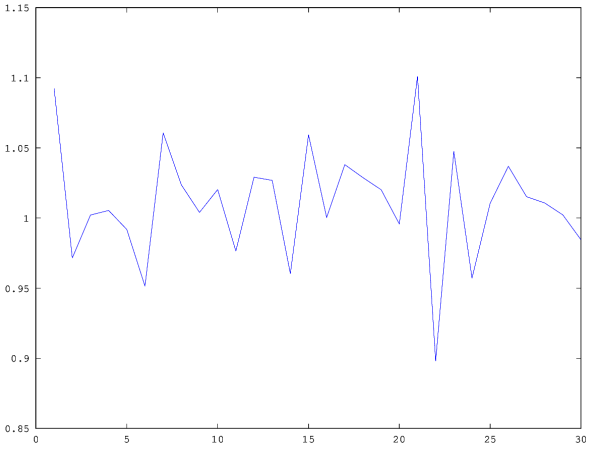


Figure 26: Simulated forward motion data $\bar{df}$

(7.16) Figures 26 and 27 show the simulated data used as observations for the twin experiment. Note the realistic signature in the curvature data of track switching. Running an ensemble of 4 members ensured that the ensemble mean precisely followed the path of the *truth run* used to generate the data. The results are shown in Figure 29.

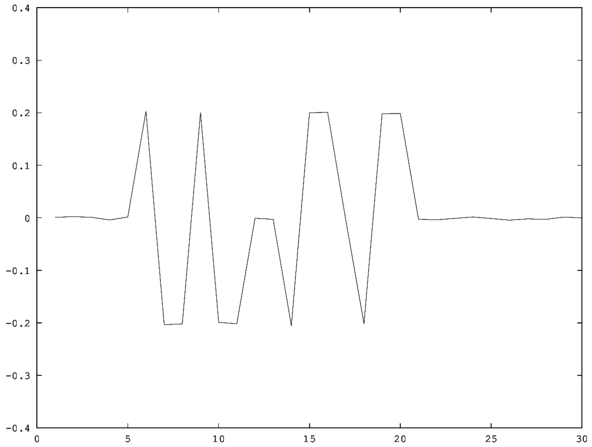


Figure 27: Simulated curvature data $\bar{ds}$

(7.17) Note that in doing so, errors accumulated by the forward *df* measurements are reset each time there is a large change in the track curvature data.

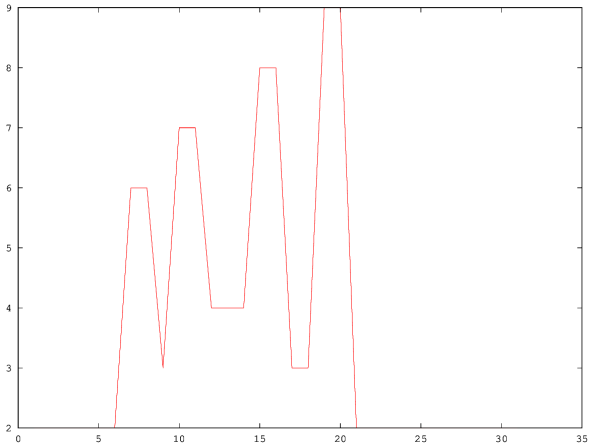


Figure 28: Track number of train used to create observations

8 Drawing a track map

8.1 The constrained optimization model

(8.1) In calculating the real-time location of a train, having an accurate track map is fundamental. We look into building a track map from information gathered through camera and GPS on routine train runs. We do not consider precise equipment such a survey-grade differential GPS equipment, but rather standard GPS equipment. Such cheap mapping techniques are also important when alterations are made to the train tracks, as one can obtain new train maps inexpensively.

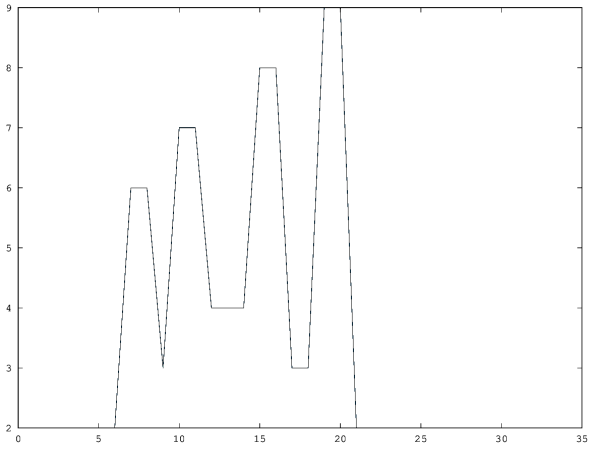


Figure 29: Ensemble of particles when simulated observation data has been assimilated into the model. Note there is no spread in the ensemble with regards which line the particles are on - all the trains are on the right track!

(8.2) Our approach is that of a constrained and regularized least-squares to draw a track from a single run. Later in the conclusion, we suggest how to use techniques in the map inference literature to combine solutions from repeated runs of the same route.

(8.3) Let $u : \mathbb{R}^n \to \mathbb{R}^2$ be our desired track map, described as a sequence of Latitude and Longitude coordinate pairs. Let $\bar{u}$ be our collected data that has some unknown noise. The optimization problem that we use to regularize this data and assist in determining the true track u is given by

$$\min_u F(u, \bar{u}) + R(u)$$
$$s.t \text{ Physical Track constraints,}$$

where $F(u, \bar{u})$ is the fidelity function which measures the distance between the data and true track and $R(u)$ is the regularization function that penalizes noise present. We minimize the Fidelity and Regularizer functions subject to physical track constraints.

(8.4) Our choice of fidelity function is a sum of weighted L_2 norms. Let $w \in \mathbb{R}^n$ be weights that determine our confidence in the measured data $\bar{u}$. The weighted L_2 norm is defined as

$$\|u - \bar{u}\|_{L_2(w)} = \sqrt{\sum_{i=1}^{n} w_i (u_i - \bar{u}_i)^2}.$$

As a short-hand we use $\|u - \bar{u}\|_{L_2} := \|u - \bar{u}\|_{L_2(\mathbf{1})}$, where $\mathbf{1}$ is a vector of ones of the appropriate size. For each set of observed data $\bar{u}$, we add a L_2 norm to the fidelity function that measures the distance between the solution u and $\bar{u}$. In the case of the GPS readings $\bar{u}_{GPS} \in \mathbb{R}^{2n}$, we add the term $\mu_{GPS} \|u - \bar{u}_{GPS}\|_{L_2}$ to the fidelity function. On the other hand, the sequence of forward and sideways dislocations $\bar{d}_{odo} = (df, ds) \in \mathbb{R}^{2n}$ as provided by the odometry videos, are not directly comparable to that of the solution u which is in latitude and longitude coordinates. Instead, the dislocation pairs (df, ds) represent the tangent vector in local coordinates. To remedy this, we apply a transformation $T_{loc} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ that receives coordinates longitude and latitude coordinates u and returns the tangent field in local coordinates. See subsection 10.1 for details.

(8.5) In choosing a regularizor, we set out to promote the inertia of train and its incapacity to make sharp turn. The TV (*total variation*) norm does just this by minimizing changes in the gradient, thus "encouraging" the train to maintain the same direction

$$\begin{aligned}\|\nabla u\|_{TV} &= \sqrt{\sum_{i=1}(\Delta u_{i+1} - \Delta u_i)^2} \\ &= \sqrt{\sum_{i=1}(u_{i+2} - 2u_{i+1} + u_i)^2},\end{aligned}$$

where $\Delta u_i := u_{i+1} - u_i$.

(8.6) The physical constraint we considered in this initial experiment is to bound the curvature of the track. Turning to the UK Track System Requirements, the minimum radius for the tracks is given by $R_{\min} = 125m$. From each sequences of three points in our discretized track u, we can find the radius of the circle that intersect them with the formula

$$\begin{aligned}f_R(\Delta u_i, \Delta u_{i+1}) &= \frac{\|\Delta u_i\| \, \|\Delta u_{i+1}\| \, \|\Delta u_i + \Delta u_{i+1}\|}{2|\Delta u_i \times (\Delta u_i + \Delta u_{i+1})|^2} \\ &= \frac{\|\Delta u_i\| \, \|\Delta u_{i+1}\| \, \|\Delta u_i + \Delta u_{i+1}\|}{2|\Delta u_i \times \Delta u_{i+1}|^2} \\ &= R.\end{aligned}$$

As large portions of the track have infinite radius, namely the line segments, we choose to bound curvature $\kappa = 1/R$ which is bounded between 0 and $\kappa_{\max} := 1/R_{\min}$. Combining the above, we have the nonlinear constrained optimization problem

$$\begin{aligned}&\min_u \mu_{GPS} \|u - \bar{u}_{GPS}\|_{L_2} + \left\|T_{loc}(u) - \bar{d}_{odo}\right\|_{L_2(w_{odo})} + \mu_{TV} \|\nabla u\|_{TV} \\ &\quad s.t \quad 1/f_R(\Delta u_i, \Delta u_{i+1}) \le \kappa_{\max}, \quad i = 1, \ldots n-1. \qquad (149)\end{aligned}$$

8.2 Numeric Tests

(8.7) We extracted GPS and odometry data, from the data set "LHD_DKG_data" provided by RDS, and implemented the objective function and constraint (149) as functions in Matlab. To solve (149), we used the fmincon function with the Interior-point algorithm option. The initial track map we use as input for the solver is a linear interpolations of the GPS points.

(8.8) The discretization used for u is a refinement of the GPS mesh $\bar{u}_{GPS}$ and $\bar{d}_{odo}$. A moving average filter with a window of 100 was applied to the odometry data in an attempt to remove some of the noise present, Figure 30. In Figure 31a is the plot of the extracted GPS data, though altered so that $(0,0)$ is the starting point of the train and transformed to km by the Haversine formula[1]. The gap in the path from $x = 2.6$km to 3.3km is due to a tunnel. Turning to approximately $x = 130$s in Figure 30, this tunnel also affected the odometry reading. The GPS signal was also lost around $x = 6km$.

(8.9) In Figures 31b, 32 and 33 we have plotted different solutions by incorporating different parts of the model (149) as to appraise their affects on the solution. By incorporating the TV-norm and the GPS Fidelity term, In Figure 31b, the solution interpolates the GPS points and also closes the gaps of lost signal with a smooth curve. Additionally using the curvature restriction, the solution in Figure 32 has an apparent displacement from the GPS points. Visually looking at the trail map from Dorking to Leatherhead, there is displacement between the true trap and GPS curve. Finally, incorporating the odometry data in Figure 33, the time to find a solution surpassed an hour on our Intel i5 1.7Ghz personal laptop. The solver returns a feasible point (one that satisfies the curvature constraint) with improved objective function over the initial track map.

9 Conclusions and Future Work

(9.1) It is apparent that there are multiple directions along which the aforementioned works could be enhanced and improved. This will constitute future work, resulting in an improved video odometry system that will yield operational and technical improvements over the existing system. Some of these ideas are as follows:

9.1 Enhancements to the work presented in Section 4

(9.2) The method implemented here is not readily applicable to a realistic scenario, as was noted earlier. Work has to be done on how best to choose a measure of distance between two curves in the camera image. The assertion made that“the combined errors introduced in theta, phi, psi, and camera position Y and Z have the same effect as a 2D rotation and translation” needs to be verified. If it checks out to be true then one can build a basis for 2D rotation and translation to use

[1] http://en.wikipedia.org/wiki/Haversine_formula

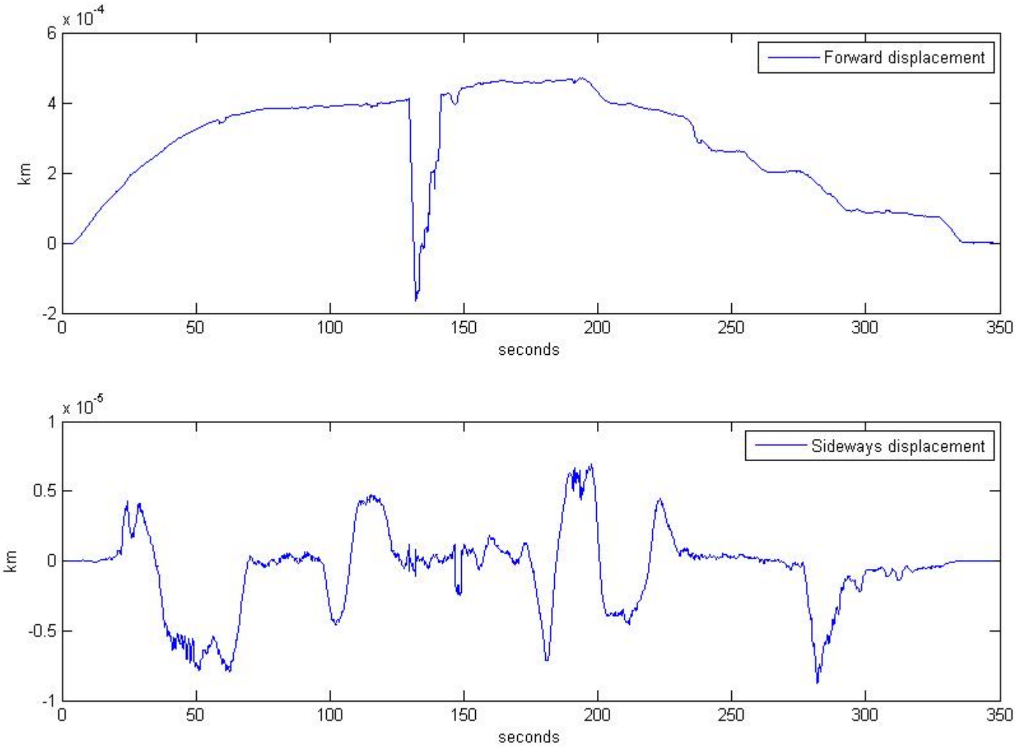
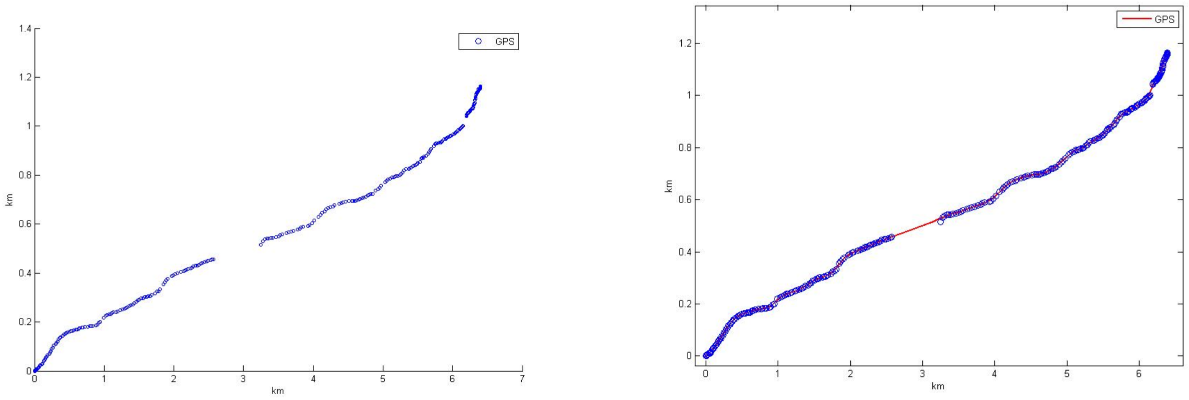


Figure 30: The forward and sideways displacement extracted from odometry.



(a) The GPS coordinates in Kilometers, starting point set arbitrarily to zero

(b) The solution to $\min_u \|u - \bar{u}_{GPS}\|_{L_2(w)} + \|\nabla u\|_{TV}$

in the method detailed. Finally there would then be the possibility to use the determinant that appears in the method to study optimal position of the camera.

9.2 Enhancements to the work presented in Section 5

(9.3) We believe that more thought could go into the work described in this section. In particular: (a). What signal would be best to match to – curvature, (x, y), or dSD and dFD? (b). What qualities should a signal have to make this signal matching type of method work? (c). How could these qualities be measured? E.g. entropy or surprise of a signal. (d). Include the quality measure Q in the calculation of the cross correlation. (e). Investigate elastic matching and dynamics warping to make the cross-correlation more adaptable to changes in the signal between different runs.

(9.4) Angle Measurement

We suggest that in addition to FD, SD and Q the system could also measure the angle of the rails. This could be done either in the warped or unwarped image. It would probably be good to exclude the parts of the image away from the rails.

It should be possible to use image processing to detect and isolate the rails (also the sleepers). The angle of the track in the image is related to the curvature of the track. This measurement can be taken from a single frame (FD and SD require comparing two consecutive frames). Unless the lighting conditions are very bad, the rails should stand out very clearly and the measurement should be very reliable. This measurement should also be fairly independent of the small scale movements of the train.

The angle ϕ is assumed to be small ($\sin(\phi) \approx \phi$). We have $\sin(\phi) = \frac{L}{R} = L\kappa$, so that

$$\kappa \approx \frac{\phi}{L}. \tag{150}$$

In principle a parabola or a circle could be fitted instead of a line to obtain second order information directly from a single frame. However, this can only work if the camera is pointing far enough ahead of the train.

Note that the positions of the intersection of the rails with the edge of the image *do* depend on the small scale movements of the train. These positions may be useful, but their measurements can be expected to be quite noisy.

(9.5) Methods for Displacement Measurement

Techniques similar to those explained in the curve matching section could be used to find FD and SD. Their values could be determined so as to minimise the difference between two consecutive (unwarped) frames. It is possible to also include rotation, again by a constrained orthogonal Procrustes problem. Whereas the curves were represented by lists of points, an image can be thought of as a list of points with weights.

A distance metric for two images can be defined by the mean squared difference between the weights of corresponding points in the region where the two images overlap. For $M \times N$ images this can be calculated in $O(MN\log(MN))$ operations via a 2D FFT (see cross-correlation section). The weights of each pixel could be intensities (grey scale levels), but it would probably be better to do some image processing first (e.g., contrast enhancement, histogram equalisation, thresholding, edge detection, elimination of lighting effects, etc.) Some of these image processing steps could be done in Fourier space.

To handle noise better, it may be beneficial to consider more than 2 frames at a time. It is also possible to use a Kalman filter or other filtering techniques.

9.3 Other possible enhancements

(9.6) An interesting direction to take could be to build a software simulator for the video odometry system in action. This would enable RDS to investigate the effects of different types of trains, video positions, train motions, track characteristics, etc..

(9.7) With the information available from running once through a certain track (GPS and possibly odometry readings), one should use a constrained regularized least-squares routine to build a single viable track. From repeated runs of the same track, it is possible to have a collection of possible tracks. Next, one can borrow ideas from the“Road map inference” literature on how to combine these multiple track runs into a single viable track. When adapting these road inference techniques to train tracks, it will become necessary to incorporate the physical constraints of trains, e.g., different from road maps, there are no sharp corners in tracks.

10 Appendices

10.1 Drawing Track Map Appendix

(10.1) The transformation $T_{loc} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ from latitude and longitude coordinates to local tangent coordinates is composed of two steps. The first transforms Latitude and Longitude coordinates into kilometers on a North-South and East-West plane. The second step brings these North-West coordinates to the local coordinates used in the odometry reading. The odometry reading does not supply the “up and down” motion of the train, in contrast to Latitude and Longitude coordinates which specify a point in three dimensional space. Therefore to compare the forwards and sideways dislocations to Latitude-Longitude coordinates, we assume that the distance of the desired track is such that the curvature of the earth is negligible.

(10.2) With this simplification, we can assume that all track maps points are in the same tangent plane to the earth, with a common North-South and West-East bearing.

The difference in kilometers between Latitudes and Longitudes of sequential points is assumed to be in the North-South and West-East orientation, respectively. To calculate distance, we use the Haversine formula[2]. Specifically, given Latitude and Longitude coordinate pairs $u_{i-1} = (u^1_{i-1}, u^2_{i-1})$ and $u_i = (u^1_i, u^2_i)$, the resulting difference in kilometers on the North-South East-West plane is calculated by

$$\begin{aligned} \Delta u^1_i &= \text{Haversine\_Distance}\big((u^1_{i-1}, u^1_{i-1}), (u^1_{i-1}, u^2_i)\big) \\ \Delta u^2_i &= \text{Haversine\_Distance}\big((u^1_{i-1}, u^2_{i-1}), (u^1_i, u^2_{i-1})\big) \end{aligned}$$

(10.3) To transform from distance in kilometers on the North-South East-West plane to local coordinates we use the previous direction vector Δu_i to define the axis of a plane, then figure out the coordinates of u_{i+1} on this plane. In detail,

$$\Delta u^{loc}_{i+1} = \left(\frac{\langle \Delta u^{\perp}_i, \Delta u_{i+1} \rangle}{\|\Delta_i u\|}, \frac{\langle \Delta u_i, \Delta u_{i+1} \rangle}{\|\Delta_i u\|} \right),$$

where $\Delta u^{\perp}_i$ is a perpendicular vector to Δu_i, specifically $\Delta u^{\perp}_i = R(\pi/2)\Delta u_i$ where $R(\pi/2)$ is a rotation of $\pi/2$ in the clockwise direction. This transformation to Δu^{loc}_{i+1} brings u to the same coordinate system as $\bar{d}_{i+1}$ so that they are directly comparable.

[2] http://en.wikipedia.org/wiki/Haversine_formula

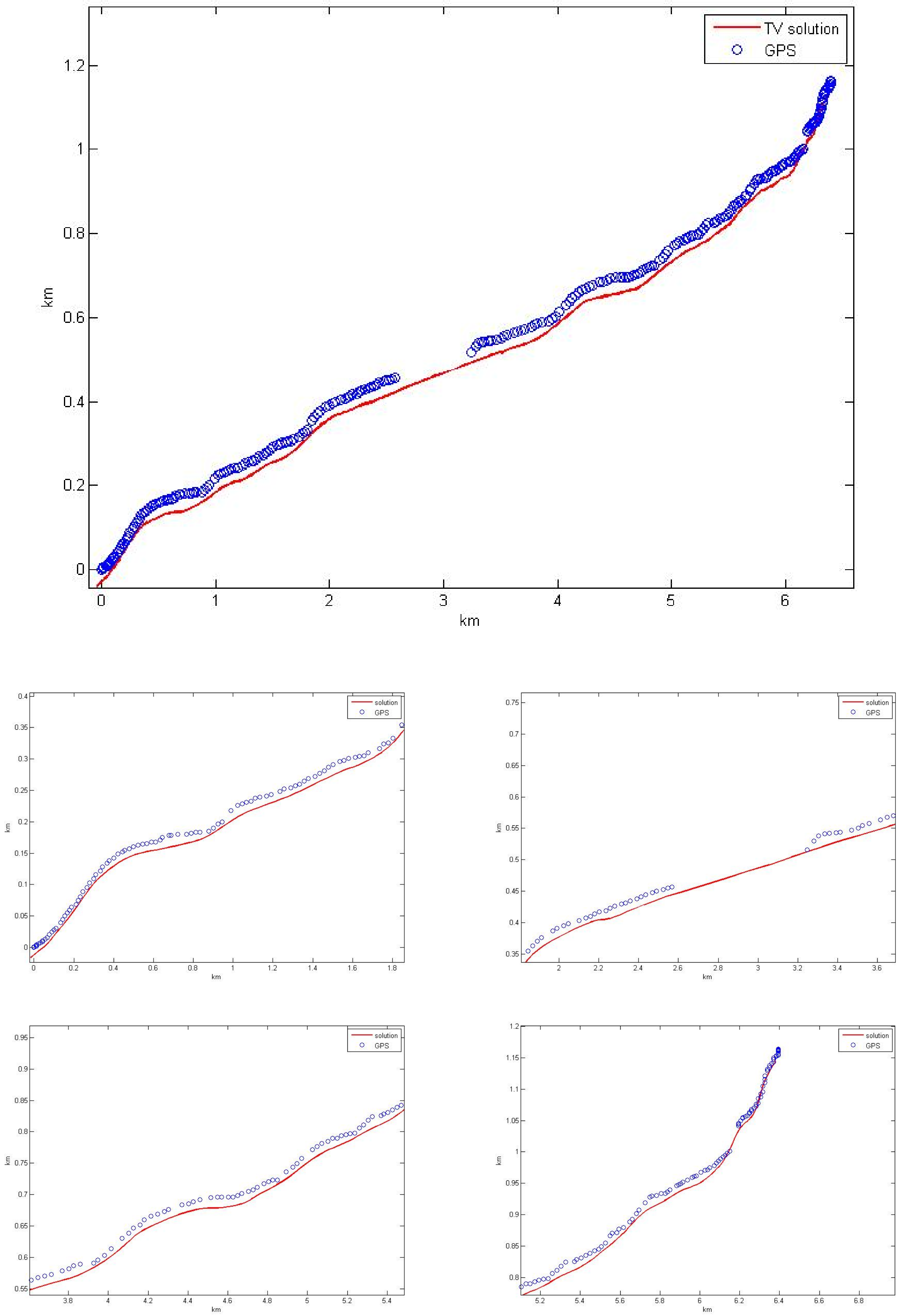


Figure 32: Four close-up frames solution to (149).

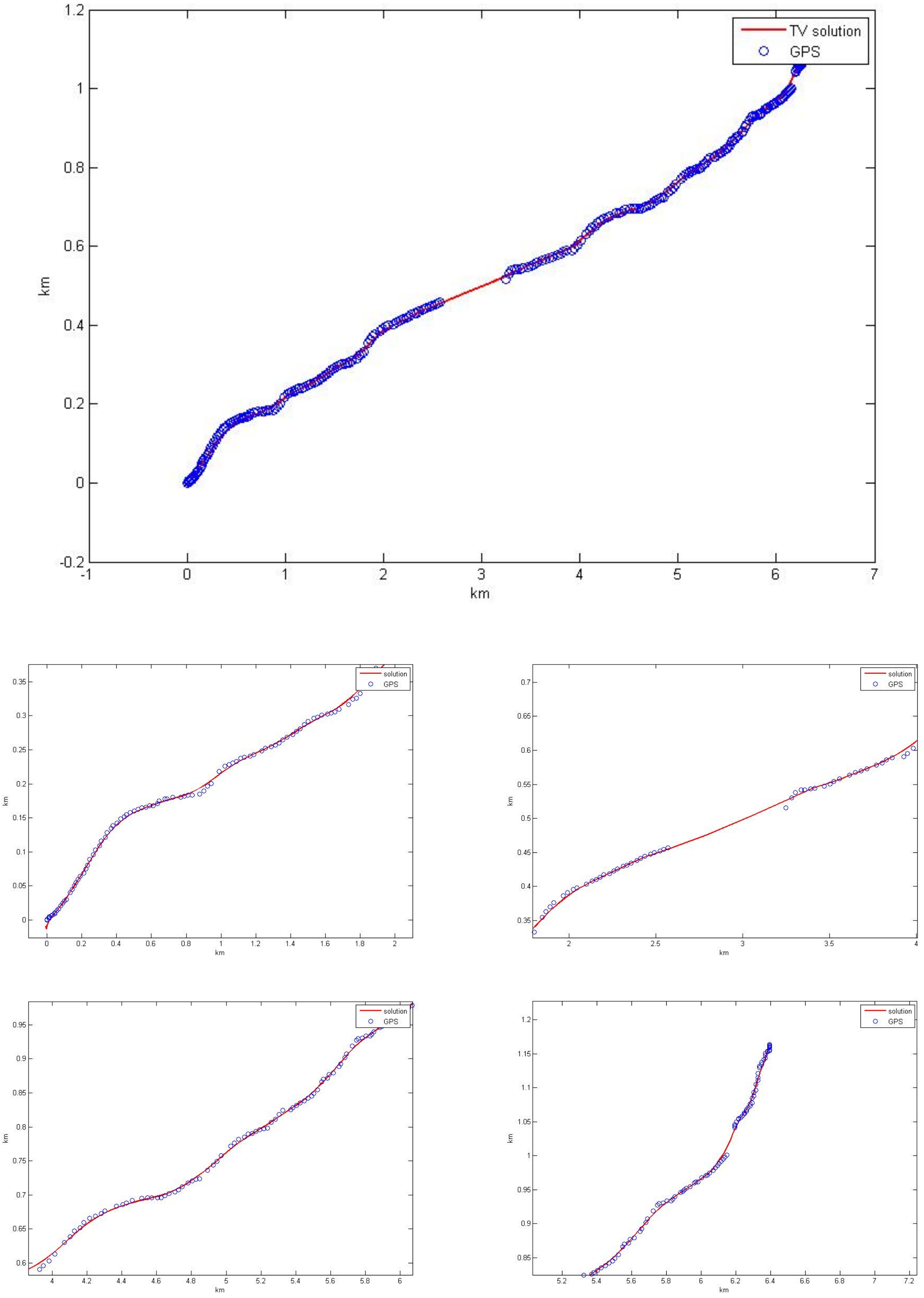


Figure 33: The solution to $\min_u \left\|u - \bar{u}_{GPS}\right\|_{L_2(w)} + \left\|\nabla u\right\|_{TV} + \left\|T_{loc}(u) - \bar{d}_{odo}\right\|_{L_2(w_{odo})}$, s.t. Curvature$(u) \leq \kappa_{\max}$

Bibliography

`www.cambridgeincolour.com/learn-photography-concepts.htm`.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 14–21.

Belmont, M. and Hotchkiss, A. (1997). Generalized cross-correlation functions for engineering applications, part i: basic theory. *Journal of Applied Mechanics*, 64:321–326.

Belmont, M., Hotchkiss, A., Maskell, S., and Morris, E. (1997). Generalized cross-correlation, part ii: discretization of generalised cross-correlation and progress to date in its implementation. *Journal of Applied Mechanics*, 64:327–335.

Belmont, M. and Jardon, S. (2000). Generalised cross-correlation functions for engineering applications. application to experimental data. *Experiments in Fluids*, 29:461–467.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923.

Lange, K., Little, R., and Taylor, J. (1989). Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84:881–896.

Muller, M. (2007). *Information Retrieval for Music and Motion*, chapter 4. Dynamic Time Warping, pages 69–84. Springer.

Safety, R. and Limited, S. B. (2011). *Railway Group Standard GC/RT5021 'Track System Requirements'*, fifth edition.

Schnemann, P. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Shenton, R. et al. (2008). Accuracy of a video odometry system for trains. In *Procedings of the 64th European Study Group with Industry.*

Thanawin, R., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. (2013). Addressing big data time series: mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data*, 7(3):10:1–10:31.

Uchida, S. and Sakoe, H. (2005). A survey of elastic matching techniques for handwritten character recognition. *IEICE Transactions on Information and Systems*, 88.8:1781–1790.