Determining a Track's Curvature from an On Board Camera

A.L. Gower[1]

[1]School of Mathematics, Statistics and Applied Mathematics, National University of Ireland Galway, Ireland.

May 4, 2014

Abstract

Keywords: Optimization, Camera Image.

1 Introduction

This current work will continue from, and use elements of, the report [ESGI 64(2008)] written as part of the ESGI 64 study group held in 2008. 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.

2 The Camera Image

The camera image is obtained by a projective transformation [ESGI 64(2008)]. For more informations and illustrations on how cameras work see the website [Cambridge in Colour]. The starting point for assembling this transformation is shown in Figure 1. 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,

$$\begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} = R_1(\psi)R_2(\theta)R_3(\phi) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{2.1}$$

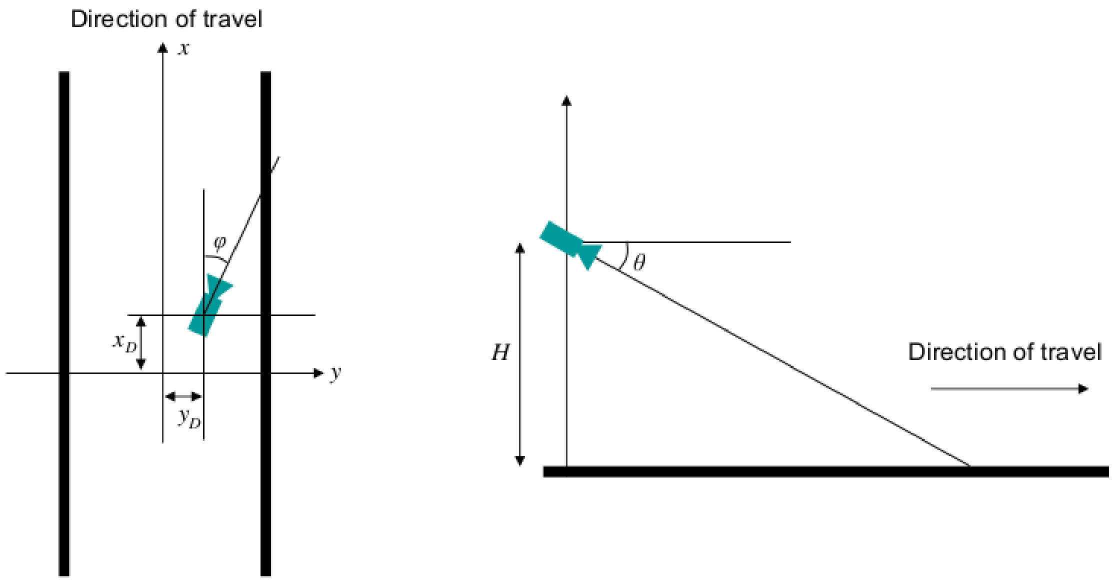

Figure 1: The Camera in its calibrated position. Note that if view from above the $y-$axis should be oriented in the opposite direction.

where

$$R_3(\phi) = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

$$R_2(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.3}$$

$$R_1(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \tag{2.4}$$

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

$$\begin{pmatrix} f \\ u \\ v \end{pmatrix} = \frac{f}{\xi} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}, \tag{2.5}$$

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

$$\begin{pmatrix} u \\ v \end{pmatrix} = F\left[\boldsymbol{\theta}, \boldsymbol{X}\right], \tag{2.6}$$

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

$$(H \cot\theta \cos\phi, H \cot\theta \sin\phi, -H)\,. \tag{2.7}$$

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

$$\begin{pmatrix} u \\ v \end{pmatrix} = F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X}\right], \quad \text{for } X \in [0, D], \tag{2.8}$$

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

$$Y = Y_L + \beta\frac{X^2}{2} \quad \text{and} \quad Z = -H + \alpha\frac{X^2}{2}, \tag{2.9}$$

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

$$\begin{pmatrix} u \\ v \end{pmatrix} = F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X} + (0, \beta, \alpha)X^2/2\right], \quad \text{for } X \in [0, D]. \tag{2.10}$$

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.

3 Minimization Method to Determine Track Shape

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.

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

$$\mathcal{R}^L[\delta\boldsymbol{\theta}, \delta\boldsymbol{X}, (\alpha, \beta)] = \left\{F\left[\boldsymbol{\theta} + \delta\boldsymbol{\theta}, (X, Y_L, -H) + \delta\boldsymbol{X} + (0, \beta, \alpha)X^2/2\right];\ X \in [0, D]\right\}, \tag{3.1}$$

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

$$\min_{\delta\boldsymbol{\theta},\delta\boldsymbol{X},\alpha,\beta} \|\mathcal{R}^L[\delta\boldsymbol{\theta},\delta\boldsymbol{X},(\alpha,\beta)] - \mathcal{R}^{L_M}\| + \|\mathcal{R}^R[\delta\boldsymbol{\theta},\delta\boldsymbol{X},(\alpha,\beta)] - \mathcal{R}^{R_M}\|, \tag{3.2}$$

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 $\pm$ 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

$$\boldsymbol{U}^M = \{\mathcal{R}^{L_M}, \mathcal{R}^{R_M}\} = \{(u_M^1, v_M^1), (u_M^2, v_M^2), \ldots, (u_M^{2N}, v_M^{2N})\}. \tag{3.3}$$

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 (3.4) 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

$$\boldsymbol{U} = \{\mathcal{R}^L, \mathcal{R}^R\} = \{(u^1, v^1), (u^2, v^2), \ldots, (u^{2N}, v^{2N})\}. \tag{3.4}$$

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

$$\|\boldsymbol{U} - \boldsymbol{U}^M\| = (\boldsymbol{U} - \boldsymbol{U}^M)^T(\boldsymbol{U} - \boldsymbol{U}^M) = \sum_{j=1}^{2N} (u^j - u_0^j)^2 + (v^j - v_0^j)^2 \tag{3.5}$$

3.1 Linearise and Project

The procedure for minimizing (3.2) for $\delta\boldsymbol{\theta}$, $\delta\boldsymbol{X}$, α and β would result in a system of non-linear 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

$$\boldsymbol{U} \approx \boldsymbol{U}^0 + \partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta}, \tag{3.6}$$

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 (3.5) to obtain

$$\|\boldsymbol{U} - \boldsymbol{U}^M\| \approx (\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta} + \boldsymbol{U}^0 - \boldsymbol{U}^M)^T(\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta} + \boldsymbol{U}^0 - \boldsymbol{U}^M). \tag{3.7}$$

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,

$$\begin{aligned}
\min_{\boldsymbol{\delta}} \|\boldsymbol{U} - \boldsymbol{U}^M\| &\implies \partial_{\boldsymbol{\delta}}\|\boldsymbol{U} - \boldsymbol{U}^M\| = \boldsymbol{0} \\
&\implies (\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0)^T(\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta} + \boldsymbol{U}^0 - \boldsymbol{U}^M) = \boldsymbol{0} \\
&\implies (\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0)^T\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0 \cdot \boldsymbol{\delta} = (\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0)^T(\boldsymbol{U}^M - \boldsymbol{U}^0) \\
&\implies \boldsymbol{\delta} = \left((\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0)^T\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0\right)^{-1}(\partial_{\boldsymbol{\delta}}\boldsymbol{U}^0)^T(\boldsymbol{U}^M - \boldsymbol{U}^0).
\end{aligned} \tag{3.8}$$

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 3.1 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.

An even worse scenario would occur, for example, if one chooses a measure of distance $\|\cdot\|$ between curves, discussed in Section 3, 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.

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.

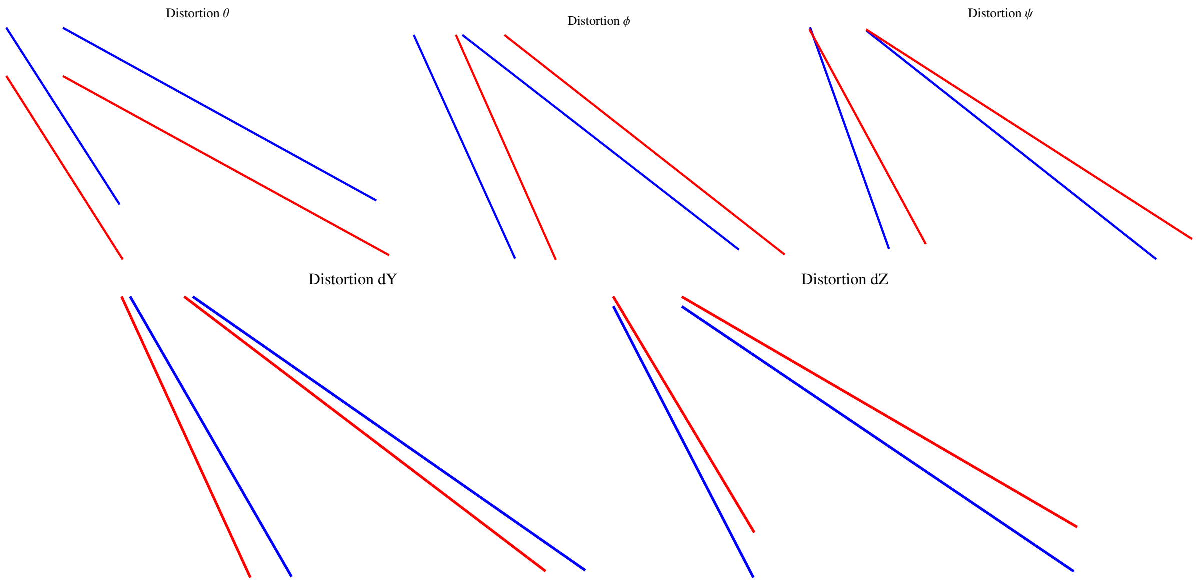

Figure 2: Distoriton to the rails in the camera image due to errors in the camera angles and position. The red rails are all perfectly straight rails with no camera errors. The blue rails are the linearised disturbances to the red rail.

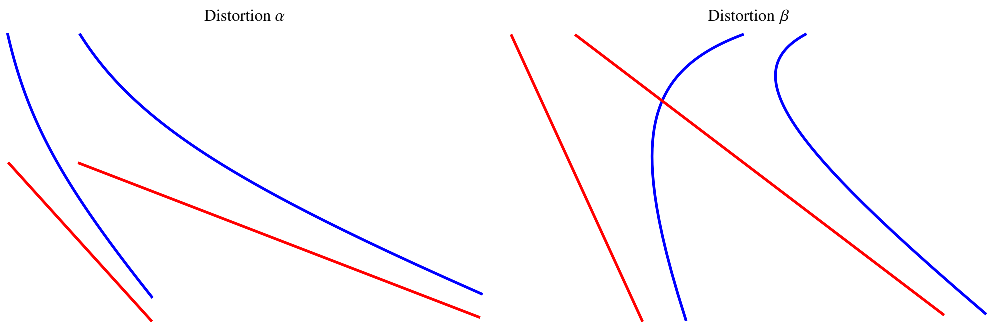

Figure 3: Distoriton to the rails in the camera image due to errors in the camera angles and position. The red rails are all perfectly straight rails with no camera errors. The blue rails are the linearised disturbances to the red rail.

3.2 Results

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

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

with a track width of 1.46 m. We assume that the camera is calibrated to point between the two tracks, so using equation (2.7) 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

$$|\delta\theta| = |\delta\phi| = |\delta\psi| = 3.6^\circ, \tag{3.9}$$

$$|dZ| = |dY| = 0.014, \tag{3.10}$$

$$\alpha = 0.01 \text{ and } \beta = 0.03. \tag{3.11}$$

This synthetic data was generated without linearising any of the equations. Errors were simply included in the model (2.10), and the analogous for the right rail, to generate the synthetic data.

The method developed in section 3.1 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 4. 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 5. We believe the errors are mainly introduced due to the nonlinearity of the camera image model (2.10) in the parameters θ, ϕ, ψ, Y_L and H, whereas the method linearises in terms of these parameters.

Acknowledgements

The support of the Hardiman Foundation (NUI Galway) and of the Irish Research Council is gratefully acknowledged. The authors of "Accuracy of a Video Odometry System for Trains" written as part of the ESGI 64 study group held in 2008 are acknowledged.

References

[Cambridge in Colour] The site illustrates many concepts about cameras:
www.cambridgeincolour.com/learn-photography-concepts.htm

[ESGI 64(2008)] Maurice Blount, Paul Dellar, Jens Gravesen, James MacLaurin, Sarah McBurnie, Andrew Stewart, David Szotten, Robert Whittaker, Dave Wood, *Accuracy of a Video Odometry System for Trains*. ESGI 64 (2008)

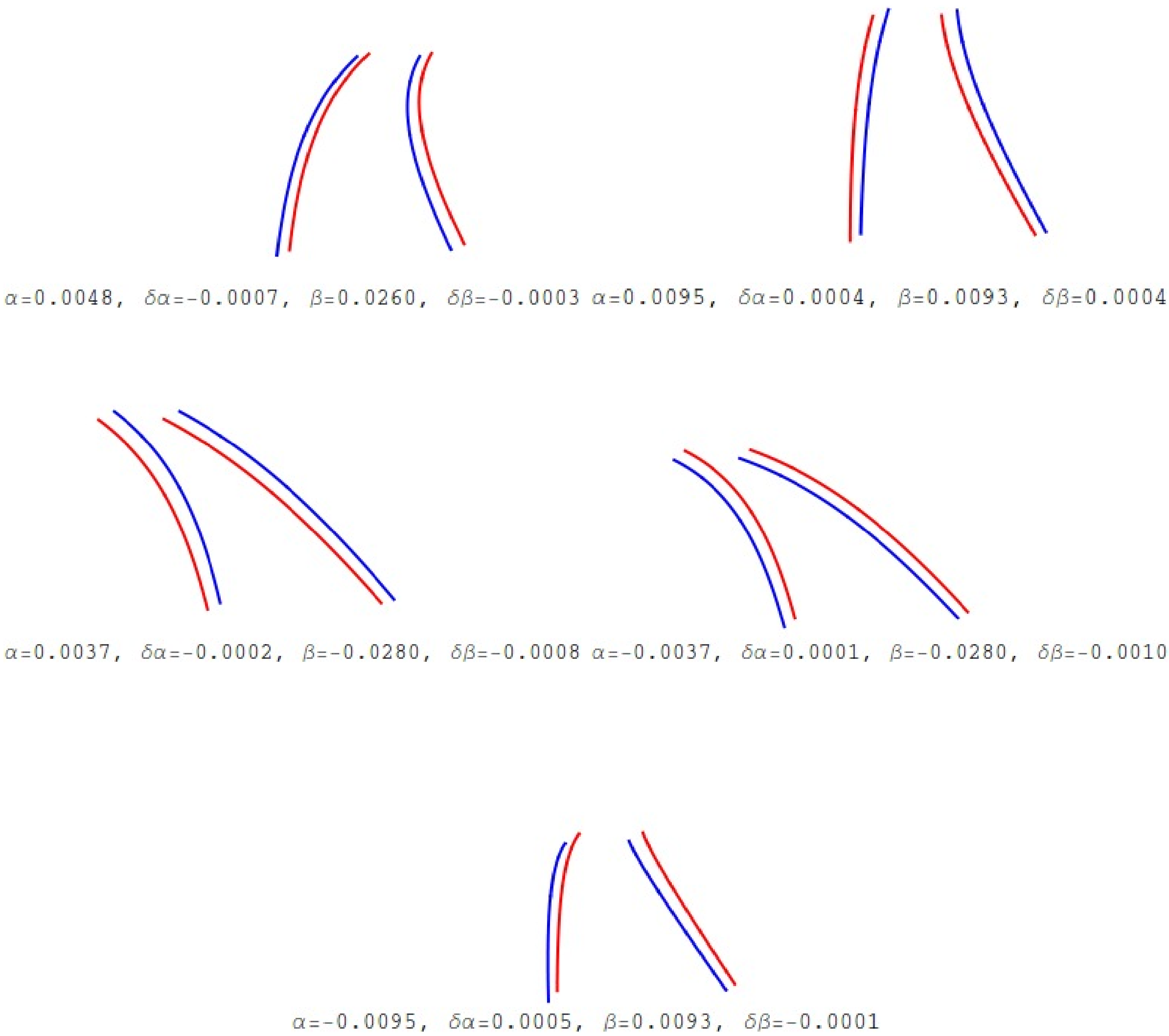

Figure 4: Distortion to the rails in the camera image due to errors in the camera angles and position. The red rails are all perfectly straight rails with no camera errors. The blue rails are synthetic data from the full nonlinear model.

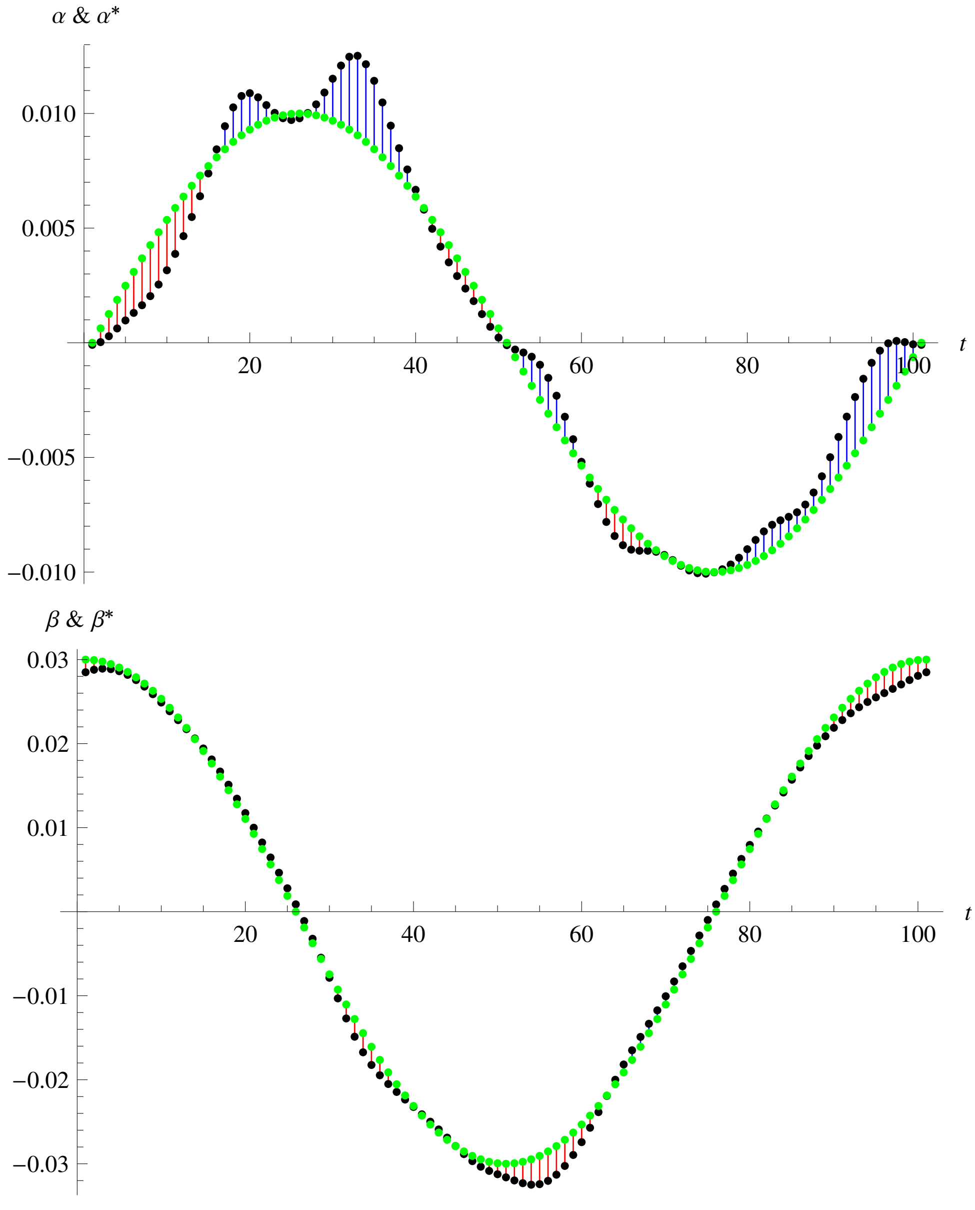

Figure 5: The green points are either the true value of α or β, while the the black points are either the estimated values α^* or β^*. The horizontal axis is the time steps of the video.