Shooting a Smooth Video with a Shaky Camera

Zoran Duric

Azriel Rosenfeld

Department of Computer Science
George Mason University
Fairfax, VA 22030-4444

Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

The image sequence in a video taken by a moving camera may suffer from irregular perturbations because of irregularities in the motion of the person or vehicle carrying the camera. We show how to use information in the image sequence to correct the effects of these irregularities so that the sequence is smoothed, i.e. is approximately the same as the sequence that would have been obtained if the motion of the camera had been smooth.

1   The Problem

This paper deals with the following problem: A camera mounted on a moving platform (carried by a ground vehicle or a person) collects a sequence of images. The platform is trying to move along a smooth trajectory. In fact, however, the motion of the platform varies from the desired smooth motion; for a ground vehicle, this would primarily be due to roughness of the terrain, and for the person, it would primarily be due to the walking motion itself. Our goal is to correct the image sequence so that it corresponds, as closely as possible, to the sequence that would have been collected if the motion had actually been smooth, or at worst piecewise smooth (the vehicle might hit an occasional pothole; the person might occasionally stumble).

There does not appear to have been much literature on this problem. In 1994 Burt et al. described methods of stabilizing an image sequence by registering the images to a reference mosaic [5, 13]; stabilization was also used for motion recovery by Irani, Rousso, and Peleg [14]. Note that in these papers the sequence was not merely smoothed, but completely stabilized. In 1996 the authors published a paper [9] which discussed the ground vehicle case in some detail, emphasizing real-time smoothing of the image sequence using normal flow. At about the same time, Yao and Chellappa discussed methods for selective stabilization of image sequences acquired by ground vehicles using a four-wheel vehicle model [26]. Giachetti, Campani, and Torre [12] showed how to smooth, in the vicinity of the image center, the optical flow obtained by a camera carried by a moving car. Srinivasan and Chellappa showed how to use fast structure-from-motion recovery from dense optical flow for 3D image stabilization [20]. Zomet, Peleg, and Arora showed how to produce rectified mosaics from image sequences obtained by a translating and rotating camera [28]. There seems to be no discussion in the computer vision literature about smoothing of image sequences obtained by a camera carried by a person.

In Section 2 we introduce a model for the motion of the platform (car or person) and discuss the sources of non-smoothness in its motion as well as the relative sizes of the smooth and non-smooth velocity components. In Section 3 we show that only the rotational components of the non-smooth motion have significant perturbing effects on the image sequence. We show how to identify image points at which rotational image flow is dominant, and how to use such points to estimate the platform's rotation. Finally, we describe two algorithms that fit piecewise smooth (in fact, piecewise constant) rotational motions to these estimates; the residual rotational motion can then be used to correct the images. We present results for an image sequence obtained from a camera carried by a ground vehicle moving across bumpy terrain; videos of several such sequences, before and after smoothing, demonstrate the effectiveness of our approach.

2   The Platform Motion Model

We have defined image sequence smoothing as eliminating (or at least greatly reducing) the effects of impulsive motion, i.e. processing the image sequence to make it look like a sequence that could have resulted from a smooth approximation to the motion. For a general motion model, the unsmoothness could involve the three parameters of the translational velocity of the platform's center of mass and the three parameters defining the rotational velocities of its principal axes, where each parameter is a function of time; thus in general, smoothing in a six-dimensional space might be required.
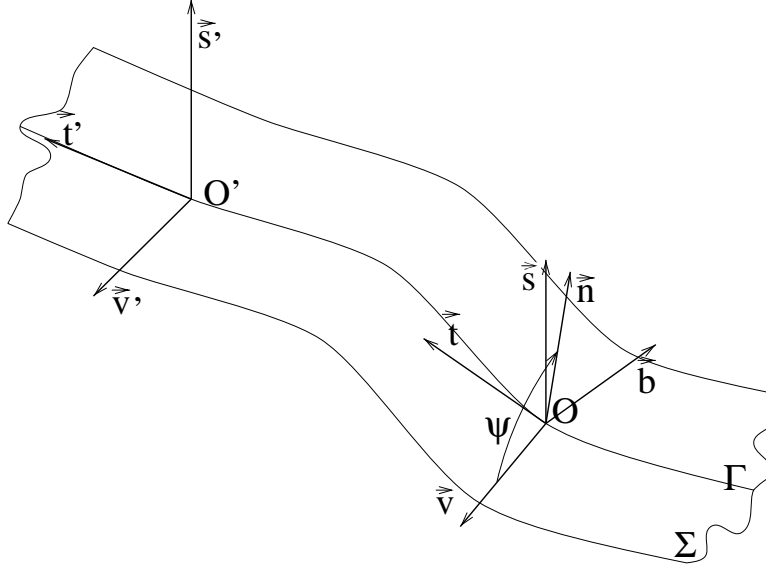
Figure 1: The Darboux frame moves along the path Γ which lies on the surface Σ.

The ideal motion of a platform (person or vehicle) moving on the ground does not have six degrees of freedom. It can be described as motion along a smooth trajectory Γ lying on a smooth surface Σ. The *Darboux frame* defined by Γ and Σ has axes defined by the tangent $\vec{\mathbf{t}}$ to Γ (and Σ), the second tangent $\vec{\mathbf{v}}$ to Σ (orthogonal to $\vec{\mathbf{t}}$), and the normal $\vec{\mathbf{s}}$ to Σ (see Figure 1). In Section 2.1 we will present a mathematical description of motion that satisfies the Darboux frame assumption, and we will also introduce the coordinate frames that will be used (in Section 3) to describe non-smooth platform motion.

General non-smooth platform motion can involve impulsive changes in any of the translational and/or rotational velocity functions. In the case of a platform moving on the ground, however, some of these changes can be expected to have small amplitudes and hence to be unimportant for smoothing. For example, as we shall see below (Section 2.2), impulsive translational changes and rotational changes around the vertical axis should be quite small. Thus the smoothing problem becomes simpler since the smoothing can be done in a space of dimensionality much lower than 6.

2.1   Smooth Motion on a Surface: The Darboux Frame

Consider a point O moving along a curve Γ which lies on a smooth surface Σ. There is a natural coordinate system $Otnb$ associated with Γ (even if it is a space curve), defined by the tangent $\vec{\mathbf{t}}$, normal $\vec{\mathbf{n}}$, and binormal $\vec{\mathbf{b}}$ of Γ. The triple $(\vec{\mathbf{t}}, \vec{\mathbf{n}}, \vec{\mathbf{b}})$ is called the *moving trihedron* or *Frenet-Serret coordinate frame.* We have the Frenet-Serret formulas [17]

$$\vec{\mathbf{t}}' = \kappa\vec{\mathbf{n}}, \quad \vec{\mathbf{n}}' = -\kappa\vec{\mathbf{t}} + \tau\vec{\mathbf{b}}, \quad \vec{\mathbf{b}}' = -\tau\vec{\mathbf{n}} \tag{1}$$

where κ is the curvature and τ the torsion of Γ.

When the curve Γ lies on a smooth surface Σ, it is more appropriate to use the *Darboux frame* $(\vec{\mathbf{t}}, \vec{\mathbf{v}}, \vec{\mathbf{s}})$ [16, 17]. We take the first unit vector of the frame to be the tangent $\vec{\mathbf{t}}$ of Γ and the surface normal $\vec{\mathbf{s}}$ to be the third frame vector; finally we obtain the second frame vector as $\vec{\mathbf{v}} = \vec{\mathbf{s}} \times \vec{\mathbf{t}}$ (see

2

Figure 1). Note that $\vec{\mathbf{t}}$ and $\vec{\mathbf{v}}$ lie in the tangent plane of Σ. Since the vector $\vec{\mathbf{t}}$ belongs to both the $Otnb$ and $Otvs$ frames, they differ only by a rotation around $\vec{\mathbf{t}}$, say through an angle $\psi \equiv \psi(s)$. We thus have

$$
\begin{pmatrix} \vec{\mathbf{v}} \\ \vec{\mathbf{s}} \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \vec{\mathbf{n}} \\ \vec{\mathbf{b}} \end{pmatrix}.
\tag{2}
$$

The derivatives of $\vec{\mathbf{t}}, \vec{\mathbf{v}}, \vec{\mathbf{s}}$ with respect to arc length along Γ can be found from (1) and (2):

$$
\vec{\mathbf{t}}' = \kappa_g \vec{\mathbf{v}} - \kappa_n \vec{\mathbf{s}}, \quad \vec{\mathbf{v}}' = -\kappa_g \vec{\mathbf{t}} + \tau_g \vec{\mathbf{s}}, \quad \vec{\mathbf{s}}' = \kappa_n \vec{\mathbf{t}} - \tau_g \vec{\mathbf{v}}
\tag{3}
$$

where

$$
\kappa_g \equiv \kappa \cos\psi, \quad \kappa_n \equiv \kappa \sin\psi, \quad \tau_g \equiv \tau + \frac{d\psi}{ds};
$$

κ_g is called the *geodesic curvature*, κ_n is called the *normal curvature*, and τ_g is called the *(geodesic) twist*.

It is well known that the instantaneous motion of a moving frame is determined by its rotational velocity $\vec{\omega}$ and the translational velocity $\vec{T}$ of the reference point of the frame. The translational velocity $\vec{T}$ of O is just $\vec{\mathbf{t}}$ and the rotational velocity of the $Otvs$ frame is given by the vector

$$
\vec{\omega}_d = \tau_g \vec{\mathbf{t}} + \kappa_n \vec{\mathbf{v}} + \kappa_g \vec{\mathbf{s}}.
$$

Hence the derivative of any vector in the $Otvs$ frame is given by the vector product of $\vec{\omega}_d$ and that vector. It can be seen that the rate of rotation around $\vec{\mathbf{t}}$ is just τ_g, the rate of rotation around $\vec{\mathbf{v}}$ is just κ_n, and the rate of rotation around $\vec{\mathbf{s}}$ is just κ_g.

If, instead of using the arc length s as a parameter, the time t is used, the rotational velocity $\vec{\omega}_d$ and translational velocity $\vec{T}$ are scaled by the speed $v(t) = ds/dt$ of O along Γ. This speed and the three components of the rotational velocity of the Darboux frame define a rigid motion model which we call *smooth surface motion*.

In Section 3 we will use two coordinate frames to describe platform motion. The "real" platform frame $C\xi\eta\zeta$ (which moves non-smoothly, in general) is defined by its origin C, which is at the center of mass of the platform, and its axes: $C\xi$ (fore/aft), $C\eta$ (crosswise), and $C\zeta$ (up/down); and the ideal platform frame $Otvs$ (the Darboux frame) corresponds to the smooth motion of the platform.

The motion of the platform can be decomposed into the motion of the $Otvs$ frame and the motion of the $C\xi\eta\zeta$ frame relative to the $Otvs$ frame. As we have just seen, the rotational velocity of the $Otvs$ (Darboux) frame is $v\vec{\omega}_d = v(\tau_g \vec{\mathbf{t}} + \kappa_n \vec{\mathbf{v}} + \kappa_g \vec{\mathbf{s}})$ and its translational velocity is $v\vec{\mathbf{t}}$. We denote the rotational velocity of the $C\xi\eta\zeta$ frame by $\vec{\omega}_v$ and its translational velocity by $\vec{T}_v$.

The position of the $C\xi\eta\zeta$ frame relative to the $Otvs$ frame is given by the displacement vector $\vec{d}_{v/d}$ between C and O, and the relative orientation of the frames is given by an orthogonal rotational matrix (matrix of direction cosines) which we denote by $R_{v/d}$. The translational velocity of the platform (the velocity of C) is the sum of three terms: (i) the translational velocity of the Darboux frame $v\vec{\mathbf{t}}$, (ii) the translational velocity $\vec{T}_{v/d} \equiv \dot{\vec{d}}_{v/d}$, and (iii) the displacement $v\vec{\omega}_d \times \vec{d}_{v/d}$ due to rotation of C in the $Otvs$ frame. The translational velocity of the platform expressed in the $Otvs$ frame is thus $v\vec{\omega}_d \times \vec{d}_{v/d} + v\vec{\mathbf{t}} + \dot{\vec{d}}_{v/d}$; its translational velocity in the $C\xi\eta\zeta$ frame is

$$
\vec{T}_v = R_{v/d}^T (v\vec{\omega}_d \times \vec{d}_{v/d} + v\vec{\mathbf{t}} + \dot{\vec{d}}_{v/d}).
\tag{4}
$$

3

Similarly, the rotational velocity of $C\xi\eta\zeta$ is the sum of two terms: (i) the rotational velocity $vR_{v/d}^T\vec{\omega}_d$ of the $Otvs$ frame, and (ii) the rotational velocity $\vec{\omega}_{v/d}$, which corresponds to the skew matrix $\Omega_{v/d} = R_{v/d}^T\dot{R}_{v/d}$. The rotational velocity of the $C\xi\eta\zeta$ frame expressed in the $Otvs$ frame is thus $v\vec{\omega}_d + R_{v/d}\vec{\omega}_{v/d}$; the corresponding expression in the $C\xi\eta\zeta$ frame is

$$\vec{\omega}_v = vR_{v/d}^T\vec{\omega}_d + \vec{\omega}_{v/d}. \tag{5}$$

The rotational axis $\vec{c}_{v/d}$ which corresponds to the rotational matrix $R_{v/d}$ is specified by its three direction cosines c_x, c_y, c_z. The rotation around this axis through an angle δ is then given by the matrix

$$R_{v/d} = \cos\delta\, I + (1 - \cos\delta) \begin{pmatrix} c_x^2 & c_x c_y & c_x c_z \\ c_y c_x & c_y^2 & c_y c_z \\ c_z c_x & c_z c_y & c_z^2 \end{pmatrix} + \sin\delta \begin{pmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{pmatrix} \tag{6}$$

where I is the identity matrix. When δ is small (in [8] it is shown that $\delta = \mathcal{O}(0.05)$rad) we have $\cos\delta \approx 1$, $\sin\delta \approx \delta$, and

$$R_{v/d}^T = I - \delta C_{v/d} + \mathcal{O}(\delta^2) \tag{7}$$

where the skew matrix $C_{v/d}$ is the matrix factor in the last term on the r.h.s. of (6). From (5) we thus have

$$\begin{aligned} \vec{\omega}_v &= vR_{v/d}^T\vec{\omega}_d + \vec{\omega}_{v/d} = v\vec{\omega}_d - \delta C_{v/d}\vec{\omega}_d + v\mathcal{O}(\delta^2)\vec{\omega}_d + \vec{\omega}_{v/d} \\ &= v\vec{\omega}_d + \vec{\omega}_{v/d} - \delta v\vec{c}_{v/d} \times \vec{\omega}_d + \mathcal{O}(v\delta^2\|\vec{\omega}_d\|). \end{aligned} \tag{8}$$

The significant terms on the r.h.s. of (8) are $v\vec{\omega}_d$, which is the smooth velocity of the platform; $\vec{\omega}_{v/d}$, which is the non-smooth velocity of the platform; and the cross-product term $\delta v\vec{c}_{v/d} \times \vec{\omega}_d$, which is also non-smooth. Since δ is small and $\|\vec{c}_{v/d}\| = 1$, the cross product term is small compared to $v\vec{\omega}_d$; indeed, it is $\mathcal{O}(v\delta\|\vec{\omega}_d\|)$. Hence

$$\vec{\omega}_v = v\vec{\omega}_d + \vec{\omega}_{v/d} + \mathcal{O}(v\delta\|\vec{\omega}_d\|). \tag{9}$$

Rotations around the fore/aft, sideways, and up/down axes of a platform are called *roll*, *pitch*, and *yaw*, respectively. In terms of our choice of the platform coordinate system, these are rotations around the ξ, η, and ζ axes.

2.2 Departures of Platform Motion from Smoothness

The motion of a platform (vehicle or person) depends on many factors: the type of intended motion, its speed, and the nature of the surface on which the platform is moving. These factors tend to remain constant; they undergo abrupt changes only occasionally, e.g. the person stumbles or the vehicle hits an obstacle. Such events may produce impulsive changes in the motion, but these changes have only short-term effects. In addition to these occasional events, there is "steady-state" non-smoothness of the motion because of the irregularities in the walking motion or the roughness of the surface.

A walking or running person moves forward by not more than a few meters at every step. The sides of the person's body alternate in moving upward and downward by at most a few centimeters at each step; this gives rise to an impulsive rolling motion. Since the person's torso does not remain equally upright at all times, there will also be some impulsive pitching motion. On the other hand, the body's translational motion along the up/down and sideways axes will be negligible, and so will its rotational motion around the yaw axis. To a good approximation, we can assume that the body's forward translational motion is its smooth motion, and its rolling and pitching motions are its undesired impulsive motion. For example, the forward velocity might be 1m/sec (not a fast walk), and the rolling and pitching motions might correspond to angular velocities of on the order of 0.1 rad/sec. This discussion is consistent with the analysis in [27], Ch. 3, and with the arguments used in [6], Chs. 10-13.

The analysis of the motion of a ground vehicle moving on a rough surface (e.g., on off-road terrain) is analogous, because the bumpiness of the terrain also produces both rolling and pitching motions. It is still the case that the translational motions along the up/down and sideways axes are negligible (any bouncing of the vehicle has a small amplitude), and its rotational yaw motion is also negligible. A more detailed discussion of ground vehicle motion is given in the Appendix. Similar analyses can be given for other types of vehicles; for example, a very interesting analysis of the stability of the motion of a unicycle can be found in [19].

3 Image Motion: Smoothing of Image Sequences

In this section we analyze the properties of images and image motion fields obtained by a camera mounted on a platform moving across the ground. Based on our analysis, we design a smoothing algorithm which we use to smooth several image sequences obtained by a camera mounted on a real ground vehicle moving on rough terrain.

In Section 3.1 we derive the relative motion equations for the camera and show that in realistic situations we only need to smooth the rotational components of the motion. In Section 3.2 we review properties of the image motion fields in sequences of images obtained by the camera; in particular, we discuss conditions under which the rotational component of the field is dominant at a given point of the image, based on the distance of the corresponding scene point. In Section 3.3 we discuss methods of classifying image points with respect to their distance. In Section 3.4 we give an algorithm for estimating the rotational motion of the camera based on such a classification of image points, and give examples of such estimates for several real image sequences; these examples confirm our analysis in Section 2. Finally, in Section 3.5 we describe two algorithms for the smoothing of image sequences, one of which uses only the preceding and current images (so that it can operate in real time), and we present the results of applying these algorithms to a real image sequence.

3.1 Camera Motion

Let $\vec{d_c}$ be the position vector of the nodal point of the camera relative to the mass center of the platform. The orientation of the camera relative to the platform coordinate system $C\xi\eta\zeta$ is given by an orthogonal rotational matrix (a matrix of direction cosines) which we denote by R_c. The columns of R_c are the unit vectors of the camera coordinate system expressed in the platform coordinate system. We will assume that the position and orientation of the camera relative to the

platform coordinate system do not change as the platform moves. (The cases of a camera that is non-rigidly mounted or that can move relative to the platform — for example, a hand-held camera — are left for future work.) Thus we will assume that R_c and $\vec{d}_c$ are constant and known.

Given the position $\vec{r}_e$ of a static scene point E in the camera coordinate system, its position $\vec{p}_e$ in the platform coordinate system $C\xi\eta\zeta$ is given by

$$\vec{p}_e = R_c \vec{r}_e + \vec{d}_c$$

Since R_c and $\vec{d}_c$ are constant we have $\dot{\vec{p}}_e = R_c \dot{\vec{r}}_e$, so that $\dot{\vec{r}}_e = R_c^T \dot{\vec{p}}_e$. The velocity of E is given by

$$\dot{\vec{r}}_e = -\vec{\omega} \times \vec{r}_e - \vec{T}. \tag{10}$$

In this expression, the rotational velocity is $\vec{\omega} = R_c^T \vec{\omega}_v$, and the translational velocity is $\vec{T} = R_c^T(\vec{T}_v + \vec{\omega}_v \times \vec{d}_c)$, where $\vec{\omega}_v$ and $\vec{T}_v$ are the rotational and translational velocities of the platform coordinate system. At the end of Section 2.1 we saw (4,9) that these velocities can be expressed in terms of the (smooth) rotational and translational velocities $v\vec{\omega}_d$ and $v\vec{t}$ of the Darboux frame and the (impulsive) rotational and translational velocities $\vec{\omega}_{v/d}$ and $\dot{\vec{d}}_{v/d}$ of the platform frame relative to the Darboux frame. Using these expressions, we have

$$\vec{\omega} = R_c^T(v\vec{\omega}_d + \vec{\omega}_{v/d} + \mathcal{O}(v\delta\|\vec{\omega}_d\|)), \tag{11}$$

$$\vec{T} = R_c^T R_{v/d}^T(v\vec{\omega}_d \times \vec{d}_{v/d} + v\vec{t} + \dot{\vec{d}}_{v/d}) + \vec{\omega} \times (R_c^T \vec{d}_c) \tag{12}$$

We saw in Section 2.2 and in the Appendix that both $\|v\vec{\omega}_d\|$ and $\|\vec{\omega}_{v/d}\|$ are $\mathcal{O}(0.1)$rad/sec and that the third term on the r.h.s. of (9) is small compared to the first two terms. The factors R_c^T and $R_{v/d}^T$ do not affect the magnitude of either $\vec{\omega}$ or $\vec{T}$. Thus the first two terms on the r.h.s. of (11) have comparable magnitudes, so that it is clearly important to smooth the effects of the rotational components on the image sequence.

As regards the translational components, for normal speeds of a vehicle we have (see the Appendix) $\|\vec{d}_{v/d}\| = \mathcal{O}(0.025)$m/sec, $\|\dot{\vec{d}}_{v/d}\| = \mathcal{O}(0.1)$m/sec, and $\|\vec{d}_c\| = \mathcal{O}(1)$m/sec. The magnitudes of the terms on the r.h.s of (12) are thus $\|v\vec{\omega}_d \times \vec{d}_{v/d}\| \leq v\|\vec{\omega}_d\| \|\vec{d}_{v/d}\| = \mathcal{O}(0.0025)$m/sec; $\|v\vec{t}\| = v = \mathcal{O}(10)$m/sec; and $\|\vec{\omega} \times (R_c \vec{d}_c)\| \leq \|\vec{\omega}\| |\vec{d}_c\| = \mathcal{O}(0.1)$m/sec. Therefore, the dominant term in the expression for $\vec{T}$ is $v\vec{t}$ since it is two orders of magnitude larger then any of the other three terms of $\vec{T}$. A similar conclusion evidently applies as regards the translational components of a walking person's motion. Smoothing the effects of the translational components on the image sequence is thus unimportant, since $v\vec{t}$ is already smooth. Furthermore, after estimating $\vec{\omega}_{v/d}$ and $R_{v/d}$ we will use the matrices $R_c^T R_{v/d} R_c$ to smooth the images. This, in turn, will smooth the effects of rotations on the translational velocity $\vec{T}$.

3.2 The Image Motion Field

We first briefly describe the image motion field when the camera is moving in a static environment.

Let (X, Y, Z) denote the Cartesian coordinates of a scene point with respect to the camera frame (see Figure 2), and let (x, y, f) denote the corresponding coordinates in the image plane. f
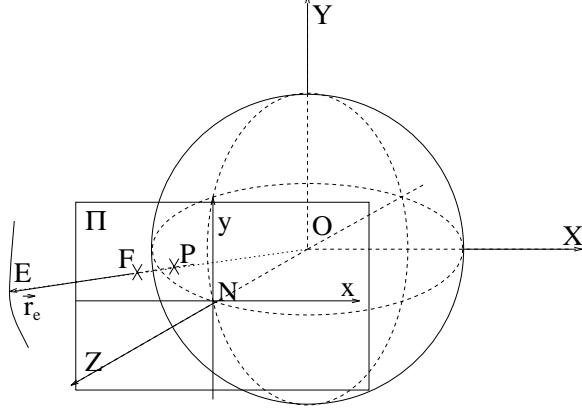
Figure 2: The plane perspective projection image of (X, Y, Z) is $(fX/Z, fY/Z, f)$.

is the focal length of the camera. The projection equations for the plane perspective projection are given by

$$x = \frac{X}{Z}f, \quad y = \frac{Y}{Z}f. \tag{13}$$

The instantaneous velocity of the image point (x, y, f) resulting from the velocity of the corresponding scene point relative to the camera frame can be obtained by taking derivatives of (13) with respect to time and using (10):

$$\dot{x} = \frac{-t_x f + x t_z}{Z} + \omega_x \frac{xy}{f} - \omega_y \left(\frac{x^2}{f} + f \right) + \omega_z y, \tag{14}$$

$$\dot{y} = \frac{-t_y f + y t_z}{Z} + \omega_x \left(\frac{y^2}{f} + f \right) - \omega_y \frac{xy}{f} - \omega_z x. \tag{15}$$

Let $\vec{i}$, $\vec{j}$, and $\vec{k}$ be the unit vectors in the x, y, and z directions, respectively; $\dot{\vec{r}} = \dot{x}\vec{i} + \dot{y}\vec{j}$ is the projected motion field at the point $\vec{r} = x\vec{i} + y\vec{j} + f\vec{k}$. For any image point $\vec{r}$ we can split the image motion field $\dot{\vec{r}}$ into the translational image motion field $\dot{\vec{r}}_t$ and the rotational image motion field $\dot{\vec{r}}_\omega$.

We say that translation (rotation) is *dominant* at an image point $\vec{r}$ if $\|\dot{\vec{r}}_t\| > (<) \|\dot{\vec{r}}_\omega\|$. Similarly, we say that translation (rotation) is *strongly dominant* at $\vec{r}$ if $\|\dot{\vec{r}}_t\|$ is an order of magnitude larger (smaller) than $\|\dot{\vec{r}}_\omega\|$.

In [8] we derive bounds on the ratio of the rotational and the translational image velocities at a given image point $\vec{r}$:

$$Z\frac{\|\vec{\omega}\|}{\|\vec{T}\|} \cdot \frac{|\sin \angle(\vec{\omega}, \vec{r})|}{|\sin \angle(\vec{T}, \vec{r})|} \leq \frac{\|\dot{\vec{r}}_\omega\|}{\|\dot{\vec{r}}_t\|} \leq \frac{\|\vec{r}\|^2}{f^2} \cdot Z\frac{\|\vec{\omega}\|}{\|\vec{T}\|} \cdot \frac{|\sin \angle(\vec{\omega}, \vec{r})|}{|\sin \angle(\vec{T}, \vec{r})|}. \tag{16}$$

In order for rotation to be (strongly) dominant at $\vec{r}$, we must have $\|\dot{\vec{r}}_\omega\|/\|\dot{\vec{r}}_t\| > 1$ (10). Consider the three factors on the l.h.s. of (16). The first factor depends on Z; the more distant the scene point the greater its value. As regards the second factor, if the camera orientation R_c is known the angle $\angle(\vec{T}, \vec{r})$ is known too. [We have seen in Section 3.1 that the translational velocity of the platform (and camera) is approximately parallel to the platform's fore/aft axis.] However, the angle $\angle(\vec{\omega}, \vec{r})$ is not readily available. [We have seen in Section 3.1 that the rotational velocity of the platform

7

consists of two parts: (i) smooth rotational velocity of the Darboux frame, which changes slowly over time, and (ii) non-smooth (impulsive) rotational velocity of the platform frame relative to the Darboux frame. We have also seen that the impulsive components of the roll and pitch rotational velocities are of comparable or higher magnitude than the respective smooth components, whereas the impulsive component of the yaw rotational velocity is small. This means that unless the platform is turning, $\vec{\omega}$ lies approximately in the plane spanned by the platform's fore/aft and crosswise axes; but it could be pointing in any direction in that plane.]

Except at the point $\vec{r}$ at which $\angle(\vec{\omega}, \vec{r}) = 0°$, dominance of rotation can be guaranteed provided that

$$Z \geq Z_r = \frac{\|\vec{T}\|}{\|\vec{\omega}\|} \cdot \frac{|\sin \angle(\vec{T}, \vec{r})|}{|\sin \angle(\vec{\omega}, \vec{r})|}. \tag{17}$$

(or $Z \geq 10 Z_r$ for strong dominance). Typically, for a vehicle we have $\|\vec{T}\| = \mathcal{O}(10)$m/sec and $\|\vec{\omega}\| = \mathcal{O}(0.1)$rad/sec; hence if we ignore the sines we have $Z_r \approx 100$m (1km for strong dominance). Similarly, for a person we have $\|\vec{T}\| = \mathcal{O}(1)$m/sec, so that $Z_r \approx 10$m (100m for strong domin!la ance). If the rotational motion of the platform is pure roll, $\vec{T}$ and $\vec{\omega}$ are parallel and the angles are equal, so that the ratio of their sines is 1 and the sines can indeed be ignored. In all other cases, the ratio of sines depends on $\vec{r}$. If the ratio of sines in (17) is smaller than 1, rotation will be (strongly) dominant even for smaller Z's; if it is greater than 1, larger Z's will be needed.

If $\angle(\vec{\omega}, \vec{r})$ is bounded away from zero, say $> \alpha$, the ratio of sines in (17) is smaller than $1/\sin \alpha$, and rotation will be (strongly) dominant provided $Z \geq \|\vec{T}\|(\|\vec{\omega}\| \sin \alpha)^{-1}$ ($\times 10$). If the field of view of the camera is greater than 2α, $\angle(\vec{\omega}, \vec{r}) > \alpha$ is guaranteed to be true in at least part of the image. For example, if in the vehicle example in the previous paragraph $\alpha = 10°$ ($\sin \alpha \approx 0.17$), then in most of the image, $Z \geq 600$m will be sufficient for dominance of rotation. In fact, for typical camera orientations and typical terrain (see Section 3.3), the horizon will be distant and visible in the image, and $\angle(\vec{\omega}, \vec{r}) > \alpha$ is guaranteed to be true along most of the horizon, so that rotation will be dominant along most of the horizon.

3.3 Identification of Distant Image Points

Typical images obtained by a camera mounted on a platform (person or vehicle) moving along the ground are not arbitrary. In general, the camera can be pointing in any direction, but its orientation relative to that of the platform (the matrix R_c) will be known. It is reasonable to assume that the camera axis is approximately parallel to the plane spanned by the fore/aft and crosswise axes of the platform. Thus if the platform is on level ground, the camera is pointing approximately horizontally. Thus, unless the platform is at the bottom of a deep hollow, or is screened by natural or artificial objects, the camera can see objects that are very close to the platform as well as objects that are very distant, as shown in Figures 3 and 4. When only nearby objects can be seen, and $\angle(\vec{T}, \vec{r})$ is significantly different from zero, translation dominates and thus (see Section 3.1) the image sequence does not need smoothing. In general, however, we expect that this will not be the case, and distant parts of the scene will be visible; for these parts, as indicated in Section 3.2, rotation will usually dominate. If distant points can be identified, their motions in the image can be used to estimate rotation. In the remainder of this section we will discuss methods of classifying image points based on their distances from the camera.

If range information is available, image points can be classified based on their ranges. A stereo
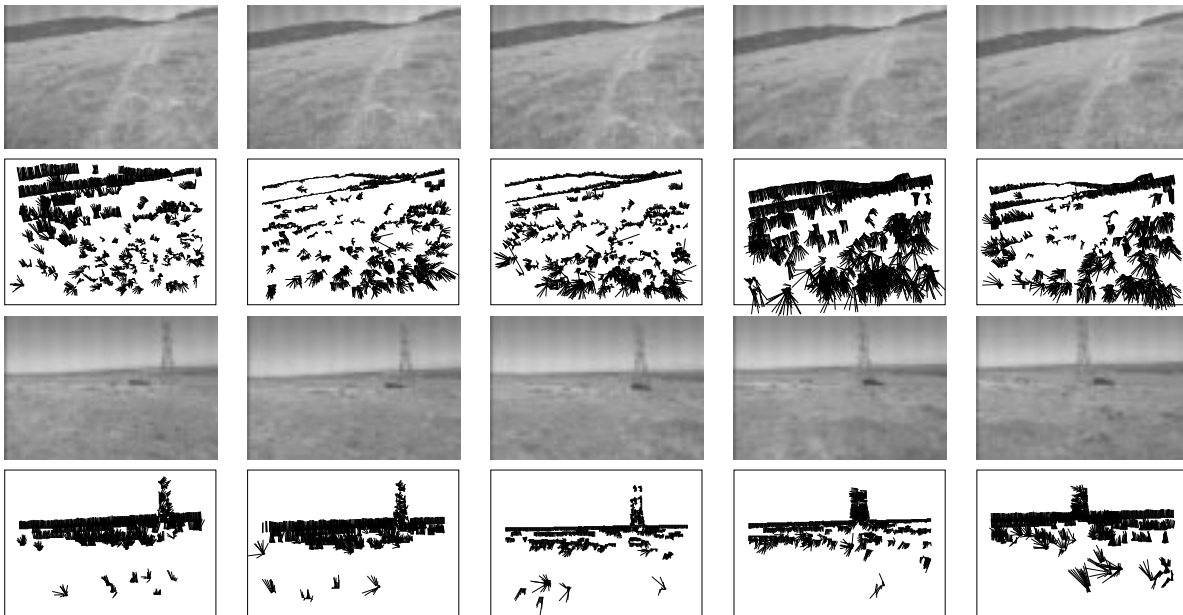
Figure 3: Frames 20, 60, 100, 140, and 180 and corresponding normal flow results from two image sequences of 200 frames each.

rig can be used for range estimation; to do the estimation in real time, a fast multiple-camera dynamic programming based stereo algorithm can be used [15]. The task here is much simpler than in general stereo since it is necessary to identify only distant points. This can be easily done since such points have near-zero disparities if the cameras are parallel. When only a single camera is available, motion stereo might be used to identify distant points. However, motion stereo works best when the motion is mostly translational, in which case the image sequence should not need to be smoothed.

When the focus of expansion (FOE) is inside the image (which is certainly true if the camera axis is approximately parallel to the fore/aft platform axis), the rates of approach (ROA — the inverse of time to contact) to objects in the scene can be estimated using line integrals of normal flow along closed image contours [2, 10]. The ROA has large (small) values for image patches which correspond to close (distant) objects, and can be used to classify the patches regardless of rotation. Unfortunately, ROA estimation is computationally expensive, and when the FOE is not inside the image (as in the case of a sideways-pointing camera), it is not practical to compute the ROA.

Horizon detection provides a potentially more useful method of identifying distant points. Typically, the horizon is the most distant part of the scene (except for the sky), and unless the terrain is very steep (or the platform is moving alongside a wall), it is usually visible in the image and usually quite distant. At points where the horizon is very close (e.g., there are tall objects directly in front of the camera) translation is dominant and the image sequence should be smooth. We therefore need only be concerned with points such that the horizon is at least at a medium distance and rotation is significant. In general, the horizon bounds the brightest part of the image, which is almost always the sky, and the gradient along the horizon is usually high in magnitude. Thus the horizon can be detected by finding the bright part(s) of the image (at its top, since the camera
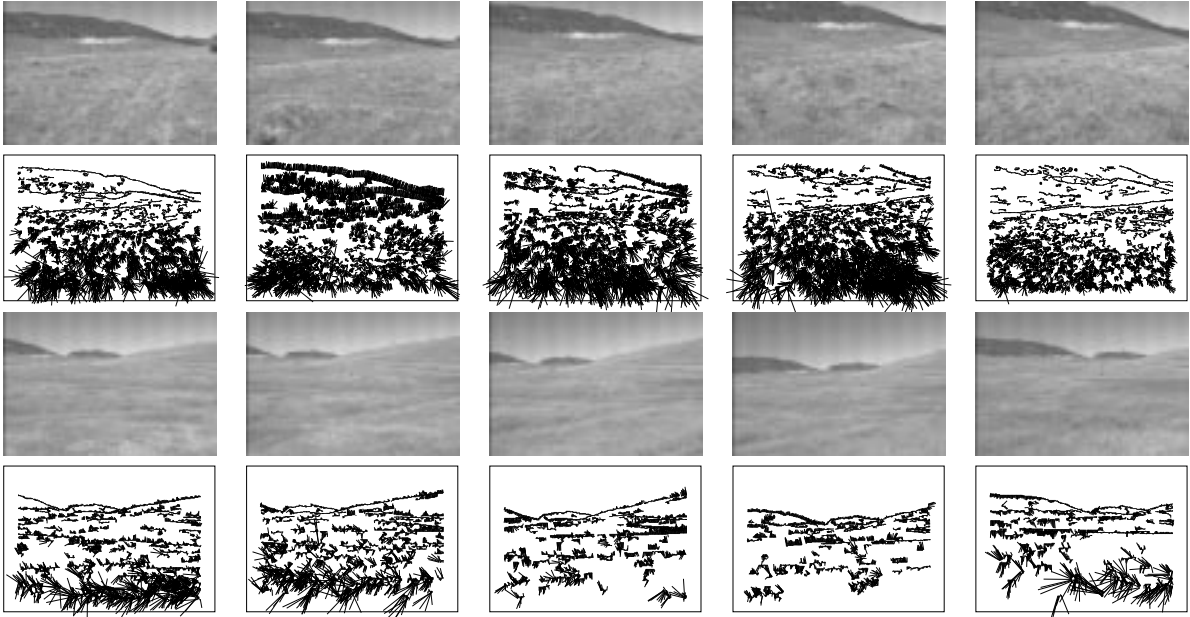
Figure 4: Frames 0, 200, 400, 600, and 800 from two image sequences of 900 frames each.

axis is approximately horizontal), and estimating the boundaries of these parts. The orientations, positions, and strengths of the edges along the horizon should change slowly with time, and the flow along the horizon should be smooth; these observations can be used to track the horizon from image to image.

Horizon detection need not be repeated in every frame; it should be easy to track distant horizon points from frame to frame. More generally, if the set of distant points is known at time t_0, the task of identifying distant points in the next frame (at time $t_0 + \Delta t$) is greatly simplified. Since Δt is small, the distant points remain distant. [When the speed $v < 10$m/sec and frame time $\Delta t = \mathcal{O}(1/30)$sec the change in distance is $\mathcal{O}(0.3)$m for a forward-looking camera and less for a side-looking camera.] Thus, the image motions of the distant points in Δt (between the frames) are mostly due to rotation and are thus small. [When $\|\vec{\omega}\| = \mathcal{O}(0.1)$rad/sec and $\Delta t = \mathcal{O}(1/30)$sec we have from (15) that an upper bound on the rotational image velocity is $\mathcal{O}(f/300)$ pixels/frame, where f is measured in pixels. Note that $f/300 < 1$ (the motion is subpixel) unless the angular field of view of the camera is relatively small, corresponding to a long focal length, in which case the resolution of the image can be reduced.] Also, the magnitudes and orientations of the image gradients at those points do not change much. Thus the images of distant points at time t_0 and time $t_0 + \Delta t$ are similar in position and appearance and can be easily tracked from frame to frame.

3.4 Estimation of Rotation

We now describe our algorithm for using distant points to estimate rotation. We shall use the following notation: Let I be the image intensity at $\vec{r}$, let $\vec{n}_r = n_x \vec{\imath} + n_y \vec{\jmath} = \nabla I / \|\nabla I\|$ be the

direction of the image intensity gradient at $\vec{r}$, and let

$$\vec{a}_1 = \begin{pmatrix} f \\ 0 \\ -x \end{pmatrix}, \quad \vec{a}_2 = \begin{pmatrix} 0 \\ f \\ -y \end{pmatrix}, \quad \vec{b}_1 = \begin{pmatrix} -xy/f \\ x^2/f + f \\ -y \end{pmatrix}, \quad \vec{b}_2 = \begin{pmatrix} -y^2/f - f \\ xy/f \\ x \end{pmatrix}.$$

The *normal motion field* at $\vec{r}$ is the projection of the image motion field onto the gradient direction $\vec{n}_r$ and is given by $\dot{\vec{r}}_n = (\dot{\vec{r}} \cdot \vec{n}_r)\vec{n}_r$. From (14-15) we have

$$\dot{\vec{r}}_n \cdot \vec{n}_r = n_x \dot{x} + n_y \dot{y} = -(n_x \vec{a}_1 + n_y \vec{a}_2) \cdot \frac{\vec{T}}{Z} - (n_x \vec{b}_1 + n_y \vec{b}_2) \cdot \vec{\omega}. \tag{18}$$

The terms on the r.h.s. of (18) are the translational normal motion and the rotational normal motion at $\vec{r}$, respectively.

Consider a set of image points on the horizon (or more generally, a set of image points which seem to be at large distances); these points are in fact usually distant and rotation is (strongly) dominant. For such points, it will usually be the case that the rotational normal motion (strongly) dominates the translational normal motion. Indeed, in order for the translational normal motion to be dominant it must be the case that $|\dot{\vec{r}}_t \cdot \vec{n}_r| > |\dot{\vec{r}}_\omega \cdot \vec{n}_r|$, which can happen only when $\dot{\vec{r}}_t$ and $\dot{\vec{r}}_\omega$ are almost orthogonal and $\vec{n}_r$ and $\dot{\vec{r}}_t$ are in almost the same direction; we will see later in this section that this can occur consistently only in certain very artificial scenes. We can thus write

$$\dot{\vec{r}}_n \cdot \vec{n}_r = \epsilon_t - (n_x \vec{b}_1 + n_y \vec{b}_2) \cdot \vec{\omega} \tag{19}$$

where ϵ_t represents the translational normal motion and is usually small relative to the rotational normal motion.

The *normal flow* at $\vec{r}$ is defined as $-I_t/\|\nabla I\|$. From [22] we know that the magnitude of the difference between the normal flow field and the normal motion field is inversely proportional to the gradient magnitude; we can thus write

$$\dot{\vec{r}}_n \cdot \vec{n}_r = -\frac{I_t}{\|\nabla I\|} + \mathcal{O}(\|\nabla I\|^{-1}). \tag{20}$$

From (19) and (20) we can write

$$(n_x \vec{b}_1 + n_y \vec{b}_2) \cdot \vec{\omega} - \frac{I_t}{\|\nabla I\|} = \epsilon \tag{21}$$

where $\epsilon = \epsilon_t - \mathcal{O}(\|\nabla I\|^{-1})$, which is small if the gradient magnitude is large.

For each horizon point $\vec{r}_i$ we have one equation (21). Let the number of horizon points be $N \geq 3$. We then have a system

$$A\vec{\omega} - \mathbf{y} = E$$

where $\mathbf{y}$ is an N-element array with elements $I_t(\vec{r}_i)/\|\nabla I(\vec{r}_i)\|$, A is an $N \times 3$ matrix with rows $n_x \vec{b}_1 + n_y \vec{b}_2$, and E is an N-element array with elements ϵ_i. We seek $\vec{\omega}$ that minimizes $\|E\| = \|\mathbf{y} - A\vec{\omega}\|$; the solution satisfies the system [21]

$$A^T A \vec{\omega} = A^T \mathbf{y} = \mathbf{d}. \tag{22}$$

We solve this system using the Cholesky decomposition [21]. (Since the matrix $A^T A$ is a positive definite 3×3 matrix there exists a lower triangular matrix L such that $LL^T = A^T A$. We solve two triangular systems $L\mathbf{e} = \mathbf{d}$ and $L^T \vec{\omega} = \mathbf{e}$ for $\vec{\omega} = (\omega_x \ \omega_y \ \omega_z)^T$ in the camera coordinate system.)

The computed $\vec{\omega}$ may be inaccurate due to various geometrical and numerical factors, to be discussed below. However, it is possible to iteratively improve on the computed solution of the system (22). Given the estimate $\vec{\omega}$ for a given frame, based on the flow between the frame and its predecessor(s) we create the skew rotational velocity matrix Ω that corresponds to $\vec{\omega}$, and using (7) we define the rotational matrix $R = I + \Delta t \Omega$ (where Δt is the time interval between frames); here we have used the fact that the angle of rotation around the axis $\vec{\omega}$ in time Δt is $\Delta t \vec{\omega}$. We then apply the rotation R to the frame (derotation/warping); as the result the rotational velocity of the image sequence at that frame will be reduced by $\vec{\omega}$. After derotating the frame we compute a residual rotational velocity $\Delta \vec{\omega}$ from the derotated frame and the (uncorrected) predecessor frame. We then replace $\vec{\omega}$ by $\vec{\omega} + \Delta \vec{\omega}$ and proceed to again create Ω and R and use R to again derotate the original frame. We repeat this process until $\|\Delta \vec{\omega}\| < \varepsilon$. At each step we do the derotation using the new $\vec{\omega} + \Delta \vec{\omega}$, and we apply it to the original image (rather than using the new $\Delta \vec{\omega}$ on the already derotated image), to avoid accumulation of errors. This method converges rapidly when the image motion is small, which is usually the case. (Typically, the magnitude of the error in $\vec{\omega}$ is $\approx 0.1\|\vec{\omega}\|$ — i.e., the error is reduced by an order of magnitude after each iteration.)

[Rotation by R transforms a scene point (X, Y, Z) into (X', Y', Z') such that $(X' \ Y' \ Z')^T = R(X \ Y \ Z)^T$. Let (x, y) and (x', y') be the images of (X, Y, Z) and (X', Y', Z'), respectively. Let r_{ij} be the elements of R. Then using (13) we have

$$x' = f \frac{X'}{Z'} = f \frac{r_{11}X + r_{12}Y + r_{13}Z}{r_{31}X + r_{32}Y + r_{33}Z} \cdot \frac{f/Z}{f/Z} = f \frac{r_{11}x + r_{12}y + r_{13}f}{r_{31}x + r_{32}y + r_{33}f} \tag{23}$$

and similarly

$$y' = f \frac{Y'}{Z'} = f \frac{r_{21}x + r_{22}y + r_{23}f}{r_{31}x + r_{32}y + r_{33}f}. \tag{24}$$

The direct application of the formulas for (x', y') yields non-integer pixel positions. To compute the transformed image, we apply the inverse transformation to (x', y') and determine the gray levels for the pixels of the transformed image by interpolating the gray levels of the original image.]

As regards the reliability of the method two questions must be answered. The first question is geometrical and can be formulated as follows: Given the spatial distribution and the orientations of the feature points in the image, which components of $\vec{\omega}$ can be computed? It is well known that the rotational image motion field (optical flow) can be zero only at the point where the direction of the axis of rotation (AOR) pierces the image plane [1, 7]. The rotational normal motion field (normal flow), however, can be zero when the normal (gradient) direction is orthogonal to the direction of the motion field, i.e. $\vec{n} \cdot \dot{\vec{r}}_\omega = 0$. However, this can cause problems only if the normal flow is (near) zero consistently at all (or the great majority of) the feature points. This can happen only if all normal (gradient) directions are orthogonal to the conic sections which are the intersections of the image plane with the circular cones centered at the focal point of the camera and having $\vec{\omega}$ as their axis. More generally, if all the normal directions are orthogonal to such a family of conic sections for which the unit vector $\vec{r}_c$ is the cone axis, the rotational velocity component $(\vec{r}_c \cdot \vec{\omega})\vec{r}_c$ cannot be detected. We say in this situation that the normal flow resulting from the rotational velocity $(\vec{r}_c \cdot \vec{\omega})\vec{r}_c$ belongs to the null space of the feature matrix A [21]. Fortunately, if such $\vec{\omega}$'s exist, the positive definite (semidefinite) matrix $A^T A$ must have a large condition number (the ratio of
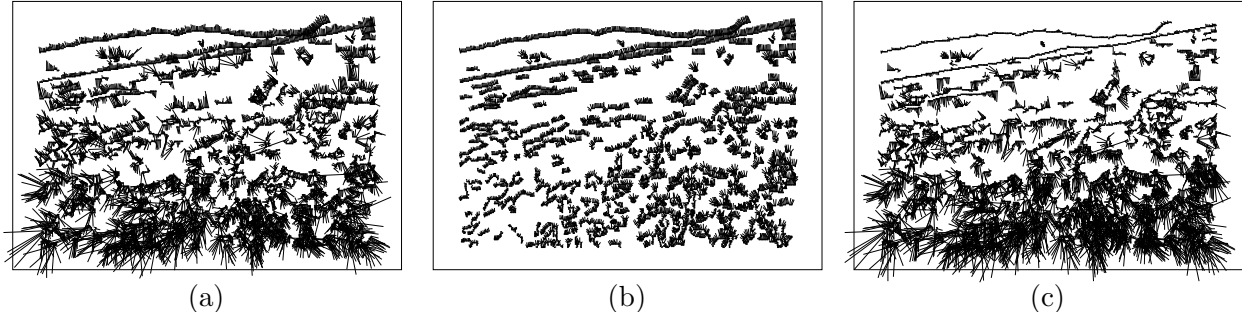
Figure 5: (a) Computed normal flow in the 100th frame of the first sequence. (b) Estimated rotational flow for the same frame. (c) Residual (translational) flow for the frame after subtracting the rotational flow.

its largest to its smallest eigenvalue); thus the existence of such situations is easy to detect by examining the eigenvalues of $A^T A$.

The second question is numerical and can be formulated as follows: Given the spatial distribution and the orientations of the feature points in the image, and the accuracy with which the normal flow can be computed, how accurately can $\vec{\omega}$ be computed and what can be done to increase the numerical accuracy of the method? This question is also related to the condition number of $A^T A$ since the errors in the computed $\vec{\omega}$ are proportional to the errors in the normal flow, where the condition number of $A^T A$ is the approximate proportionality coefficient. If the condition number is small we expect that the solution of (22) is reliable and that a few iterations of our algorithm will be enough to obtain a reliable estimate of $\vec{\omega}$. In all the examples shown in Figures 3 and 4 the condition numbers are smaller than 50.

As an example of the performance of our algorithm we show the results of applying it to the 100th frame of the first sequence in Figure 3. Figure 5a shows the normal flow and Figures 5b and 5c respectively show the estimated rotational and residual translational flow after two iterations. The condition number for the feature matrix $A^T A$ was ≈ 15; it remained close to this value through the entire sequence. The rotational velocity was estimated in two iterations. After the first iteration the estimated $\vec{\omega} = (\omega_x \ \omega_y \ \omega_z)^T$ (in camera coordinates) was $(0.000784 \ 0.000203 \ -0.000512)^T$ rad/frame, and after two iterations it was $(0.000775 \ 0.000204 \ -0.000528)^T$ rad/frame, so that $\|\Delta\vec{\omega}\| < 0.000011$ rad/frame.

3.5   Smoothing Algorithms

We saw in Section 3.1 that we need only smooth the effects (on the image sequence) of the rotational part of the motion. In Section 3.4 we showed how the rotational velocity $\vec{\omega}$ of the camera can be estimated in each frame. We cannot simply "smooth" the image sequence by using these estimates to derotate the frames; if we did this, we would eliminate the effects of all the rotational motion, including the smooth motion. But since the camera is not always pointing in the same direction (e.g., the platform may make turns, go over hills, etc.), this would cause large parts of the images to be lost.

We saw in Appendix that the non-smooth part of the rotational velocity of a ground vehicle has larger magnitude than the smooth part. However, we saw in Section 2.2 that the cumulative

13

effects of the non-smooth part on the orientation of the platform (the angles between the platform frame and the Darboux frame) remain relatively small. It follows that eliminating the effects of the non-smooth part (only) will not cause loss of large parts of the images.

To smooth the image sequence, we will first estimate the smooth part $\vec{\omega}_f$ of $\vec{\omega}$. We will then use the residual rotational velocity $\vec{\omega} - \vec{\omega}_f$, which corresponds to the non-smooth part of $\vec{\omega}$, to construct a sequence of rotation matrices; we can then use these matrices to correct the corresponding frames as in (23-24). The matrices can be constructed as follows: When the integral of the residual velocity $\vec{\mathbf{w}}_r = \int_0^t \vec{\omega} - \vec{\omega}_f \, dt$ is small its components W_1, W_2, and W_3 (in any coordinate frame) are approximately equal to the angles between the axes of that frame and the directions that the axes would have had if the rotational velocity of the camera were $\vec{\omega}_f$. The first order approximation of the rotation matrix R (see (7)) defined by $\vec{\mathbf{w}}_r$ is then

$$R = \begin{pmatrix} 1 & -W_3 & W_2 \\ W_3 & 1 & -W_1 \\ -W_2 & W_1 & 1 \end{pmatrix} + \mathcal{O}(\|\vec{\mathbf{w}}_r\|^2). \tag{25}$$

To use this matrix to correct the images as in (23-24), we need only transform it to a rotation matrix in the camera frame. For example, if the given frame is the platform frame, the matrix in the camera frame is $R_c^T R R_c$. We will in fact use the components in the platform frame (whose axes are the roll, pitch, and yaw axes) in this section.

We saw in the Appendix that a vehicle driven on a well-designed road (or a well-driven vehicle on any terrain) may undergo smooth rotations when it turns or goes over a hill. A smooth turn involves a transition from a straight part of the road to a circular arc part, so that a zero yaw velocity is followed by a constant, non-zero yaw velocity. (The transition is not abrupt, but the transition period is quite brief.) If the turn is banked, the transition between the unbanked and banked parts of the road results in a brief period of nonzero roll velocity. Finally, when the vehicle crests a hill (or passes through the bottom of a depression) there is a period of approximately constant, non-zero pitch velocity. Thus piecewise constant fits are reasonable approximations to the smooth rotational components of a vehicle's velocity around its roll, pitch, and yaw axes. Since our camera is fixed relative to the vehicle, the same is true about the components of the camera's rotational motion around these axes. We can therefore estimate the smooth part of $\vec{\omega}$ by "fitting" piecewise constant functions to these components. At the same time, the rotational velocities about these axes are always small; hence in doing the fitting, we should also try to keep the magnitudes of the components as small as possible.

Figure 6 shows the (unsmoothed) roll, pitch, and yaw components of $\vec{\omega}$ that were estimated from the first image sequence in Figure 4. We see that all three components fluctuate strongly, with major periods of oscillation on the order of 20-30 frames (i.e., one second or less). The components also remain relatively small in amplitude, usually less than 4×10^{-3} rad/frame; the fluctuations in the roll and pitch components have the highest amplitudes, as predicted at the end of the Appendix.

When we "fit" piecewise constant functions ω_f of small magnitude to the rotational velocity components ω, it is important that we (try to) keep the integrals of the residual velocities $W_r = \int_0^t \omega - \omega_f \, dt$ small, so that our method of correcting the frames, described earlier in this section, can be used. In the remainder of this section we will describe two algorithms for doing the "fitting". One algorithm is able to fit piecewise constant functions to the rotational velocities because it allows unlimited time delays before computing the fits. The algorithm actually fits a piecewise linear function to the integral of each component, rather than a piecewise constant function to
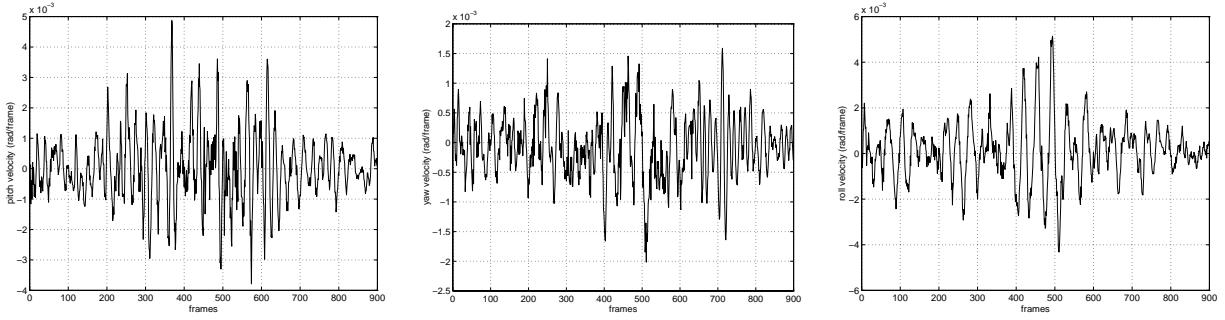
Figure 6: Raw velocity components for the third sequence (the two top rows in Figure 5). Left to right: pitch, yaw, roll.

the component itself. This has two advantages: (1) It allows the algorithm to incorporate explicit bounds on the magnitude of W_r. (2) Fitting to an integral is easier than fitting to the original component, since the integrals are much less noisy, as we see in the top row of Figure 7.

The algorithm takes $W = \int_0^t \omega\,dt$ as input and produces a polygonal approximation $W_f = \int_0^t \omega_f dt$ such that $|W_r| = |W - W_f|$ is smaller than a preassigned bound h. [Typical values of h are 0.05 ($\approx 3°$), 0.025, and 0.0125 radians for the roll, pitch, and yaw components, respectively.] For some initial time interval, there exist rays ρ through the origin (in fact, an angular sector of such rays) such that $|W - \rho| \leq h$. If beyond some time t_0 this is no longer true, then just prior to t_0 there will usually be exactly one ray ρ_0 (shown as a dashed line in Figure 8) for which it is true; we take this ρ_0, up to the last point at which it crosses W (say at time t_0^*), as the first segment in our polygonal approximation. We now take the crossing point as our new origin and repeat the process. After we have constructed W_f in this way, we can now obtain ω_f as the derivative of W_f.

The bottom row of Figure 7 shows the results of applying this algorithm to the sequence. The thin jagged lines correspond to actual estimated velocities; note that they differ by a small amount from the desired (thick line) velocities. These estimates are all quite small, confirming that, as already observed, during the 30-second period shown in the figures, relatively large unsmooth velocities can be effectively replaced by small smooth velocities.

We now describe the limited time-delay algorithm[1]. This algorithm assumes that the smoothed component ω_f is zero at $t = 0$, and repeatedly "corrects" its estimate of ω_f, based on the behavior of W_r, in such a way as to keep it (almost always) bounded. Thus for this algorithm too, the W_rs remain small, and so can be used to correct the frames.

Let the time delay be d frames; we will show examples with $d = 5, 15$, and 30 frames (note that 30 frames corresponds to a one-second delay). Initially we set $\omega_f = 0$. We use the values of ω computed for frames 1 through d to estimate the value of the residual $W_r(d)$; we use this estimate to compute a correction δ of ω_f; we then replace ω_f by δ and proceed with estimating ω_f for frames 2, 3, and so on.

Let the index of the current frame be i. Using the values of ω for frames $i + 1, i + 2, \ldots, i + d$

---

[1]This algorithm can be adapted for real-time operation by making the time delay zero.
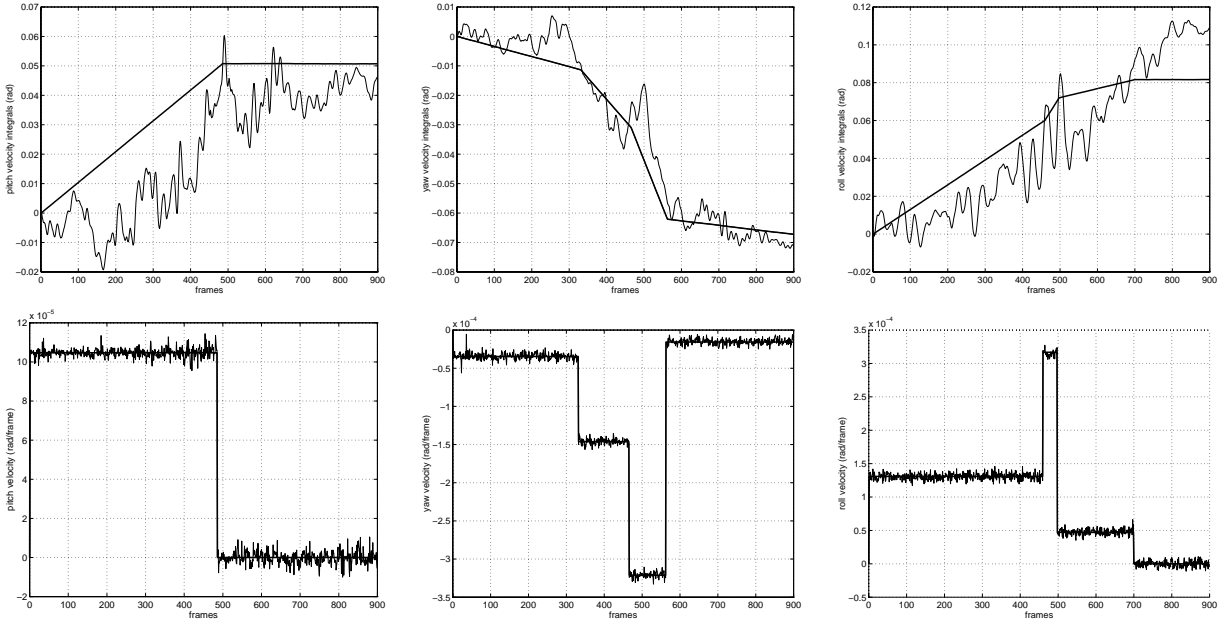
15

Figure 7: Results of the unlimited time-delay algorithm for the same sequence. First row: integrals of velocity components; thin lines correspond to integrals of raw velocity components and thick lines correspond to polygonal approximations of these integrals. Second row: thick lines correspond to desired smooth velocities and thin lines correspond to estimated velocities computed from the corrected image sequence.

we compute the predicted value of $W_r(i + d)$ as

$$W_r(i + d) = W_r(i) + \sum_{i=1}^{d} \omega(i) - d\omega_f(i).$$

Next, we compute

$$\delta = K_1 d^{-1} W_r(i + d);$$

if $|W_r(i + d)| > 0.75h$ we replace δ by

$$\delta + K_2 d^{-1}(W_r(i + d) - 0.75h) \quad \text{for} \quad W_r(i + d) > 0$$

and by

$$\delta + K_2 d^{-1}(W_r(i + d) + 0.75h) \quad \text{for} \quad W_r(i + d) < 0.$$

When ω_f and $W_r(i + d)$ have opposite signs we replace δ by 5δ. Finally, we compute $\omega_f(i + 1) = \omega_f(i) + \delta$. We have used $K_1 = 0.0005$ and $K_2 = 0.01$ in our experiments. Larger values of K_1 and (especially) K_2 can lead to oscillatory behavior of ω_f.

The results of applying the limited time-delay algorithm to the sequence using $d = 5, 15$, and 30 are shown in Figure 9. The first row shows velocity integrals and the second row shows corresponding smooth velocities. We see that the smooth velocity components are relatively small compared to the unsmooth velocity components of Figure 6. We also see that one obtains slightly better estimates using larger ds, but the difference between the estimates for different ds are not large. Finally, Figure 10 shows several frames of the sequence, smoothed using $d = 30$.
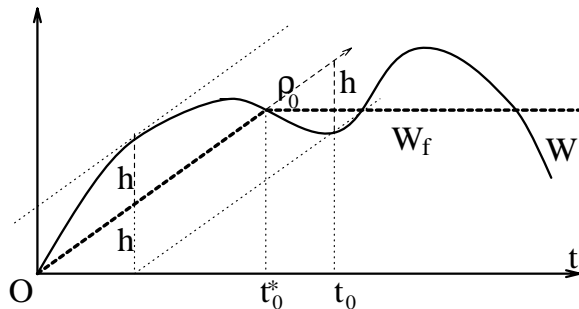
16

Figure 8: Polygonal approximation of W by W_f using the algorithm that allows unlimited time delays.

4 Conclusions

We have analyzed the smooth and non-smooth motions of a platform (person or vehicle) moving along the ground and have shown that only the rotational components of the non-smooth motion have significant perturbing effects on the images. We have analyzed the relationship between the shape of the terrain along which the platform travels and the parameters of the smooth motion of the platform. We have also used this analysis, together with the highway design recommendations in [3], to estimate the relative sizes of the smooth and non-smooth components of the motion. We have shown how to identify image points at which rotational image flow is dominant, and how to use such points to estimate the platform's rotation. Finally, we have described two algorithms that fit piecewise smooth rotational motions to these estimates and we have shown how the residual rotational motion (the difference between the estimated actual motion and the fitted motion) can be used to correct the images. We have presented results for an image sequence obtained from a camera carried by a ground vehicle moving across bumpy terrain; videos of several such sequences, before and after stabilization, demonstrate the effectiveness of our approach. These can be seen at **www.cs.gmu.edu/~zduric/ImageSmoothing/**.

5 Appendix: The Motion of a Ground Vehicle on a Rough Surface

A ground vehicle drives over roads or surfaces (for brevity: RSs) that have varying degrees of roughness [4]. An RS may be a paved, gravel, or dirt road; grassy, muddy, sandy, or gravel-strewn ground; and so forth. The degree of roughness of an RS will be considered as piecewise stationary, i.e. static in a statistical sense with parameters that remain constant over a finite time period. The roughness consists primarily of small irregularities in the RS (stones, litter, holes, etc.). The RS may also contain occasional roughness outliers such as rocks, bushes, potholes, speed bumps, etc., but we will ignore them in the discussion below.

In discussing the effects of the roughness of the RS on the motion of a ground vehicle we will assume, for simplicity, an ordinary, well-balanced four-wheeled vehicle moving on a planar surface that is smooth except for occasional small bumps (protrusions). The bumps are assumed to be small relative to the size of the wheels, so that the effect of a wheel passing over a bump is impulsive. (We could also allow the surface to have small depressions, but a large wheel cannot deeply penetrate a small depression, so the depressions have much smaller effects than the bumps.)
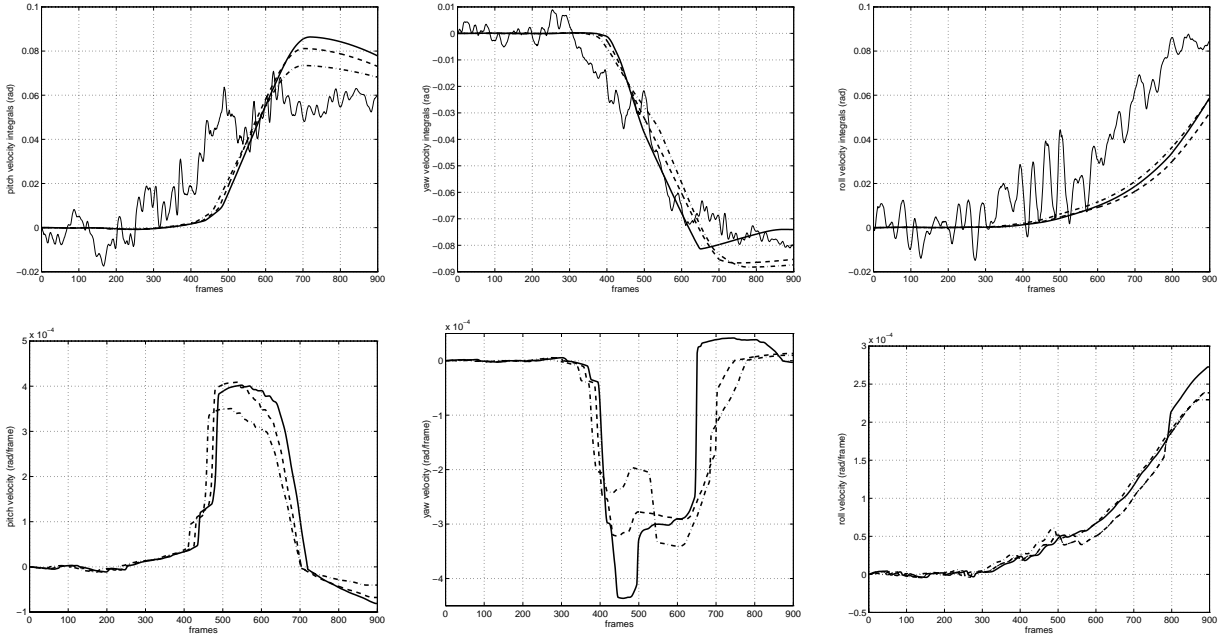
17

Figure 9: Results of the limited time-delay algorithm for the same sequence. First row: integrals of velocity components; thin lines correspond to integrals of raw velocity and thick lines correspond to smooth approximations of these integrals (full line, $d = 5$; dashed line, $d = 15$; dash-dot line, $d = 30$). Second row: desired and computed smooth velocities (full line, $d = 5$; dashed line, $d = 15$; dash-dot line, $d = 30$).

As the vehicle moves over rough terrain, each wheel hits bumps repeatedly. We assume that the vehicle has a suspension mechanism which integrates and damps the impulsive effects of the bumps. Each suspension element is modeled by a spring with damping; its characteristic function is a sine function multiplied by an exponential damping function (see [18, 24]). We assume that the suspension elements associated with the four wheels are independent of each other and are parallel to the up/down axis of the vehicle. (A discussion of the dynamics of the suspension of a ground vehicle on rough terrain can be found in [25, 26].)

On bumpy terrain the vehicle will usually hit new bumps while the effects of the previous bumps are still being felt. Each hit forces the suspension and adds to the accumulated energy in the spring; thus we can assume that the suspension is constantly oscillating, which has the effect of moving the corners of the vehicle up and down. The period of oscillation is typically on the order of 0.5 sec (see [18, 24]). In general, it takes several periods to damp out the spring; for example, the damping ratio provided by shock absorbers of passenger cars is in the range $0.2 - 0.4$. The maximum velocity of the oscillation is typically on the order of 0.1 m/sec.

Consider the coordinate system $Cxyz$ with origin at the center of mass C of the vehicle (see Figure 11). Let v_i be the velocity of corner C_i of the vehicle, and let the length and width of the
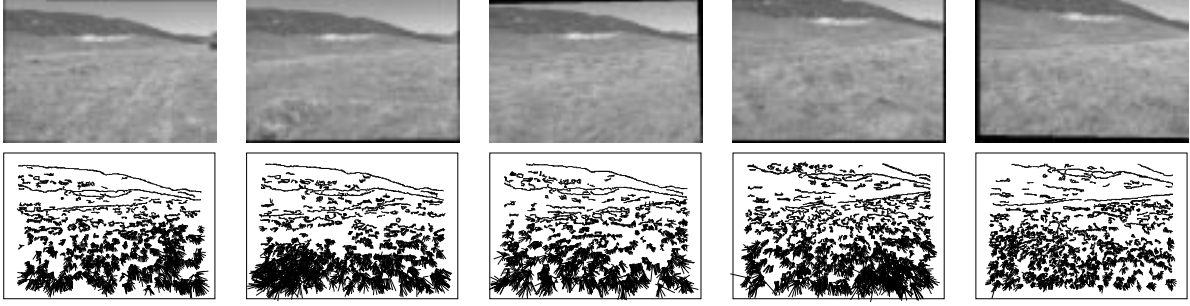
18

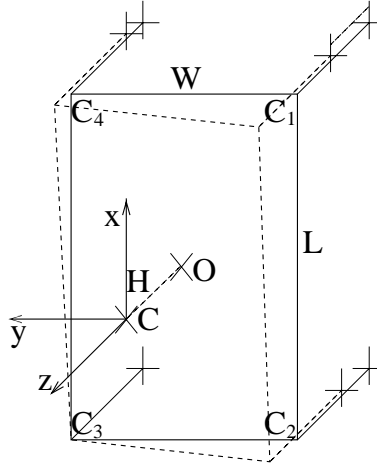Figure 10: Frames 0, 200, 400, 600, and 800 of the smoothed image sequence.

Figure 11: A possible motion of the base of the vehicle as the vehicle hits a bump.

vehicle be L and W. From the v_is we can compute the angular velocity matrix

$$\tilde{\Omega} \equiv \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & (v_3 - v_4)/L \\ 0 & 0 & (v_2 - v_3)/W \\ (v_4 - v_3)/L & (v_3 - v_2)/W & 0 \end{pmatrix}. \quad (26)$$

Note that any of the v_is can be positive or negative. Multiplication by $\tilde{\Omega}$ can be replaced by the vector product with the angular velocity vector $\vec{\omega} = \vec{\imath}(v_3 - v_2)/W + \vec{\jmath}(v_3 - v_4)/L$ where the rate of rotation around the x axis (the roll velocity) is $\omega_x = (v_3 - v_2)/W$ and the rate of rotation around the y axis (the pitch velocity) is $\omega_y = (v_3 - v_4)/L$. As noted above, we typically have $|v_i| < 0.1\,\mathrm{m/sec}$. If we assume that $W > 1\,\mathrm{m}$ and $L > 2\,\mathrm{m}$ we have $|\omega_x| < 0.2\,\mathrm{rad/sec}$ and $|\omega_y| < 0.1\,\mathrm{rad/sec}$. The yaw velocity component is $|\omega_z| = \mathcal{O}(v_i^2/\sqrt{W^2 + L^2})$ which is $|\omega_z| = \mathcal{O}(0.01)\,\mathrm{rad/sec}$. (For a complete derivation see [8].)

The translational velocity vector of the center C of the vehicle is obtained by using the velocities $v_1 - v_3$, $v_2 - v_3$, 0, and $v_4 - v_3$ for the corners and adding v_3 to the velocity in the direction of the z-axis. We thus have

$$\vec{T} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} -(v_4 - v_3)H/L \\ (v_2 - v_3)H/W \\ (v_2 + v_4)/2 \end{pmatrix}. \quad (27)$$

19

If we assume that $H < 0.5\,\mathrm{m}$ ($< W/2$) we have $|t_x| < 0.05\,\mathrm{m/sec}$, $|t_y| < 0.1\,\mathrm{m/sec}$, and $|t_z| < 0.1\,\mathrm{m/sec}$.

We can draw several conclusions from this discussion: (i) The effects of small bumps are of short duration, i.e. they can be considered to be impulsive. The suspension elements integrate and damp these effects, resulting in a set of out-of-phase oscillatory motions. (ii) The yaw component of rotation due to the effect of a bump is very small compared to the roll and pitch components. (iii) The translational effects of a bump are proportional to the velocities (or displacements) of the suspension elements and the dimensions of the vehicle and are quite small.

We can now compare the sizes of the velocity components which are due to the ideal motion of the vehicle — i.e., the velocity components of the Darboux frame (Section 2.1) — to the sizes of the velocity components which are due to departures of the vehicle frame from the Darboux frame (Section 2.2).

The translational velocity of the Darboux frame is just $v\vec{\mathbf{t}}$; thus the magnitude of the translational velocity is just v. If $v = 10\,\mathrm{m/sec}$ ($= 36\,\mathrm{km/hr} \approx 22\,\mathrm{mi/hr}$) this velocity is much larger than the velocities which are due to departures of the vehicle from the Darboux frame, which, as we have just seen, are on the order of $0.1\mathrm{m/sec}$ or less.

The rotational velocity of the Darboux frame is $v\vec{\omega}_d = v(\tau_g\vec{\mathbf{t}} + \kappa_n\vec{\mathbf{v}} + \kappa_g\vec{\mathbf{s}})$; thus the magnitude of the rotational velocity is $v\sqrt{\tau_g^2 + \kappa_n^2 + \kappa_g^2}$. In the following paragraphs we will estimate bounds on τ_g, κ_n, and κ_g. Our analysis is based on the analyses in [4, 11, 24] and on the highway design recommendations published by the American Association of State Highway Officials [3]. Note that for safety reasons, a driver is likely to drive more slowly on terrain than on a highway, so bounds derived from our highway analysis should also hold for cross-country driving.

Good highway design allows a driver to make turns at constant angular velocities, and to follow spiral arcs in transitioning in and out of turns, in order to reduce undesirable acceleration effects on the vehicle. A well-designed highway turn also has a transverse slope, with the outside higher than the inside, to counterbalance the centrifugal force on the turning vehicle. Thus the ideal (smooth) motion of a vehicle has piecewise constant translational and rotational velocity components, with smooth transitions between them. Note that the translational components are constant in the vehicle coordinate frame even when the vehicle is turning, unless it slows down to make the turn.

To illustrate the typical sizes of these components, consider a ground vehicle moving with velocity v along a plane curve Γ on the surface Σ. If Σ is a plane and Γ is a circular arc with radius of curvature $\rho_g = |\kappa_g|^{-1}$ (i.e., the vehicle is turning with a constant steering angle), the angular velocity of the vehicle is $v\vec{\omega}_d = v\kappa_g\vec{\mathbf{s}}$ and there is a centripetal acceleration $\vec{a}_c = v^2\kappa_g\vec{\mathbf{v}}$ at the vehicle's center [23]. As a result there is a centrifugal force on the vehicle proportional to $\|\vec{a}_c\|$ and the mass of the vehicle. If skidding is to be avoided the limit on $\|\vec{a}_c\|$ (see [4]) is given by

$$\|\vec{a}_c\| = v^2\kappa_g \leq g(\tan\alpha + \mu_a) \tag{28}$$

where g is the gravitational acceleration, α is the transverse slope, and μ_a is the coefficient of adhesion between the wheels and the surface. [Typical values of μ_a range from $0.8 - 0.9$ for dry asphalt and concrete to 0.1 for ice (see [24], page 26).] From (28) we have either a lower bound on ρ_g for a given v or an upper bound on v for a given ρ_g. For example, if $v = 30\,\mathrm{m/sec}$ ($\approx 108\,\mathrm{km/hr}$), $\alpha = 0.05\,\mathrm{rad}$, and $\mu_a = 0.2$ from $v^2/\rho_g < 0.25g$ we have $\rho_g > 367\,\mathrm{m}$. This yields an upper bound on the yaw angular velocity of $< v|\kappa_g| = v/\rho_g \approx 0.08\,\mathrm{rad/sec}$, which is somewhat larger than the

yaw angular velocity arising from the departures from Darboux motion.

Other dynamic constraints on a vehicle such as limits on torques and forces can be used to obtain constraints on τ_g and κ_n. (These and other considerations such as safety and comfort were used in [3] to make recommendations for highway design; for a summary of these recommendations see [8].) For both vertical curves (crossing a hill) and turning curves the (recommended lower bound on the) radius of curvature $\rho_{\min}$ grows with the square of the design velocity v_d. However, the resulting (design) yaw and pitch angular velocities are limited by $v_d/\rho_{\min}$. Thus for smaller velocities v the vehicle can negotiate tighter vertical and turning curves and thus have even larger values of the yaw and pitch angular velocities. Typical values of the roll and pitch angular velocities are given in [8].

For realistic vehicle speeds we can conclude the following about the impulsive and smooth translational and rotational velocity components of the vehicle [8]. The impulsive effects on the translational velocity are approximately two orders of magnitude smaller than the smooth velocity components themselves. Impulsive effects on the yaw angular velocity are somewhat smaller than the smooth yaw component arising from worst-case turns of the RS; for moderate turns the impulsive effects are comparable in size to the smooth yaw velocity. Impulsive effects on the roll angular velocity are approximately an order of magnitude larger than the smooth roll component arising from worst-case twists (and turns) of the RS; for gentler twists the smooth roll velocity is even smaller. Similarly, impulsive effects on the pitch angular velocity are approximately an order of magnitude larger than the smooth pitch velocity arising from worst-case changes of vertical slope (i.e., vertical curves) of the RS; for gentler vertical curves the smooth pitch angular is even smaller. (The impulsive effects are not significantly affected by turns, twists, or vertical slope.) We can thus conclude that impulsive effects on the roll and pitch angular velocities are significant and larger than the corresponding smooth velocities, and that impulsive effects on the yaw angular velocity are on the order of the smooth yaw velocity.

References

[1] Y. Aloimonos and Z. Duric. Estimating the heading direction using normal flow. *International Journal of Computer Vision*, 13:33–56, 1994.

[2] N. Ancona and T. Poggio. Optical flow from 1D correlation: Application to a simple time-to-crash detector. In *Proc. DARPA Image Understanding Workshop*, pages 673–682, 1993.

[3] American Association of State Highway Officials. *A Policy on Geometric Design of Rural Highways* (9th edition). AASHO, 1977.

[4] M. G. Bekker. *Introduction to Terrain-Vehicle Systems*. The University of Michigan Press, Ann Arbor, 1969.

[5] P. J. Burt and P. Anandan. Image stabilization by registration to a reference mosaic. In *Proc. ARPA Image Understanding Workshop*, pages 425–434, 1994.

[6] J. Cutting. *Perception with an Eye for Motion*. MIT Press, Cambridge, MA, 1986.

[7] Z. Duric, A. Rosenfeld, and L. S. Davis. Egomotion analysis based on the Frenet-Serret motion model. *International Journal of Computer Vision*, 15:105–122, 1995.

[8] Z. Duric, and A. Rosenfeld. Stabilization of Image Sequences. Technical Report CAR–TR–778, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, 1995.

[9] Z. Duric and A. Rosenfeld. Image sequence stabilization in real time. *Real-Time Imaging*, 2:271–284, 1996.

[10] Z. Duric, A. Rosenfeld, and J. Duncan. The applicability of Green's theorem to computation of rate of approach. *International Journal of Computer Vision*, 31:83–98, 1999.

[11] J. R. Ellis. *Vehicle Handling Dynamics*. Mechanical Engineering Publications Limited, London, 1994.

[12] A. Giachetti, M. Campani, and V. Torre. The use of optical flow for road navigation. *IEEE Transactions on Robotics and Automation*, 14:34–48, 1998.

[13] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. J. Burt. Real-time scene stabilization and mosaic construction. In *Proc. ARPA Image Understanding Workshop*, pages 457–465, 1994.

[14] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, 1994.

[15] T. Kanade. Development of a video-rate stereo machine. In *Proc. ARPA Image Understanding Workshop*, pages 549–557. 1994.

[16] J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, MA, 1990.

[17] E. Kreyszig. *Differential Geometry*. University of Toronto Press, Toronto, Canada, 1959.

[18] R. Oldenburger. *Mathematical Engineering Analysis*. Macmillan, New York, 1950.

[19] Z. Sheng and K. Yamafuji. Postural stability of a human riding a unicycle and its emulation by a robot. *IEEE Transactions on Robotics and Automation*, 13:709–720, 1997.

[20] S. Srinivasan and R. Chellappa. Fast structure from motion recovery applied to 3D image stabilization. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3357–3360, 1999.

[21] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[22] A. Verri and T. Poggio. Against quantitative optical flow. In *Proc. International Conference on Computer Vision*, pages 171–180, 1987.

[23] J. H. Williams, Jr. *Fundamentals of Applied Dynamics*. Wiley, New York, 1996.

[24] J. Y. Wong. *Theory of Ground Vehicles* (2nd edition). Wiley, New York, 1993.

[25] Y. S. Yao and R. Chellappa. Model-based vehicular motion and structure estimation. In *Proc. International Conference on Robotics and Automation*, 1995.

[26] Y. S. Yao and R. Chellappa. Selective stabilization of images acquired by unmanned ground vehicles. *IEEE Transactions on Robotics and Automation*, 13:693–708, 1997.

[27] V. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics, Champaign, IL, 1998.

[28] A. Zomet, A. Peleg, and C. Arora. Rectified mosaicing: Mosaics without the curl. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages II:459-465, 2000.