96-22

Image Sequence Stabilization in Real Time

This paper describes a method of *stabilizing* image sequences obtained by a camera carried by a ground vehicle. The motion of the vehicle can usually be regarded as consisting of a desired smooth motion combined with an undesired non-smooth motion that includes impulsive or high-frequency components. The goal of the stabilization process is to correct the images so that they are approximately the same as the images that would have been obtained if the motion of the vehicle had been smooth.

We analyse the smooth and non-smooth motions of a ground vehicle and show that only the rotational components of the non-smooth motion have significant perturbing effects on the images. We show how to identify image points at which rotational image flow is dominant, and how to use such points to estimate the vehicle's rotation. Finally, we describe an algorithm that fits smooth (ideally, piecewise constant) rotational motions to these estimates; the residual rotational motion can then be used to correct the images. We have obtained good results for several image sequences obtained from a camera carried by a ground vehicle moving across bumpy terrain.

© 1996 Academic Press Limited

Zoran Duric*† and Azriel Rosenfeld*

*Computer Vision Laboratory, University of Maryland, College Park, MD 20742-3275, USA
†Machine Learning and Inference Laboratory, George Mason University, Fairfax, VA 22030-4444, USA

The Problem

In this paper the problem of image stabilization is defined as follows: a camera mounted on a moving vehicle collects a sequence of images. The vehicle is trying to move along a smooth trajectory. In fact, however, the motion of the vehicle varies from the desired smooth motion; for a ground vehicle, this would primarily be due to roughness of the terrain. The goal of image stabilization 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 least piecewise smooth.

The motion of a vehicle is described by the translational velocity of its center of mass and the rotational velocities of its principal axes; these quantities are functions of time [5]. We consider a vehicle's motion to be non-smooth when significant impulsive or high-frequency changes occur in (some of) these velocity functions. We will stabilize the image sequence by smoothing out the effects of these impulsive changes.

The problem of image sequence stabilization has been considered by a number of investigators [6,7,11,13,20]. They all agree that image stabilization deals with the removal of the effects of unwanted motion from an image sequence. In Burt and Anandan [6] and Hansen *et al.* [11], information from previous frames is used to create a mosaic image; subsequent images are then registered to the mosaic image and used to update it. Stabilization is then

based on computing total image motion as well as relating each new frame to the mosaic image. This allows new frames to have little or (in the extreme) no overlap with the preceding frame(s); this may happen when a narrow field of view camera is used (say 5°) or when the time between frames is large (much larger than the usually assumed 1/30 s). Since most of the flow computation is done locally the affine flow model is used. In Irani *et al.* [13], motion and structure are computed from image flow for an assumed planar patch; the planar image flow is then extrapolated to the whole image and subtracted from the original image flow. The residual flow is rotationless, although it does not correspond to the true translational flow; the image sequence that contains only the residual flow is regarded as stabilized. In Vieville *et al.* [20], stabilization is achieved by aligning linear segments extracted from the images with the absolute vertical direction; this eliminates rotation around the viewing direction. In Davis *et al.* [7] several methods of computing rotational motion and de-rotating images are described, including a method of fitting an affine motion model to the far field and using the parameters of the model to transform the images.

This paper analyses the problem of stabilizing image sequences obtained by a ground vehicle. In the next section we review the equations describing the image motion field in images of a static scene obtained by a rigidly moving observer. We then introduce a model for ground vehicle motion and discuss the sources of non-smoothness in this type of motion as well as the relative sizes of the smooth and non-smooth velocity components. Following this, we show that only the rotational components of the non-smooth motion have significant perturbing effects on image sequences collected by the vehicle. We show how to identify image points at which rotational image flow is dominant, and how to use such points to estimate the vehicle's rotation. Finally, we describe an algorithm that fits smooth (ideally, piecewise constant) rotational motions to these estimates; the residual rotational motion can then be used to correct the images. We have obtained good results for several image sequences obtained from a camera carried by a ground vehicle moving across bumpy terrain.

Preliminaries

In this section, we summarize the equations of motion for a rigid body moving in a static environment, using both body-centered and static coordinate systems. We then derive projected motion equations for images obtained using the plane perspective imaging model, and we derive the relationship between the image velocities and the projected motion.

Rigid Body Motion

Let us associate a coordinate system with a body that moves rigidly in a static environment. Let C_0 and C_t be the coordinate systems of the body at times 0 and t and let O be the center of the body (and the origin of the body coordinate system). We also take C_0 to be the static coordinate system – i.e., the two systems coincide at time 0. A static point E whose position in C_0 is $\vec{r}_e(0)$ has position $\vec{r}_e(t)$ in C_t given by

$$\vec{r}_e(t) = R^T(t)\left[\vec{r}_e(0) - \vec{d}(t)\right]. \tag{1}$$

If we differentiate (1) with respect to time and use the fact that $\vec{r}_e(0) = R(t)\,\vec{r}_e(t) + \vec{d}(t)$, we obtain

$$\dot{\vec{r}}_e = \dot{R}^T\left(\vec{r}(0) - \vec{d}\right) - R^T\dot{\vec{d}} = \dot{R}^T R\vec{r}_e - R^T\dot{\vec{d}} \equiv -\Omega\vec{r}_e - \vec{T}. \tag{2}$$

The skew matrix $\Omega = -\dot{R}^T R = R^T\dot{R}$ is the rotational velocity matrix and $\vec{T} = R^T\dot{\vec{d}} = (t_x\, t_y\, t_z)^T$ is the translational velocity vector. Multiplying a vector $\vec{a}$ by the matrix Ω can be replaced by taking the cross product $\vec{w} \times \vec{a}$ where $\vec{w} = (w_x\, w_y\, w_z)^T$ is the rotational velocity vector. For a given $\vec{r}_e = (X\, Y\, Z)^T$ we have

$$\dot{\vec{r}} = \left(\dot{X}\, \dot{Y}\, \dot{Z}\right)^T = -\vec{w} \times \vec{r}_e - \vec{T} \tag{3}$$

Later in this paper we will need some basic facts about first-order approximations to rotations and rotational velocities. A (signed) rotational axis 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

$$R = \cos\,\delta I + (1 - \cos\,\delta)\begin{pmatrix} c_x^2 & c_x c_y & c_x c_z \\ c_y c_x & c_y^2 & c_y c_z \\ c_z c_x & c_z c_y & c_z^2 \end{pmatrix}$$

$$+ \sin\,\delta\begin{pmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{pmatrix} \tag{4}$$

where I is the identity matrix When δ is small ($\delta < 0.1$rad) we have $\cos\,\delta \approx 1$, $\sin\,\delta \approx \delta$, and

$$R = I + \delta C + \mathcal{O}(\delta^2) \tag{5}$$

where the skew matrix C is the matrix factor in the last term on the r.h.s. of (4) and $\mathcal{O}()$ means 'on the order of'. For a small rotational angle δ the rotation given by (4) is approximately equal to three successive rotations (in any

order) around the x, y, and z axes through the angles $c_x\delta$, $c_y\delta$, and $c_z\delta$, respectively.

We obtain an analogous expression in terms of the rotational velocity matrix Ω by expanding $R(t)$ in a Taylor series at $t = t_0$:

$$R(t) = R(t_0) + \Delta t \, \dot{R}(t_0) + \mathcal{O}\left((\Delta t)^2\right)$$

$$= R(t_0)\left[I + \Delta t \, R^T(t_0)\dot{R}(t_0)\right] + \mathcal{O}\left((\Delta t)^2\right)$$

$$= R(t_0)\left[I + \Delta t \, \Omega(t_0)\right] + \mathcal{O}\left((\Delta t)^2\right)$$

where we have used $\Omega(t_0) = R^T(t_0) \, \dot{R}(t_0)$, $\Delta t = t - t_0$. If we are interested only in the relative rotation between t and t_0, we can set $R(t_0) = I$ and obtain

$$R(t) = I + \Delta t \, \Omega(t_0) + \mathcal{O}\left((\Delta t)^2\right). \qquad (6)$$

The Image Motion Field and the Optical Flow Field

In this section we refer to the moving body as the camera and to the static environment as the scene. Let (X, Y, Z) denote the Cartesian coordinates of a scene point with respect to the camera frame (see Figure 1), and let (x, y, f) denote the corresponding coordinates in the image plane. f is the focal length of the camera. The projection equations for the plane perspective projection are given by

$$x = \frac{X}{Z} f, \, y = \frac{Y}{Z} f. \qquad (7)$$

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 (7) with respect to time and using (3):
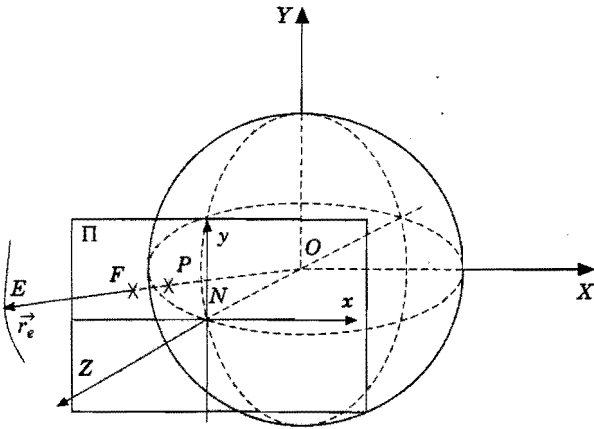

Figure 1. The plane perspective projection image of (X, Y, Z) is $(fX/Z, fY/Z, f)$.

$$\dot{x} = \frac{-t_x f + xt_z}{Z} + w_x \frac{xy}{f} - w_y\left(\frac{x^2}{f} + f\right) + w_z y, \qquad (8)$$

$$\dot{y} = \frac{-t_y f + yt_z}{Z} + w_x\left(\frac{y^2}{f} + f\right) - w_y \frac{xy}{f} - w_z x. \qquad (9)$$

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 $\vec{r} = x\vec{i} + y\vec{j} + f\vec{k}$.

If we choose a unit direction vector, $\vec{n}_r$ at the image point $\vec{r}$ and call it the normal direction, then the *normal motion field* at $\vec{r}$ is $\dot{\vec{r}}_n = (\dot{\vec{r}} \cdot \vec{n}_r)\, \vec{n}_r$. $\vec{n}_r$ can be chosen in various ways; the usual choice is the direction of the image intensity gradient.

Let $I(x, y, t)$ be the image intensity function. The time derivative of I can be written as

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = \left(I_x \vec{i} + I_y \vec{j}\right) \cdot \left(\dot{x}\vec{i} + \dot{y}\vec{j}\right) + I_t$$

$$= \nabla I \cdot \dot{\vec{r}} + I_t$$

where ∇I is the image gradient and the subscripts denote partial derivatives.

If we assume $dI/dt = 0$, i.e. that the image intensity does not vary with time [12], then we have $\nabla I \cdot \vec{u} + I_t = 0$. The vector field $\vec{u}$ in this expression is called the *optical flow*. If we choose the normal direction $\vec{n}_r$ to be the image gradient direction, i.e. $\vec{n}_r \equiv \nabla I \, \|\nabla I\|$, we then have

$$\vec{u}_n = \left(\vec{u} \cdot \vec{n}_r\right) \vec{n}_r = \frac{-I_t \nabla I}{\|\nabla I\|^2} \qquad (10)$$

where $\vec{u}_n$ is called the *normal flow*.

It was shown by Verri and Poggio [19] that the magnitude of the difference between $\vec{u}_n$ and the normal motion field, $\dot{\vec{r}}_n$, is inversely proportional to the magnitude of the image gradient. Hence, $\dot{\vec{r}}_n \approx \vec{u}_n$ when $\|\nabla I\|$ is large. Equation (10) thus provides an approximate relationship between the 3-D motion of a scene point in the camera frame and the image derivatives at the corresponding image point. We will use this approximation later in this paper.

The Vehicle Motion Model

We have defined image stabilization as smoothing out 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 non-smoothness could involve the three parameters of the translational velocity of the vehicle'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, stabilization could involve smoothing in a six-dimensional space.

The ideal motion of a ground vehicle does not have six degrees of freedom. If the motion is (approximately) smooth it can be described as motion along a smooth trajectory, Γ, lying on a smooth surface, Σ. Moreover, we shall assume that the axes of the vehicle (the fore/aft, crosswise, and up/down axes) are respectively parallel to the axes of the *Darboux frame* defined by Γ and Σ. These axes are defined by the tangent $\vec{t}$ to Γ (and Σ), the second tangent $\vec{v}$ to Σ (orthogonal to $\vec{t}$), and the normal $\vec{s}$ to Σ (see Figure 2). Our assumption about the axes is reasonable for the ordinary motions of standard types of ground vehicles; in particular, we are assuming that the first two vehicle axes are parallel to the surface (this would not be the case, e.g., if the front wheels and the rear wheels were not the same size) and that the vehicle's motion is parallel to its first axis (this might not be the case, e.g., if the vehicle were skidding). In the next section 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 to describe non-smooth vehicle motion.

Non-smooth vehicle motion can involve impulsive changes in any of the translational and/or rotational velocity functions. In the case of a ground vehicle, however, some of these changes can be expected to have small amplitudes (and hence to be unimportant for stabilization). Thus, the smoothing problem becomes simpler since the
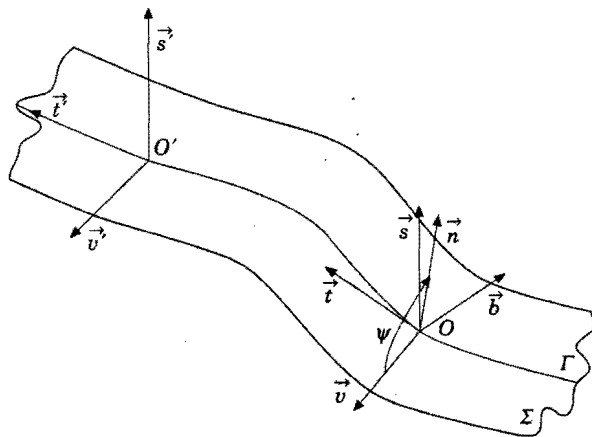
smoothing can be done in a space of dimensionality lower than six. Knowledge about the motion obtained from other sensors (i.e., not from images) can also be used in the smoothing process; e.g., if the steering angle is known it can be used as 'ground truth' for the desired component of rotational velocity around the normal ($=$ up/down) axis of the vehicle.

Later we will compare the sizes of the smooth and non-smooth velocity components of a ground vehicle.

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 $\vec{t}$, normal $\vec{n}$, and binormal $\vec{b}$ of Γ. The triple $(\vec{t}, \vec{n}, \vec{b})$ is called the *moving trihedron* or *Frenet-Serret coordinate frame*. We have the Frenet-Serret formulas [16]

$$\vec{t}' = \kappa\vec{n}, \ \vec{n}' = -\kappa\vec{t} + \tau\vec{b}, \ \vec{b}' = -\tau\vec{n} \qquad (11)$$

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* $(\vec{t}, \vec{v}, \vec{s})$ [15,16]. We take the first unit vector of the frame to be the tangent $\vec{t}$ of Γ and the surface normal $\vec{s}$ to be the third frame vector; finally we obtain the second frame vector as $\vec{v} = \vec{s} \times \vec{t}$ (see Figure 2). Note that $\vec{t}$ and $\vec{v}$ lie in the tangent plane of Σ. Since the vector $\vec{t}$ belongs to both the *Otnb* and *Otvs* frames they differ only by a rotation around $\vec{t}$, say through an angle $\psi \equiv \psi(s)$. We thus have

$$\begin{pmatrix} \vec{v} \\ \vec{s} \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \vec{n} \\ \vec{b} \end{pmatrix}. \qquad (12)$$

The derivatives of $\vec{t}, \vec{v}, \vec{s}$ with respect to arc length along Γ can be found from (11) and (12):

$$\vec{t}' = \kappa_g\vec{v} - \kappa_n\vec{s}, \ \vec{v}' = -\kappa_g\vec{t} + \tau_g\vec{s}, \ \vec{s}' = \kappa_n\vec{t} - \tau_g\vec{v} \qquad (13)$$

where

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

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

The translational velocity $\vec{T}$ of O is just $\vec{t}$ and the rotational velocity of the *Otvs* frame is given by the vector

Figure 2. The Darboux frame moves along the path Γ which lies on the surface Σ.

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

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

If, instead of using the arc length s as a parameter, the time t is used, the rotational velocity, $\vec{w}_d$, 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 (SSM)*.

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

The motion of the vehicle 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 $v\vec{w}_d = v (\tau_g \vec{t} + \kappa_n \vec{v} + \kappa_g \vec{s})$ and its translational velocity is $v\vec{t}$. We denote the rotational velocity of the $C\xi\eta\zeta$ (vehicle) frame by $\vec{w}_v$ 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 vehicle (the velocity of C) is the sum of three terms: (i) the translational velocity of the Darboux frame $v\vec{t}$, (ii) the translational velocity $T_{v/d} \equiv \dot{\vec{d}}_{v/d}$, and (iii) the displacement $v\vec{w}_d \times \vec{d}_{v/d}$ due to rotation of C in the *Otvs* frame. The translational velocity of the vehicle expressed in the *Otvs* frame is thus $v\vec{w}_d \times \vec{d}_{v/d} + v\vec{t} + \dot{\vec{d}}_{v/d}$; its translational velocity in the $C\xi\eta\zeta$ frame is

$$\vec{T}_v = R_{v/d}^T \left(v\vec{w}_d \times \vec{d}_{v/d} + v\vec{t} + \dot{\vec{d}}_{v/d} \right). \qquad (14)$$

Similarly, the rotational velocity of $C\xi\eta\zeta$ is the sum of two terms: (i) the rotational velocity $vR_{v/d}^T\vec{w}_d$ of the *Otvs* frame, and (ii) the rotational velocity $\vec{w}_{v/d}$, 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 $v\vec{w}_d + R_{v/d}\vec{w}_{v/d}$; the corresponding expression in the $C\xi\eta\zeta$ frame is

$$\vec{w}_v = vR_{v/d}^T\vec{w}_d + \vec{w}_{v/d}. \qquad (15)$$

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

Departures of Ground Vehicle Motion from Smoothness

The motion of a ground vehicle depends on many factors: the type of intended motion; the speed of the vehicle; the skill of the driver; the size, height and weight of the vehicle; the type and size of the wheels (or tractor treads), and the nature of the suspension mechanism, if any; and the nature of the surface on which the vehicle is being driven. These factors tend to remain constant; they undergo abrupt changes only occasionally, e.g. if a tyre blows out, or the vehicle suddenly brakes or swerves to avoid an obstacle, or the type of surface changes. Such events may produce impulsive changes in the vehicle's motion, which will often have longer-lasting effects (though these effects will eventually be damped out). In addition, to these occasional events, 'steady-state' non-smoothness of a ground vehicle's motion may result from roughness of the surface. (For other types of vehicles, non-smoothness may also result from variations in the medium through which the vehicle moves.)

A ground vehicle drives over roads or surfaces (for brevity: *RS*s) that have varying degrees of roughness. 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, and will deal only with stationary roughness.

We assume that we are dealing with a well-balanced four-wheeled vehicle moving on an approximately 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.) We assume that the body of the vehicle is rigid and that the wheels are attached to it by suspension elements at the four corners of its base.

Under these assumptions, using simple geometric arguments, it is not hard to show [10] that the effect of a single bump on a wheel is impulsive; that the translational effects have small amplitude; and that the rotational effects are much smaller around the yaw axis of the vehicle than around the roll and pitch axes.

As the vehicle moves over rough terrain, each wheel hits bumps repeatedly. The suspension integrates and damps the impulsive effects of the bumps. Each suspension element can be modeled by a spring with damping; its characteristic function is a sine function multiplied by an exponential damping function (see [17]).We assume that the suspension elements associated with the four wheels are independent of each other and are parallel to the vertical axis of the vehicle. (A discussion of the dynamics of the suspension of a ground vehicle on rough terrain can be found in Yao and Chellappa [21].)

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's base up and down. In real vehicles, the period of oscillation is typically on the order of 0.5 s and the maximum vertical displacement of the suspension elements is typically 0.025m. In general, it takes several periods to damp out the spring. The maximum velocity of the oscillation is typically in the order of 0.1m/s.

The suspension elements integrate and damp the short duration effects of the individual bumps, resulting in a set of out-of-phase oscillatory motions. The translational effects of the bumps are proportional to the velocities (or displacements) of the suspension elements and the dimensions of the vehicle and are quite small. It was found in Duric and Rosenfeld [10] that for a 'typical' vehicle, whose base has width of 1m and length of 2m, the translational bounce velocity is $< 1m/s$, the yaw velocity is $\mathcal{O}$ (0.01) rad/s, the roll velocity is < 0.2 rad/s, and the pitch velocity is < 0.1 rad/s. Also, the maximum roll angle of the base of the vehicle is < 0.05 rad and the maximum pitch angle is < 0.025 rad; the maximum yaw angle is negligible.

We 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 – to the sizes of the velocity components which are due to departures of the vehicle frame from the Darboux frame.

The translational velocity of the Darboux frame is just $v\vec{t}$; thus the magnitude of the translational velocity is just v. If $v = 30mi/h$ ($\approx 13\cdot41$ m/s $\approx 48km/h$) 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 in the order of 0.1m/s or less.

The rotational velocity of the Darboux frame is $v\vec{w}_d = v$ ($\tau_g\vec{t} + \kappa_n\vec{v} + \kappa_g\vec{s}$); thus the magnitude of the rotational velocity is $v\sqrt{\tau_g^2 + \kappa_n^2 + \kappa_g^2}$. We can estimate bounds on τ_g, κ_n and κ_g by examining the highway design recommendations published by the American Association of State Highway Officials [3]. An analysis of these recommendations (see Duric and Rosenfeld [10] for details) leads to the following conclusions about the impulsive and smooth translational and rotational velocity components of the vehicle, for realistic vehicle speeds. 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 velocity 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 smaller than (or comparable to) the smooth yaw velocity.

The highway recommendations [3] also have important implications about the nature of the smooth motion of a ground vehicle. Except during transitions, if a vehicle is being driven normally on a well-designed highway, its smooth rotational and translational components are approximately constant. (The translational components are constant in the vehicle coordinate frame even when the

vehicle is turning.) It is also observed that a driver on a poorly designed road attempts to drive as though the road were well designed, e.g. to make turns at constant angular velocities, and to follow spiral arcs in transitioning between turns, in order to reduce undesirable acceleration effects on the vehicle. We will assume in the next section that an off-road driver also attempts to drive in this way – i.e., that the ideal (smooth) motion of the vehicle has piecewise constant translational and rotational velocity components.

Camera Motion and Image Sequence Smoothing

In this section we analyse the properties of images and image motion fields obtained by a camera mounted rigidly on a moving ground vehicle. Based on our analysis, we design a smoothing algorithm which we use to smooth ('stabilize') several image sequences obtained by a camera mounted on a real moving vehicle.

During the course of this section we derive the relative motion equations for a camera rigidly mounted on a moving ground vehicle and show that we only need to smooth the rotational components of the motion. Evidently, the rotational components are dominant at image points whose corresponding scene points are distant. We present methods of classifying image points with respect to their distance, together with 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 the previous section. Finally, we describe an algorithm for the smoothing (stabilization) of image sequences which uses only the preceding and current images (so that it can operate in real time), and we present the results of applying this algorithm to an image sequence obtained by a vehicle moving over bumpy terrain.

Camera Motion

Assume that a camera is mounted on the vehicle; let $\vec{d}_c$ be the position vector of the nodal point of the camera relative to the mass center of the vehicle. The orientation of the camera relative to the vehicle coordinate system $C\xi\eta\zeta$ is given by an orthogonal rotational matrix (a matrix of the 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 vehicle coordinate system. We will assume that the position and orientation of the camera relative to the vehicle coordinate system do not change as the vehicle moves. (The cases of a vehicle-mounted camera that is non-

rigidly mounted or that can move relative to the vehicle – for example, a camera that is able to track an object in the scene – 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 vehicle 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 (2):

$$\dot{\vec{r}}_e = -\vec{w} \times \vec{r}_e - \vec{T}.$$

In this expression, the rotational velocity is $\vec{w} = R_c^T \vec{w}_v$, and the translational velocity is $\vec{T} = R_c^T (\vec{T}_v + \vec{w}_v \times \vec{d}_c)$, where $\vec{w}_v$ and $\vec{T}_v$ are the rotational and translational velocities of the vehicle coordinate system. Earlier, we saw (14–15) that these velocities can be expressed in terms of the rotational and translational velocities, $v\vec{w}_d$ and $v\vec{t}$, of the Darboux frame and the rotational and translational velocities, $\vec{w}_{v/d}$ and $\vec{d}_{v/d}$ of the vehicle frame relative to the Darboux frame. We thus have

$$\vec{w} = R_c^T \left(v R_{v/d}^T \vec{w}_d + \vec{w}_{v/d} \right), \tag{16}$$

$$\vec{T} = R_c^T R_{v/d}^T \left(v\vec{w}_d \times \vec{d}_{v/d} + v\vec{t} + \dot{\vec{d}}_{v/d} \right) + \vec{w} \times \left(R_c^T \vec{d}_c \right) \tag{17}$$

We shall now show that (up to a term of $\mathcal{O}(v\delta\|\vec{w}_d\|)$) we can replace the matrix $R_{v/d}^T$ in (16) by I. By (5) we have $R_{v/d}^T = I - .\delta C_{v/d} + \mathcal{O}(\delta^2)$, where δ is the angle of rotation and $C_{v/d}$ is the skew matrix that corresponds to the direction $\vec{c}_{v/d}$ of the rotational axis. From (15) we thus have

$$\vec{w}_v = v R_{v/d}^T \vec{w}_d + \vec{w}_{v/d}$$
$$= v\vec{w}_d - \delta C_{v/d}\vec{w}_d + v\mathcal{O}(\delta^2)\vec{w}_d + \vec{w}_{v/d}$$
$$= v\vec{w}_d - \delta v\vec{c}_{v/d} \times \vec{w}_d + \vec{w}_{v/d} + \mathcal{O}(v\delta^2\|\vec{w}_d\|) \tag{18}$$

where the last term on the r.h.s. is a vector whose elements are $\mathcal{O}(v\delta^2\|\vec{w}_d\|)$. The significant terms on the r.h.s. of (18) are $v\vec{w}_d$, which is the smooth velocity of the vehicle; $\vec{w}_{v/d}$, which is the non-smooth velocity of the vehicle; and the cross-product term $\delta v\vec{c}_{v/d} \times \vec{w}_d$, which is also non-smooth because of the non-smoothness of δ and $\vec{c}_{v/d}$. However, since $\delta = \mathcal{O}(0.05)$ rad and $\|\vec{c}_{v/d}\| = 1$, the cross product term is small compared to $v\vec{w}_d$; indeed, it is $\mathcal{O}(v\delta\|\vec{w}_d\|)$. Therefore, $\vec{w}_v$ is approximately the sum of the smooth rotational velocity, $v\vec{w}_d$, and the non-smooth rotational velocity, $\vec{w}_{v/d}$. Finally, since R_c is constant the smooth part of $\vec{w}$

is $v R_c^T \vec{w}_d$ and the non-smooth part of $\vec{w}$ is approximately $R_c^T \vec{w}_{v/d}$, and we have

$$\vec{w} = v R_c^T \vec{w}_d + R_c^T \vec{w}_{v/d} + \mathcal{O}\left(v\delta \|\vec{w}_d\|\right). \qquad (19)$$

We have seen that both $\|v\vec{w}_d\|$ and $\|\vec{w}_{v/d}\|$ are $\mathcal{O}(0.1)$ rad/s. The factors R_c^T and $R_{v/d}^T$ do not affect the magnitude of either $\vec{w}$ or $\vec{T}$. Thus, the terms on the r.h.s. of (16) or (19) have comparable magnitudes. Since $\vec{w}_d$ is the (smooth) rotational velocity of the Darboux frame, and $\vec{w}_{v/d}$ is the (impulsive) rotational velocity of the vehicle relative to this frame, it is clearly important to smooth the effects of the rotational components on the image sequence.

As regards the translational components, note that for normal speeds of the vehicle ($v > 10$m/s ≈ 22mi/h), typical suspension elements, and the camera mounted on the vehicle close to the center of mass we have $\|\vec{d}_{v/d}\| = \mathcal{O}(0.025)$m/s, $\|\dot{\vec{d}}_{v/d}\| = \mathcal{O}(0.1)$m/s, and $\|\dot{\vec{d}}_c\| = \mathcal{O}(1)$m/s. The magnitudes of the terms on the r.h.s of (17) are thus $\|v\vec{w}_d \times \vec{d}_{v/d}\| \leq v \|\vec{w}_d\| \|\vec{d}_{v/d}\| = \mathcal{O}(0.0025)$m/s; $\|v\vec{t}\| = v = \mathcal{O}(10)$m/s; and $\|\vec{w} \times (R_c \vec{d}_c)\| \leq \|\vec{w}\| |\vec{d}_c\| = \mathcal{O}(0.1)$m/s. 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 $\vec{T}$. Smoothing the effects of the translational components on the image sequence is thus unimportant, since $v\vec{t}$ is already smooth.

Identification of Distant Image Points

Typical images obtained by a camera mounted on a ground vehicle are not arbitrary. In general, the camera can be pointing in any direction relative to the vehicle, but its orientation relative to the vehicle (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 vehicle. In other words, if the vehicle is on level ground, the camera is pointing approximately horizontally. Thus, unless the vehicle is at the bottom of a deep hollow, or is screened by natural or artificial objects, the camera can see objects which are very close to the vehicle as well as objects which are very distant, as shown in Figure 3. When only nearby objects can be seen, and $\angle(\vec{T}, \vec{r})$ is significantly different from zero, translation dominates and thus 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 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 identifying image points that represent distant scene points.

If a range sensor is available in addition to the video camera, and the video and range images can be registered, image points can be classified based on their ranges. A stereo rig is another possibility for range estimation; to do the estimation in real time, a fast multiple-camera dynamic programming based stereo algorithm could be used [14]. 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. Ordinarily, however, multiple cameras mounted on a vehicle would not have overlapping FOVs, so that a stereo rig would ordinarily not be available. 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 vehicle 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 [2,8]. 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 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 vehicle is driving 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 axis is approximately horizontal), and estimating the boundaries of these parts. The orientations, positions, and strengths of
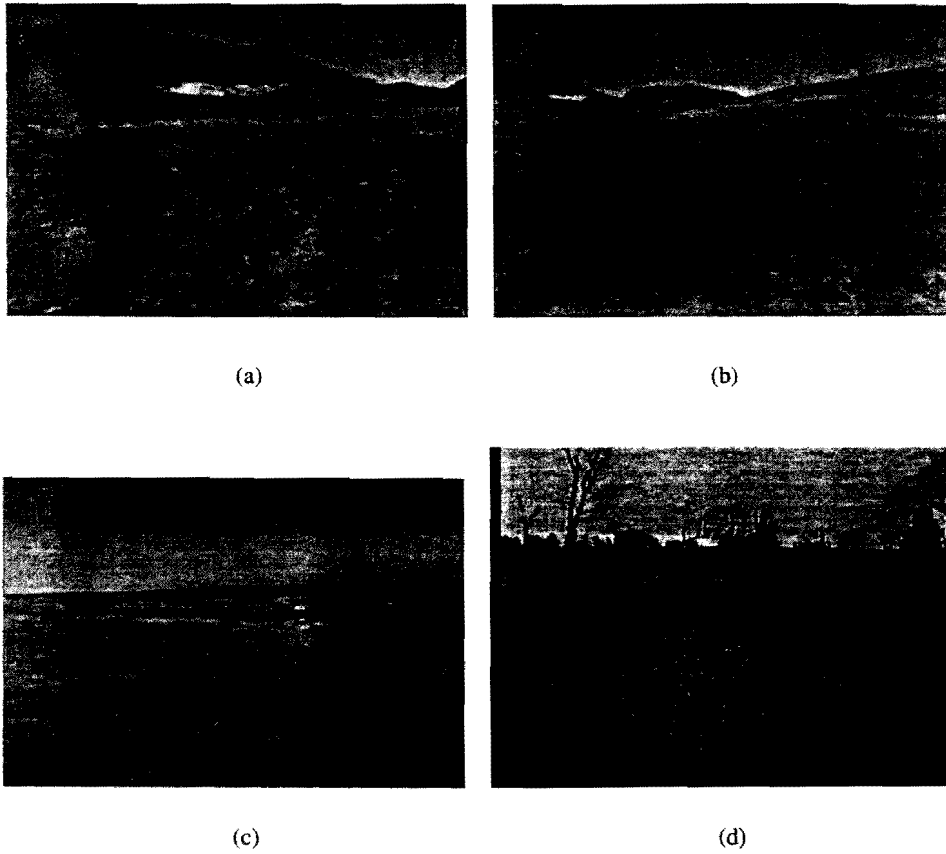
(a)

(b)

(c)

(d)

Figure 3. Typical scenes obtained by a horizontally pointing camera carried by a ground vehicle.

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. Figure 4 shows horizon detection results for the images in Figure 3.

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 $v \approx 10$m/s and frame time $\Delta t = \mathcal{O}$ (1/30)s 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. Also, the magnitudes and the 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.

Estimation of Rotation

We now describe an algorithm for using distant points to estimate rotation. We shall use the following notation: let $\vec{n}_r = n_x \vec{i} + n_y \vec{j} = \nabla I / \|\nabla I\|$ be the normal direction at $\vec{r}$. The normal motion field at $\vec{r}$ is given by $\vec{r}_n = (\vec{r} \cdot \vec{n}_r)\vec{n}_r$, so that from (8, 9) we have

$$\dot{\vec{r}}_n \cdot \vec{n}_r = -\frac{1}{Z}\left[n_x\left(-t_x f + xt_z\right) + n_y\left(-t_y f + yt_z\right)\right]$$

$$+ \left[n_x \frac{xy}{f} + n_y\left(\frac{y^2}{f} + f\right)\right] w_x$$

$$- \left[n_x\left(\frac{x^2}{f} + f\right) + n_y \frac{xy}{f}\right] w_y + \left(n_x y - n_y x\right)w_z. \quad (20)$$

The first term on the r.h.s. of (20) is the translational normal motion and the other three terms are the rotational normal motion at $\vec{r}$.

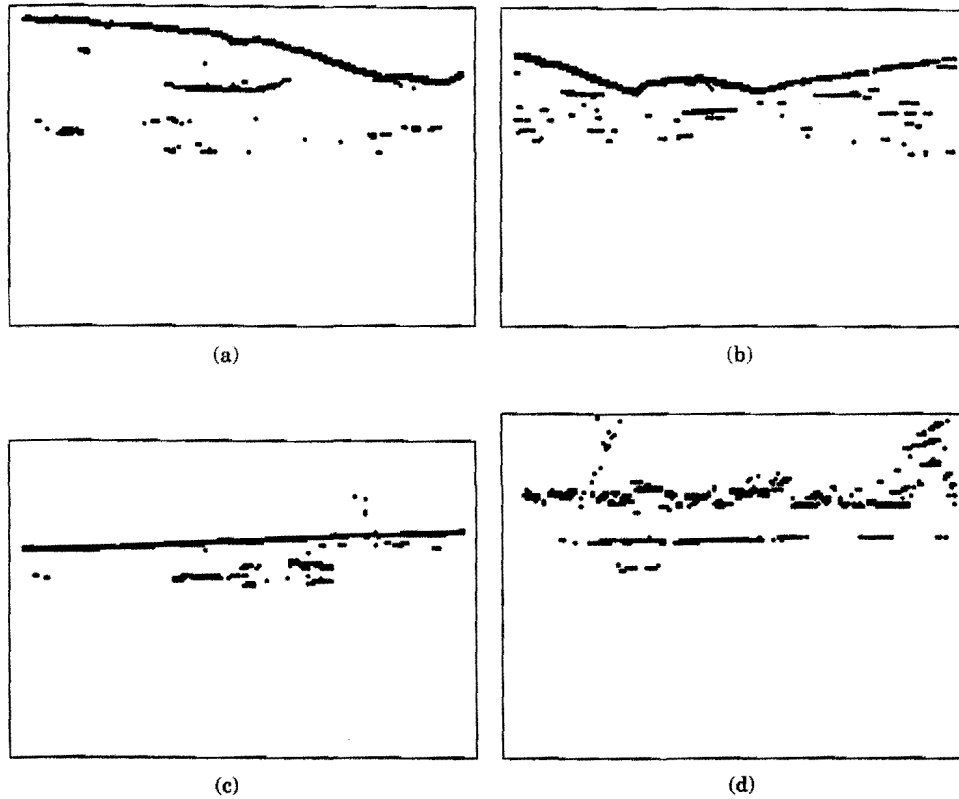Consider a set of image points on the horizon (or more generally, a set of image points which seem to be at large

Figure 4. The results of horizon detection for the images in Figure 3.

distances); these points are in fact usually distant and rotation is dominant. For such points, it will usually be the case that the rotational normal motion dominates the translational normal motion. The last three terms of (20) can be written as $\vec{a} \cdot \vec{w}$, where $\vec{a}$ is computable from images. From (20) we thus have

$$\dot{\vec{r}}_n \cdot \vec{n}_r = \epsilon_t + \vec{a} \cdot \vec{w} \qquad (21)$$

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

From (10), the normal flow at $\vec{r}$ is $\vec{u}_n \cdot \vec{n}_r = -I_t / \|\nabla I\|$. Also, from Verri and Poggio [19] we know that the difference between the normal flow and the normal motion is inversely proportional to the gradient magnitude; we can thus write

$$\dot{\vec{r}}_n \cdot \vec{n}_r = \vec{u}_n \cdot \vec{n}_r + \mathcal{O}\left(\|\nabla I\|^{-1}\right) = -\frac{I_t}{\|\nabla I\|} + \mathcal{O}\left(\|\nabla I\|^{-1}\right). \qquad (22)$$

From (21) and (22) we can write

$$\vec{a} \cdot \vec{w} + \frac{I_t}{\|\nabla I\|} = \epsilon \qquad (22)$$

where $\epsilon = \epsilon_t + \mathcal{O}(\|\nabla I\|^{-1})$ is the error.

For each horizon point $\vec{r}_i$ we have one equation (23). Let the number of horizon points be $N \geq 3$. We then have a system

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

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 $\vec{a}$, and E is an N-element array with elements ϵ_i. We seek $\vec{w}$ that minimizes $\|E\| = \|\mathbf{y} - A\vec{w}\|$; the solution satisfies the system [18]

$$A^T A \vec{w} = A^T \mathbf{y} = \mathbf{d}. \qquad (24)$$

We solve this system using the Cholesky decomposition [18]. (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 $L^T \vec{w} = \mathbf{e}$ for $\vec{w} = (w_x\ w_y\ w_z)^T$ in the camera coordinate system.)

The computed $\vec{w}$ may be inaccurate due to various geometrical and numerical factors, to be discussed below.

However, it is possible to improve on the computed solution of the system (24) iteratively. Given the estimate $\vec{w}$ 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 $\vec{w}$, and using (6) we define the rotational matrix $R = I + \Delta t \Omega$ (where Δt is the time interval between frames). We then apply the rotation R to the frame (de-rotation/warping); as a result the rotational velocity of the image sequence at that frame will be reduced by $\vec{w}$. After de-rotating the frame we compute the residual rotational velocity $\Delta \vec{w}$ from the de-rotated frame and the (uncorrected) predecessor frame. We then replace $\vec{w}$ by $\vec{w} + \Delta \vec{w}$ and proceed to again create Ω and R, and use R to de-rotate the original frame again. We repeat this process until $\|\Delta \vec{w}\| < \epsilon$. At each step we do the de-rotation using the new $\vec{w} + \Delta \vec{w}$, and we apply it to the original image (rather than using the new $\Delta \vec{w}$ on the already de-rotated image) to avoid an 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 $\vec{w}$ is $\approx 0.5\|\vec{w}\|$ – i.e., the error is halved 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 (7) we have

$$x' = f\frac{X'}{Z'} = f\frac{r_{11}X + r_{12}Y + r_{13}Z}{r_{31}X + r_{32}Y + r_{33}Z} \cdot \frac{f/Z}{f/Z}$$
$$= f\frac{r_{11}x + r_{12}y + r_{13}f}{r_{31}x + r_{32}y + r_{33}f} \qquad (25)$$

and similarly

$$y' = f\frac{Y'}{Z'} = f\frac{r_{21}x + r_{22}y + r_{23}f}{r_{31}x + r_{32}y + r_{33}f}. \qquad (26)$$

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 $\vec{w}$ 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 [1,9]. 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. $\vec{n} \cdot \vec{r}_w = 0$. 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 $\vec{w}$ 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 $(\vec{r}_c \cdot \vec{w})\vec{r}_c$ cannot be detected. We say in this situation that the normal flow resulting from the rotational velocity $(\vec{r}_c \cdot \vec{w})\vec{r}_c$ belongs to the null space of the feature matrix A [18]. Fortunately, if such $\vec{w}$s exist, the positive definite (semidefinite) matrix $A^T A$ must have a large condition number (the ratio of 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 $\vec{w}$ 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 $\vec{w}$ 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 (24) is reliable and that a few iterations of our algorithm will be enough to obtain a reliable estimate of $\vec{w}$. In all the examples shown in Figure 3 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 30th frame of the sequence in Figure 3(a) Figure 5 shows the normal flow and Figures 6 and 7 respectively show the estimated rotational and the residual translational flow after four iterations. The condition number for the feature matrix $A^T A$ was ≈ 28; it remained close to this value through the entire sequence. The rotational velocity was estimated in four iterations. After the first iteration the estimated $\vec{w} = (w_x\ w_y\ w_z)^T$ (in camera coordinates) was $(0.00014\ 0.00019\ -0.00066)^T$ rad/frame, and after four iterations it was $(0.00026\ 0.00037\ -0.00125)^T$ rad/frame, so that $\|\Delta \vec{w}\| = 0.0001$ rad/frame.

Figure 5. Computed normal flow in the 30th frame of the sequence containing Figure 3(a).

Figure 6. Estimated rotational flow for the same frame.

Figure 7. The residual (translational) flow for the frame after subtracting the rotational flow.

Smoothing Algorithm

We saw that we need only smooth the effects (on the image sequence) of the rotational part of the motion. Earlier, we showed how the rotational velocity, $\vec{w}$, of the camera can be estimated at each frame. We cannot simply 'smooth' the image sequence by using these estimates to de-rotate 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 vehicle may make turns, go over hills, etc.), this would cause large parts of the images to be lost.

We saw that the non-smooth part of the rotational velocity of the vehicle has larger magnitude than the smooth part. However, we also saw that the cumulative effects of the non-smooth part on the orientation of the vehicle (the angles between the vehicle 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 $\vec{w}_f$ of $\vec{w}$. We will then use the residual rotational velocity $\vec{w} - \vec{w}_f$, which corresponds to the non smooth part of $\vec{w}$, to construct a sequence of rotational matrices; we can then use these matrices to correct the corresponding frames as in (25, 26). The matrices can be constructed as follows: when the integral of the residual velocity, $\vec{w}_r = \int_0^t \vec{w} - \vec{w}_f \, dt$ 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 $\vec{w}_f$. The first order approximation of the rotational matrix R [see (5)] defined by $\vec{w}_r$ is then

$$R = \begin{pmatrix} 1 & -W_3 & W_2 \\ W_3 & 1 & -W_1 \\ -W_2 & W_1 & 1 \end{pmatrix} + \mathcal{O}\left(\|\vec{w}_r\|^2\right). \quad (27)$$

To use this matrix to correct the images as in (25, 26), we need only transform it to a rotation matrix in the camera frame. For example, if the given frame is the vehicle frame the matrix in the camera frame is $R_c^T R R_c$. We will in fact use the components in the vehicle frame (whose axes are the roll, pitch, and yaw axes) in this section.

A vehicle driven on a well-designed road (or a well-driven vehicle on any terrain) may undergo smooth rotation when it turns or goes over a hill. A smooth turn involves 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 non-zero 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 $\vec{w}$ 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 8 shows the (unsmoothed) roll, pitch, and yaw components of $\vec{w}$ that were estimated from 100 successive frames (a 3-s subsequence) of the image sequence that began with Figure 3(a). We see that all three components fluctuate strongly, with major periods of oscillation of the order of 20–30 frames (i.e. 1s or less). The components also remain relatively small in amplitude, usually less than 2×10^{-3} rad/frame; the fluctuations in the roll component have the highest amplitude, as predicted earlier. Note that the components seem to fluctuate around approximately constant values – in fact, around values that are close to zero, indicating that this subsequence does not include a turn or a hill crest.

When we 'fit' piecewise constant functions, $\vec{w}_f$, of small magnitude to the rotational velocity components $\vec{w}$, it is
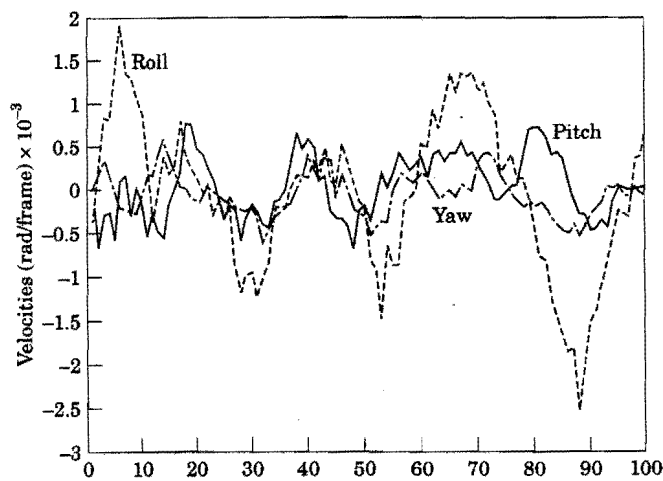
important that we (try to) keep the integrals of the residual velocities $W_r = \int_0^t w - w_f \, dt$ small, so that our method of correcting the frames, described earlier in this section, can be used. In the next paragraph we will describe an algorithm for doing the 'fitting'. Our algorithm assumes that the smoothed component w_f is zero at $t = 0$, and repeatedly 'corrects' its estimate of w_f, based on the behavior of W_r, in such a way as to keep it (almost always) bounded. Thus the W_rs remain small, and so can be used to correct the frames.

In the algorithm, initially we set $w_f = 0$. We do not change w_f as long as $|W_r|$ is smaller than some preassigned bound h. [Typical values of h are 0.006 ($\approx (1/3)°$), 0.003, and 0.002 rad for the roll, pitch, and yaw velocity components, respectively.] When $|W_r|$ reaches or exceeds h, say at time t_0, we add to w_f the amount $0.25 \, W_r/t_0$ (1/4 of the average slope of W_r during the time interval $[0, t_0]$; see Figure 9). This value of w_f is maintained unless $|W_r|$ further increases to $3h/2$, say at time t_1; if so, we again add to w_f the amount $0.25 \, [W_r(t_1) - W_r(t_0)]/(t_1 - t_0)$ (again, 1/4 of the average slope of W_r since the last change in w_f; see Figure 9). This is repeated if $|W_r|$ again grows by $h/2$. If $|W_r|$ exceeds $2h$, we multiply w_f by 1.1, repeatedly if necessary (at each new frame), until $|W_r|$ drops below $2h$. Similarly, if $|W_r|$ drops below h, we multiply w_f by 0.9, repeatedly if necessary, as long as $|W_r|$ is decreasing. Finally, if $|W_r|$ drops below $h/2$, we reset w_f to zero. The approximations of the velocities in Figure 8 by this algorithm are shown in Figure 10. [Note that they include intervals over which w_f is not constant; these are the intervals during which the multiplicative adjustment process was operating.]

We have successfully applied the algorithm to several image sequences obtained by a camera on a ground vehicle moving across bumpy terrain; individual frames from some of these sequences were shown in Figure 3. The stabilization results cannot be shown here, but digital video versions of the sequences before and after stabilization are available from the authors.
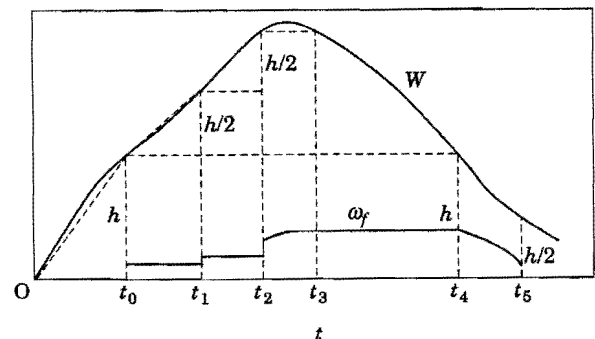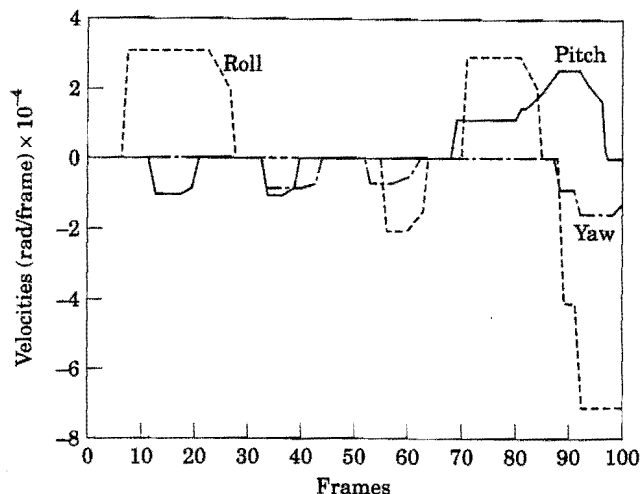

Figure 8. Rotational velocity components for the sequence containing Figure 3(a).

Figure 9. Piecewise constant approximation of w by w_f.

Figure 10. Estimates obtained by the algorithm from the components in Figure 8.

Conclusions

The motion of a vehicle can usually be regarded as consisting of a desired smooth motion that includes impulsive or high-frequency components. If the vehicle is carrying a camera, the non-smooth motion will perturb the sequence of images obtained by the camera. The goal of image sequence stabilization is to correct the images so that they are approximately the same as the images that would have been obtained if the motion of the vehicle had been smooth.

We have analysed the smooth and non-smooth motions of a ground vehicle and have shown that only the rotational components of the non-smooth motion have significant perturbing effects on the images. We have analysed the relationship between the shape of the terrain along which the vehicle travels and the parameters of the smooth motion of the vehicle. We have used this analysis, together with highway design recommendations [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 vehicle's rotation. Finally, we have described an algorithm that fits smooth (ideally, piecewise constant) rotational motions to these estimates and we have shown how the residual rotational motion (the difference between the estimated actual motion and the fitted smooth 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, which are available on request, demonstrate the effectiveness of our approach.

References

1. Aloimonos, Y. & Duric, Z. (1994) Estimating the heading direction using normal flow. *Int. J. Computer Vision* 13: 33–56.
2. Ancona, N. & Poggio, T. (1993) Optical flow from 1D correlation: application to a simple time-to-crash detector. In *Proc. DARPA Image Understanding Workshop*, pp. 673–682.
3. American Association of State Highway Officials (1977) *A Policy on Geometric Design of Rural Highways* (9th edition). AASHO.
4. Boas, M. L. (1983) *Mathematical Methods in the Physical Sciences* (2nd edn). New York: Wiley.
5. Bottema, O. & Roth, B. (1979) *Theoretical Kinematics*. Amsterdam: North-Holland.
6. Burt, P. J. & Anandan, P. (1994) Image stabilization by registration to a reference mosaic. In *Proc. ARPA Image Understanding Workshop*, pp. 425–434.
7. Davis, L. S., Bajcsy, R., Nelson, R. & Herman, M. (1994) RSTA on the move. In *Proc. ARPA Image Understanding Workshop*, pp. 435–456.
8. Duric, Z., Rosenfeld, A. & Duncan, J. (1994) The applicability of Green's theorem to computation of rate of approach. In *Proc. ARPA Image Understanding Workshop*, pp. 1209–1217.
9. Duric, Z., Rosenfeld, A. & Davis, L. S. (1995) Egomotion analysis based on the Frenet-Serret motion model. *Int. J. Computer Vision*, 15: 105–122.
10. Duric, Z. & Rosenfeld, A. (1995) Stabilization of Image Sequences. Technical Report CAR-TR-778, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park.
11. Hansen, M., Anandan, P., Dana, K., van der Wal, G. & Burt, P. J. (1994) Real-time scene stabilization and mosaic construction. In *Proc. ARPA Image Understanding Workshop*, pp. 457–465.
12. Horn, B. K. P. & Schunck, B. G. (1981) Determining optical flow. *Artificial Intelligence*, 17: 189–203.
13. Irani, M., Rousso, B. & Peleg, S. (1994) Recovery of egomotion using image stabilization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 454–460.
14. Kanade, T. (1994) Development of a video-rate stereo machine. In *Proc. ARPA Image Understanding Workshop*, pp. 549–557.
15. Koenderink, J. J. (1990) *Solid Shape*. Cambridge, MA: MIT Press.
16. Kreyszig, E. (1959) *Differential Geometry*. Toronto: University of Toronto Press.
17. Oldenburger, R. (1950) *Mathematical Engineering Analysis*. New York: Macmillan.
18. Stewart, G. W. (1973) *Introduction to Matrix Computations*. New York: Academic Press.
19. Verri, A. & Poggio, T. (1987) Against quantitative optical flow. In *Proc. International Conference on Computer Vision*, pp. 171–180.
20. Vieville, T., Clergue, E. & Facao, P. E. D. S. (1993) Computation of ego-motion and structure from visual and internal sensors using the vertical cue. In *Proc. International Conference on Computer Vision*, pp. 591–598.
21. Yao, Y. S. & Chellappa, R. (1995) Model-based vehicular motion and structure estimation. In *Proc. of International Conference on Robotics and Automation*.