Time to Collision Warning Chip:

Background:

Under certain circumstances, it is possible to estimate the time to collision (TTC) from a time varying image. It would be useful to encapsulate an algorithm for doing this into a cheap camera, ideally with computation done on the chip that senses the image. That way, there is no need to move large volumes of image data from an image sensor to a processing chip.

Such a chip would perform an extreme bandwidth compression: it has high bandwidth in (image sequence) and low bandwidth out (time to collision). Such a device could be outfitted with a cheap plastic lens and used as a warning system, aimed, for example, out the rear of car into each of the two 'blind spots' that are not easily visible in the driver's mirrors.

The key to recovering the time to collision is the realization that there are constraints between the brightness gradient (spatial derivatives of brightness) and the time derivative of brightness at a point in the image. These depend on motion field, and the motion field in turn depends on the rigid body motion between the camera and the object(s) being viewed.

Focus of Expansion:

A related chip is the 'focus of expansion' (FOE) chip built by Ig McQuirk. This is an analog VLSI chip that finds the focus of expansion. His chip uses spatial and time derivatives of brightness to locate the point in the image towards which motion is taking place. Again, this is a high band width in, low band width out application.

The difference between the two projects (FOE versus TTC) is that in the FOE chip, the intent is to be *insensitive* to the magnitude of the velocity, and to distances to points in the scene, while in the TTC chip the opposite is the case: the intent is to be insensitive to the direction of motion instead, which is exactly what the FOE chip recovers. Note that the time to collision is just the ratio of velocity to distance.

Interestingly, it is not possible to determine either absolute velocity or absolute distance from an image sequence, while the *ratio* of distance to velocity *can* be determined. Absolute distance or velocity cannot be determined because of the 'scale factor ambiguity': if we scale distances in the scene by some constant factor k, while also scaling the velocities by the same factor k, then we get the same image sequence.

In some sense the two chips (FOE and TTC) serve complementary functions: one finds the direction of motion (FOE) while the other finds the time until we ‘get there.’ While the FOE chips exploits ‘stationary points’ — points where the time derivative of brightness is zero while the spatial derivatives are not — the TTC chip cannot make use of information at stationary points, since this is insensitive to the velocity.

Method:

The basis of both FOE and TCC algorithms is the ‘constant brightness assumption’:

$$\frac{dE(x,y,t)}{dt} = 0$$

which is based on the observation that in many situations the brightness of a point does not change as it moves in the image. The above total derivative can be expanded into:

$$uE_x + vE_y + E_t = 0$$

where $u = dx/dt$, $v = dy/dt$ are the x and y components of the motion field in the image, while E_x, E_y, E_t are the x, y, and t derivatives of brightness. For convenience this can also be written

$$E_{\mathbf{r}}.\mathbf{r}' + E_t = 0$$

where $E_{\mathbf{r}} = (E_x, E_y, 0)^T$ while $r' = (u, v, 0)^T$.

The constant brightness assumption naturally does not apply in all cases. It is violated by a specular surfaces, since typically such a surface will have different brightnesses when viewed from different directions. It is also violated if the light sources move relative to the scene. But in many practical situations it is a useful and reasonable assumption.

Motion Field:

The image velocity (u, v) is a projection of the velocity of the corresponding point in the world in front of the camera. The image point and the scene point are connected by a ray going through the center of projection. The perspective projection equation is simply

$$\frac{1}{f}\mathbf{r} = \frac{1}{\mathbf{R}\cdot\hat{\mathbf{z}}}\mathbf{R}$$

where $\mathbf{r} = (x, y, f)^T$ is a vector to a point in the image, while $\mathbf{R} = (X, Y, Z)^T$ is the vector to the corresponding point in the scene. Here $\mathbf{R}\cdot\hat{\mathbf{z}} = Z$, with $\hat{\mathbf{z}}$ the unit vector along the optical axis, and f the principal distance of the camera (usually just slightly larger than the focal length).

To obtain the motion field (a vector field in the image plane) we differentiate the perspective projection equation with respect to time.

$$\frac{1}{f}\mathbf{r}' = \frac{\mathbf{R}'}{\mathbf{R}\cdot\hat{\mathbf{z}}} - \frac{\mathbf{R}'\cdot\hat{\mathbf{z}}}{(\mathbf{R}\cdot\hat{\mathbf{z}})^2}\mathbf{R}$$

where $\mathbf{r}' = d\mathbf{r}/dt$ and $\mathbf{R}' = d\mathbf{R}/dt$. This equation relates velocities of scene points to corresponding velocities projected into the image plane. This can also be re-written using some vector identities and the perspective projection equation:

$$\frac{1}{f}\mathbf{r}' = \frac{(\mathbf{r} \times \mathbf{R}') \times \hat{\mathbf{z}}}{\mathbf{R} \cdot \hat{\mathbf{z}}}$$

where × denotes the cross product and · denotes dot-product.

Brightness and Motion Constraint:

The image velocity $\mathbf{r}'$ is $(u, v, 0)^T$ while the velocity in the scene is $\mathbf{R}'$. If we insert the above equation for $\mathbf{r}'$ in the brightness change constraint equation we get — after some manipulation:

$$E_t = \frac{\mathbf{R}' \cdot \mathbf{s}}{\mathbf{R} \cdot \hat{\mathbf{z}}}$$

where $\mathbf{s} = \mathbf{r} \times (E_{\mathbf{r}} \times \hat{\mathbf{z}})$. The vector $\mathbf{s}$ can also be written in terms of its components

$$\mathbf{s} = (fE_x, fE_y, -(xE_x + yE_y))^T$$

The above equation — which applies at every image point — is a constraint relating measurable quantities such as spatial and time derivatives of brightness to unknowns such as velocities and distances in the scene.

In the case of rigid body motion

$$\mathbf{R}' = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{R}$$

where $\mathbf{t} = (U, V, W)^T$ is the instantaneous translational velocity of the camera, while $\boldsymbol{\omega} = (A, B, C)^T$ is the instantaneous rotational velocity.

Since the above equation applies at every picture cell, there potentially is a lot of constraint to work with. If the only unknowns are the parameters of rigid body motion (3 parameters for translation and 3 for rotation) then we have highly over-determined system and a stable solution can be achieved even in the presence of a lot of noise in the measurements.

However, in an arbitrary scene, the 'depth' $Z = \mathbf{R} \cdot \hat{\mathbf{z}}$, is not known and this can vary from place to place in the image. The total number of unknowns increases dramatically if we add unknown depths at every picture cell. This can make the problem not so well posed (i.e. number of constraints is not greater than the number of unknowns).

This is why we expect to have to work instead with a reduced model. The simplest case to investigate is a constant depth model, the next easiest a planar surface approximation, and then a quadratic surface.

These simple models may prove adequate if we divide the image into patches small enough so that the surface images in any given patch is approximately at constant depth, or approximately planar. We have to be careful, however, not to make the patches too small, since the component of motion along the optical axies becomes less well determined with a narrow field of view. Also, while the larger the patches, the more constraint it provides, a large patch is more likely to contain image components from more than one object.

Ideally, the division into patches should be adaptive. If there is a lot of textureal detail, a small patch can be used, while if brightness varies only slowly, a large patch may be needed to get satisfatory results.

Rigid Body Motion:

If the camera is translating with velocity $\mathbf{t}$ and rotating with angular velocity $\boldsymbol{\omega}$ with respect to an object in the scene, then

$$\mathbf{R}' = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{R}$$

where $\mathbf{t} = (U, V, W)^T$ and $\boldsymbol{\omega} = (A, B, C)^T$ say. Our task is to recover the components of motion and the depth.

For recovering the time to impact we are actually only concerned about

$$T = \frac{Z}{W} = \frac{\mathbf{R} \cdot \hat{\mathbf{z}}}{\mathbf{t} \cdot \hat{\mathbf{z}}}.$$

Plan:

We will first explore a number of algorithmic alternatives on existing work stations using both real image sequences created under controlled conditions as well as synthetic sequences where the motion and shapes of objects is known exactly.