Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Time to Contact Relative to a Planar Surface

Berthold K.P. Horn* Yajun Fang** Ichiro Masaki***

Abstract— **We show how to determine the time to contact from time varying images using only accumulated sums of suitable products of image brightness derivatives. There is no need for feature or object detection, tracking of features, estimation of optical flow, or any "higher level" processing. This so-called "direct" method for determining the time to contact is based on analysis of the motion field resulting from rigid body motion under perspective projection and the constant brightness assumption. The method has essentially no latency, since it can be based on analysis of just two frames of a video sequence, and does not require a calibrated camera. An implementation of the method is demonstrated on synthetic image sequences and stop motion sequences — where the ground truth is accurately know — as well as on video sequences taken by a camera mounted on moving vehicles.**

I. Background

The time to contact (TTC) is defined as the time that would elapse before the center of projection (COP) reaches the surface being viewed if the current relative motion between the camera and the surface were to continue without change. The TTC is essentially the ratio of distance to velocity:

$$T = -Z \Big/ \frac{dZ}{dt} = -1 \Big/ \frac{d}{dt} \log_e(Z), \tag{1}$$

where Z is the distance from the center of projection (COP) to the object, while $W = dZ/dt$ is the velocity at which the object is moving relative to the COP (which will be negative if the object is approaching the camera). While distance and velocity can not be recovered from images taken with a single camera without additional information, such as the principal distance and the size of the object, the *ratio* of distance to velocity *can* be recovered directly, even with an uncalibrated sensor.

Consider a simple situation where the camera is approaching an elongated planar object lying perpendicular to the optical axis, with the direction of translational motion along the optical axis. If the (linear) size of the object is S and the size of its image is s, then, from the perspective projection equation, we have $(s/f) = (S/Z)$, that is, $sZ = fS$. Differentiating w.r.t time yields

$$s\frac{dZ}{dt} + Z\frac{ds}{dt} = 0 \tag{2}$$

Together with (1), this shows that the TTC is equal to the ratio of the size s of the image of the object to the rate of change of the size, that is

$$T = s \Big/ \frac{ds}{dt} = 1 \Big/ \frac{d}{dt} log_e(s) \tag{3}$$

It is convenient to use the inter-frame interval as the unit of time and express the TTC as a multiple of that interval.

Naturally, formulae such as (3) beg the question of *how* one determines the size of the image of an object and the change in that size over time [12]. In order to use the formulae, one has to be able to extract features and track features from frame to frame. In addition, this idea does not easily generalize to translational motions that do not happen to be along the optical axis — or to objects other than planar ones that happen to lie at right angles to the optical axis.

Further, the time varying image is sampled at regular intervals and the time derivative of size is estimated using the difference between sizes of the images of the object in two frames. High accuracy is needed in measuring the size of the image in order to obtain accurate estimates of the TTC when it is large compared to the inter-frame interval. For example, when the time to collision is 100 frames, then an image of size 100 pixels changes by only 1 pixel from frame to frame, and so, to achieve even 10% error in the TTC one would have to measure the size of the image with an accuracy of better than $1/10$ of a pixel. The tolerance for measurement error becomes even smaller when the object is further away and the TTC larger.

An alternative approach to determining the TTC is to first estimate the optical flow (or the so-called normal flow) and then use the expected form of the flow field based on rigid body motion relative to a known shape to recover the TTC [2], [8], [9], [10], [11]. However, methods for estimating optical flow are iterative, need to work at multiple scales, tend to be computationally expensive and require a significant effort to implement properly.

II. Direct Method for Time to Contact

The method described here instead works directly with the derivatives of image brightness and does not require feature detecting, feature tracking, or estimation of the optical flow. The key to the new method is exploitation of the constraints between the brightness gradient (spatial derivatives of brightness) and the time derivative of brightness. These derivatives depend on the motion field, and the motion field in turn depends on the rigid body motion between the camera and the surface being viewed, as well as the shape of the surface. Related "gradient based" methods have been exploited before in so-called "direct" motion vision [3], [4], [5], [6], [7].

* Department of Electrical Engineering and Computer Science and CSAIL, MIT, Cambridge, MA 02139, USA bkph@csail.mit.edu
** Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA yajufang@csail.mit.edu
*** Department of Electrical Engineering and Computer Science, and MTL, MIT, Cambridge, MA 02139, USA IMasaki@aol.com

1-4244-1068-1/07/$25.00 ©2007 IEEE.

A related problem is that of recovering the "focus of expansion" (FOE). Spatial and temporal derivatives of brightness can be used to locate the point in the image towards which motion is taking place [7]. One difference between the two tasks is that, in the FOE case, the intent is to be *insensitive* to the magnitude of the velocity and distances to points in the scene, while in the TTC case the intent is to be insensitive to the direction of motion. In some sense the two tasks are complementary: one finds the direction of motion (FOE) while the other finds the time until we 'get there' (TTC). While a FOE method may exploit 'stationary points' — points where the time derivative of brightness is zero while the spatial derivatives are not [7] — such points do not help in the determination of the TTC since the information there is insensitive to velocity.

A. Constant Brightness Assumption

Methods for finding the FOE and TTC can both be based on the "constant brightness assumption":

$$\frac{d}{dt}E(x,y,t) = 0 \tag{4}$$

which encapsulates the observation that in many situations the brightness of the image of a point in the scene does not change significantly as it moves in the image [1]. The above total derivative can be expanded into:

$$uE_x + vE_y + E_t = 0 \tag{5}$$

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 partial derivatives of brightness w.r.t. x, y, and t.

The constant brightness assumption naturally does not apply in all cases. It is violated by specular surfaces, for example, 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 reasonable — and useful — approximation.

B. Perspective Projection and the Motion Field

It is convenient to establish a camera-oriented coordinate system, with the origin at the COP, the Z axis along the optical axis (i.e. the perpendicular dropped from the COP to the image plane), and the X and Y axes parallel to axes of the image sensor. Image coordinates x and y are measured from the principal point (foot of the perpendicular dropped from the COP) — *not* an arbitrary reference such as the corner of the sensor array. Finally, the units of measurement for x and y are the same as that used for the principal distance, f (e.g. the inter-pixel spacing). The perspective projection equations of image formation can then be written in the simple form:

$$\frac{x}{f} = \frac{X}{Z} \quad \text{and} \quad \frac{y}{f} = \frac{Y}{Z} \tag{6}$$

where X, Y, and Z are coordinates of a point in space, while x and y are the corresponding image coordinates These equations are subject to the familiar scale factor ambiguity since multiplying X, Y and Z by the same factor does not change the image coordinates x or y.

By differentiating the perspective projection equations (6) with respect to time, we obtain

$$\frac{u}{f} = \frac{U}{Z} - \frac{X}{Z}\frac{W}{Z} \quad \text{and} \quad \frac{v}{f} = \frac{V}{Z} - \frac{Y}{Z}\frac{W}{Z} \tag{7}$$

where $(u,v) = (\dot{x},\dot{y})$ is the motion field and $(U,V,W) = (\dot{X},\dot{Y},\dot{Z})$ is the velocity of a point on the object relative to the sensor (which is *opposite* to the motion of the sensor relative to the object). The motion field is also subject to a scale factor ambiguity, since multiplying the coordinates X, Y, and Z and the velocity components U, V, and W by the same factor does not change u or v.

Using the perspective projection equations (6) we can rewrite the above in the form

$$\frac{u}{f} = \frac{U}{Z} - \frac{x}{f}\frac{W}{Z} \quad \text{and} \quad \frac{v}{f} = \frac{V}{Z} - \frac{y}{f}\frac{W}{Z} \tag{8}$$

or

$$u = \frac{1}{Z}(fU - xW) \quad \text{and} \quad v = \frac{1}{Z}(fV - yW) \tag{9}$$

This can also be written in terms of the FOE (x_0, y_0) as

$$u = -\frac{W}{Z}(x - x_0) \quad \text{and} \quad v = -\frac{W}{Z}(y - y_0) \tag{10}$$

where $x_0 = f(U/W)$ and $y_0 = f(V/W)$.

III. Translation Relative to a Planar Surface

We show details of three particular cases: (A) Translational motion along the optical axis towards a planar surface perpendicular to the optical axis; (B) Translational motion in an *arbitrary* direction relative to a planar surface that is perpendicular to the optical axis; and (C) Translational motion along the optical axis relative to a planar surface of *arbitrary* orientation. Clearly (A) is a special case of both (B) and (C).

These special cases can be generalized to arbitrary translational motion relative to a planar surface of arbitrary orientation. While the simpler cases have closed form solutions, the more general case requires non-linear optimization techniques. The basic principle stays the same, however.

A. Translational Motion Along the Optical Axis towards a Plane Perpendicular to the Optical Axis

In this case the motion field is simple, since $U = V = 0$. Substituting $u = -x(W/Z)$ and $v = -y(W/Z)$ into the brightness change equation yields

$$-\frac{W}{Z}(xE_x + yE_y) + E_t = 0 \tag{11}$$

or

$$CG + E_t = 0 \tag{12}$$

where $C = -W/Z$ is the inverse of the TTC, and G is a short-hand for the "radial gradient" $(xE_x + yE_y)$. We can formulate a least squares method that has us minimize

$$\sum\left(CG + E_t\right)^2 \tag{13}$$

where the sum is over all pixels of a region of interest (which could be the whole image). Differentiating w.r.t. C and setting the result equal to zero yields

$$\sum(CG + E_t)G = 0 \tag{14}$$

or

$$C\sum G^2 = -\sum GE_t \tag{15}$$

So C, the inverse of the TTC, is given by

$$C = -\sum GE_t \,/\sum G^2 \tag{16}$$

which shows the importance of the "radial gradient" term $G = xE_x + yE_y$. Note that the computation requires only accumulation of products of brightness gradients, time derivatives of brightness, and image coordinates x and y.

When the translational motion is not along the optical axis, but is known, then the above computation can be done using the modified "radial gradient"

$$G' = (x - x_0)Ex + (y - y_0)Ey, \tag{17}$$

where (x_0, y_0) is the location of the known FOE.

B. Arbitrary Translational Motion Relative to a Plane Perpendicular to the Optical Axis

In this case, the translational motion need not be in the direction of the optical axis of the imaging system (nor perpendicular to the surface). Substituting the equations for the motion field components u and v into the brightness change constraint yields

$$AE_x + BE_y + CG + E_t = 0 \tag{18}$$

where $A = f(U/Z)$, $B = f(V/Z)$, $C = -W/Z$, and G is again a convenient short-hand for the "radial gradient" $(xE_x + yE_y)$. We can formulate a least squares method that has us minimize

$$\sum\left(AE_x + BE_y + CG + E_t\right)^2 = 0 \tag{19}$$

To find the best fit values of A, B, and C, we differentiate with respect to A, B, and C and set the results equal to zero:

$$\begin{aligned}\sum\left(AE_x + BE_y + CG + E_t\right)E_x &= 0,\\ \sum\left(AE_x + BE_y + CG + E_t\right)E_y &= 0,\\ \sum\left(AE_x + BE_y + CG + E_t\right)G &= 0.\end{aligned} \tag{20}$$

or

$$\begin{aligned}A\sum E_x^2 + B\sum E_xE_y + C\sum GE_x &= -\sum E_xE_t,\\ A\sum E_xE_y + B\sum E_y^2 + C\sum GE_y &= -\sum E_yE_t,\\ A\sum GE_x + B\sum GE_y + C\sum G^2 &= -\sum GE_t.\end{aligned} \tag{21}$$

We can solve the three linear equations to obtain the unknowns A, B, and C. Note that the coefficients of the symmetric 3×3 matrix are all sums of products of components of the brightness gradient and image coordinates, while the quantities on the right-hand sides of the equations are sums of products of components of the brightness gradient, image coordinates, and the time derivative of brightness.

We are typically only interested in C, since the TTC is the inverse of C. If desired, however, we can find the FOE as well using

$$x_0 = -A/C \quad \text{and} \quad y_0 = -B/C \tag{22}$$

Note that the principal distance, f, need not be known in order to compute the TTC or the FOE. If f is known, however, then the actual direction of translational motion can be computed using

$$\frac{U}{W} = -\frac{1}{f}\frac{A}{C} \quad \text{and} \quad \frac{V}{W} = -\frac{1}{f}\frac{B}{C} \tag{23}$$

Finally, if $U = V = 0$, that is, if the translational motion happens to be along the optical axis, then the least squares problem simplifies, leading to a single equation for C only, as already discussed above in case (A).

C. Translational Motion Along the Optical Axis Relative to an Arbitrary Plane

In this case, the planar surface need not be oriented perpendicular to the optical axis (or the direction of the translational motion). Let p and q be the slopes of the planar surface in the X and Y directions. Then we can write

$$Z = Z_0 + pX + qY \tag{24}$$

The perspective projection equation yields $X = (x/f)Z$ and $Y = (y/f)Z$. Substituting into the above equation for the plane we obtain

$$Z\left(1 - p\frac{x}{f} - q\frac{y}{f}\right) = Z_0 \tag{25}$$

The motion field is $u = -x(W/Z)$ and $v = -y(W/Z)$, when $U = V = 0$. Substituting the expression for Z given above in these expressions for u and v and then inserting these into the brightness change constraint equation leads to

$$-G\frac{W}{Z_0}\left(1 - p\frac{x}{f} - q\frac{y}{f}\right) + E_t = 0 \tag{26}$$

where G is again a convenient short-hand for the "radial gradient" $(xE_x + yE_y)$. The above can be written

$$G(C + Px + Qy) + E_t = 0 \tag{27}$$

where $P = (p/f)(W/Z_0)$, $Q = (q/f)(W/Z_0)$, and also $C = -W/Z_0$. Since the translational motion here is along the optical axis, the contact point on the plane is $(0, 0, Z_0)^T$, and so the TTC is again just the inverse of C.

We can formulate a least squares problem that has us minimize

$$\sum\big(G(C + Px + Qy) + E_t\big)^2 \tag{28}$$

where the sum is over all pixels of a region of interest (which could be the whole image). To find the best fit values of P, Q, and C, we differentiate with respect to P, Q, and C and set the three results equal to zero:

$$\begin{aligned}\sum\big(G(C + Px + Qy) + E_t\big)Gx &= 0,\\ \sum\big(G(C + Px + Qy) + E_t\big)Gy &= 0,\\ \sum\big(G(C + Px + Qy) + E_t\big)G &= 0.\end{aligned} \tag{29}$$

or

$$P\sum G^2x^2 + Q\sum G^2xy + C\sum G^2x = -\sum GxE_t,$$
$$P\sum G^2xy + Q\sum G^2y^2 + C\sum G^2y = -\sum GyE_t, \quad (30)$$
$$P\sum G^2x + Q\sum G^2y + C\sum G^2 = -\sum GE_t.$$

We can solve these three linear equations to obtain the unknowns P, Q, and C. Note that the coefficients of the symmetric 3×3 matrix are all sums of products of components of the brightness gradient and image coordinates, while the quantities on the right-hand sides of the equations are sums of products of components of the brightness gradient, image coordinates, and the time derivative of brightness.

The TTC is the inverse of C since $C = -W/Z_0$. The principal distance, f, need not be known to compute the TTC. If f is known, then the actual surface orientation can also be determined using

$$p = -f\frac{P}{C} \quad \text{and} \quad q = -f\frac{Q}{C} \quad (31)$$

Finally, if $p = q = 0$, that is, if the planar surface happens to lie perpendicular to the optical axis, then the least squares problem simplifies, again leading to a single equation in C only, as already discussed above in case (A).

IV. Experiments

The new method for recovering the TTC was implemented as a MontiVision® "filter" and also in MatLab®. Only the green channel of the images was used. Images were sub-sampled after approximate low-pass filtering. Block averaging was used as a computationally cheap approximation to low pass filtering. Importantly, some of the extra bits in the sums of blocks of pixel values were retained rather than discarded in the division by the number of picture cells in a block. The partial derivatives were estimated in a consistent way using a $2\times2\times2$ cube of pixel values from two images as illustrated in Fig. 2 of [1]. A threshold on the time derivative of brightness blocked contributions from parts of the image that did not change.

Experiments were performed with: (A) Synthetic Sequences; (B) Stop Motion Sequences; and (C) Video from a camera mounted on an automobile.

A. Synthetic Image Sequences

The actual motion of the camera relative to the object being imaged must be known in order to test the accuracy of the method, and to discover how the accuracy depends on parameters such as the sub-sampling factor and the threshold on the time derivative. Synthetic image sequences can be generated with arbitrary known "ground truth." They can be based on a single image which is shifted and magnified so as to simulate a specified translational motion relative to a planar surface. Such sequences were generated from side views of large trucks for assumed constant forward motion with various positions for the FOE.

The algorithm produced TTC values that dropped linearly as expected. There was a bias in the estimate towards larger values than the actual TTC values (i.e. the motion was

Fig. 1. One of 225 frames of a synthetic image sequence.

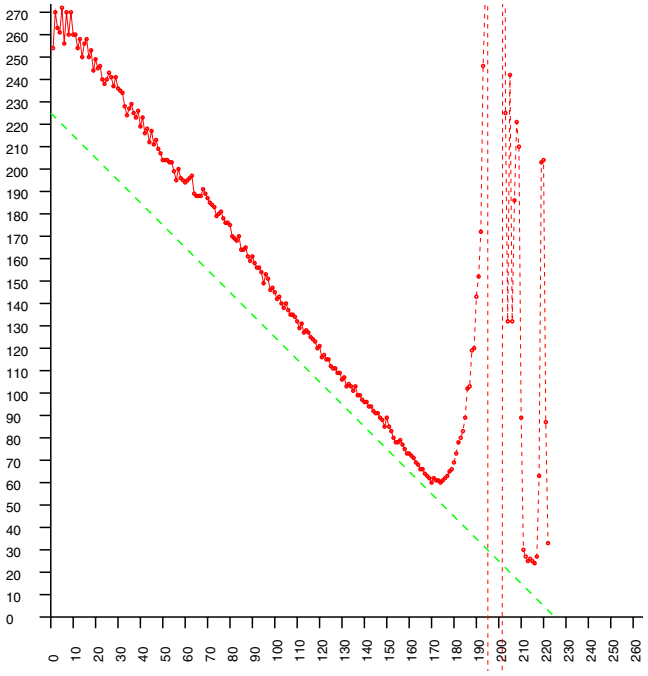

Fig. 2. TTC from algorithm (red dots) for synthetic image sequence compared to true TTC (dashed green line). The horizontal axis shows the frame number; the vertical axis shows the TTC.

underestimated somewhat). The bias could be as much as 20% for large TTCs, dropping to less than 10% for smaller TTCs. The systematic bias was particularly noticeable when images were not averaged and sub-sampled, but dropped significantly even for 2×2 block averaging and sub-sampling.

Near the end of a sequence, the results became unreliable. There appear to be two reasons for this: One is that at the very end of a sequence, the original image has to be greatly magnified and so appears "out of focus" or blurry, lacking high frequency content, and may have artifacts resulting from the interpolation method.

However, the main problem near the end of the sequence is temporal aliasing, in that the frame to frame change is

very large (e.g. when the TTC is 10, the size of the image of an object changes by 10%). As with other gradient-based methods, it appears that results become unreliable when frame to frame image movement is greater than about half the size of typical "texture elements." If the expected size of texture elements is not known, then a useful rule of thumb appears to be, that, to obtain reasonable results, motion should be less than about a pixel from frame to frame in most parts of the image region used.

This implies that spatial averaging and sub-sampling can change the range in which TTC estimates are reliable, since the effective "pixel" size is increased by sub-sampling. This was confirmed experimentally: reasonable TTC estimates were available closer to the actual time to contact with sub-sampled images. For example, with the FOE in the center of 480×640 pixel images, sub-sampling using 4×4 block averages yielded reliable TTC estimates down to a TTC of about 60 frames, With 16×16 block averaging and sub-sampling, good results are obtained down to a TTC of less than 15 frames.

Increased spatial averaging and subsampling comes with a price, in that the estimation of *long* TTCs becomes less accurate. Long TTCs are already hard to estimate accurately because the frame to frame motion of image patterns is only a small fraction of a pixel. This problem is only made more difficult with spatial subsampling. However, *temporal* averaging and sub-sampling can extend the range of TTC estimates to larger values. Taken together this suggests that a wide range of TTCs can be accommodated if spatial averaging and subsampling is used for short TTCs and temporal averaging and subsampling for long TTCs.

B. Stop Motion Image Sequences

Fig. 3. One of 130 frames of a stop motion image sequence.

Artificial video sequences with known "ground truth" can also be constructed using a stop motion technique, where the camera — or the object — is moved by a controlled amount between exposures. This approach does require some care, since accurate increments in position are required For example, if the motion increment is 10 mm, then the motion has to be accurate to better than 1 mm in order to achieve even 10% accuracy in the effective TTC.

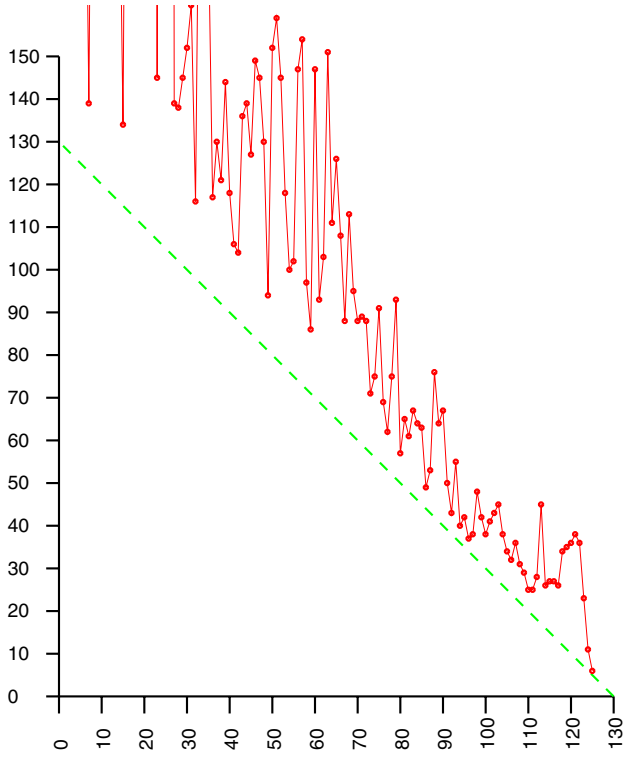

Fig. 4. TTC from algorithm (red dots) for stop motion image sequence compared to true TTC (dashed green line). The horizontal axis shows the frame number; the vertical axis shows the TTC.

Video sequences of toy cars were created using the stop motion method, moving either the camera or the toy car in regular steps. Sequences involving camera motion are subject to small camera rotations that are hard to avoid and even small camera rotations cause significant apparent lateral image motions. It was somewhat easier to create reasonable stop motion sequences with the toy car moving instead. we found that stop motion sequences produced with ordinary digital cameras suffer from the effects of automatic focus and automatic exposure adjustments, as well as artifact introduced by image compression.

The estimated TTCs were again found to be somewhat larger than the true TTCs, particularly for large TTCs. This may be in part because, when the toy car was far away, a large fraction of the image was constant background. This could be avoided by masking or segmenting the image. At the other end of the range, small TTCs were relatively inaccurate because of the large motions of patterns from frame to frame and de-focus, as explained above. Overall the accuracy was less than in the synthetic image case mostly because of the difficulty of accurately positioning the toy car by hand.

C. Video from Cameras mounted on Automobiles

Finally, the method was tested with real video sequences taken with a camera mounted on an automobile driven around

Cambridge, Massachusetts.

The original "log" file recorded in the vehicle contained 600×800 images taken at 57 FPS with a camera equipped with a standard "Bayer" color filter mask. The "log" file also contained accurate timing information (analysis of which indicated that in a number of places individual frames were lost). Video sequences were extracted from the log file in the form of 300×400 images in normal RGB color format.

Fig. 5. One of 400 frames of a video sequence taken from a moving automobile.

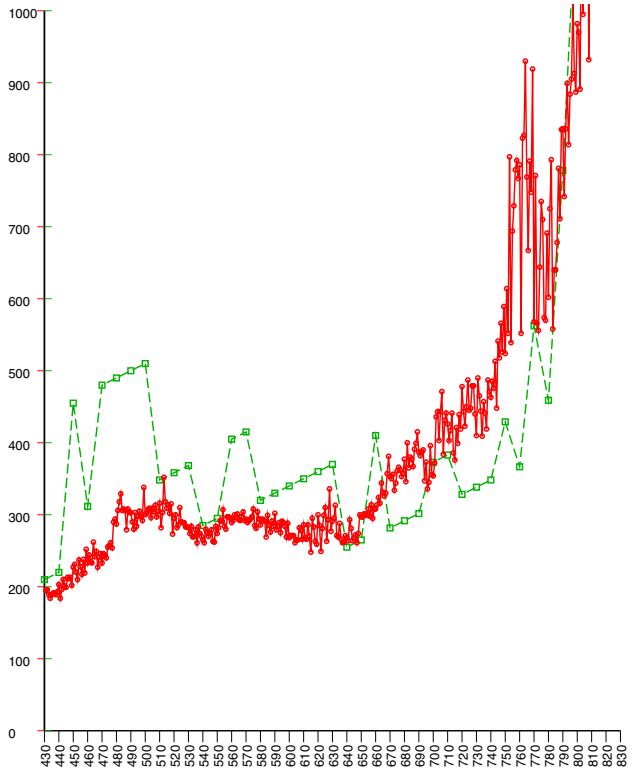

Fig. 6. TTC from algorithm (red dots) applied to video sequence compared to manually estimated TTC (green squares). The horizontal axis shows the frame number; the vertical axis shows the TTC.

The TTC results obtained agreed with visual estimates of vehicle motion and distances. Interestingly, in the segment used, the driver appeared to initially brake so as to keep the TTC more or less constant. The vehicle was then brought to a complete halt (at which point the computation of the TTC become unstable since C approached zero). Unfortunately, the actual "ground truth" is not known in this case.

In order to provide some way of accessing the performance of the TTC algorithms in this real world setting, a manual method for estimating the TTC from the sequence was devised. Visual estimates of the sizes of objects in the images were made and differences between sizes observed in different frames used to estimate the TTC using eq. (3). It turned out, however, that frame to frame changes in size — being small fractions of a pixel — were too small to be measured. So, instead, changes in size over ten frames were used. Even then, serious quantization occurred, because the image size could be estimated only to about a pixel accuracy.

Graphs of the TTC from the algorithm and the manually estimated TTC generally agree, although detailed comparison is not possible because of the coarse quantization of the manual estimates. The difficulty with manual estimation of the TTC once again illustrates that the TTC algorithm presented here works with remarkably small image motions. For example, when the size of the image of the van is about 100 pixels, and the estimated TTC is 500 frames, the frame to frame change in size of the image is only 0.2 pixel and the motion of either end of the van is about 0.1 pixel. Measuring the TTC to an accuracy of say 10% requires, in effect, measuring image positions with an accuracy better than 0.01 pixel.

In the case of these real world video sequences, the TTC estimated by the algorithm generally tended to be somewhat lower than that estimated manually — in contrast to the results with the synthetic sequences. This is because faster moving parts of the image corresponding to nearby objects contributed to the result. This effect could be reduced by masking or segmentation of the image.

The computing time per frame depends, of course, on the number of pixels and the sub-sampling rate. It is about 3 msec on a 1.7 GHz Pentium® for 400×300 images subsampled on a 4×4 grid. For high sub-sampling rates, the cost of block averaging dominates the computation, which drops to below 2 msec per frame.

By the way, since TTC estimates in general, and manual estimates in particular, tend to be noisy, it can be instructive to instead plot

$$\int_0^t \frac{1}{T(t)} dt \quad \text{versus} \quad \log_e \left(\frac{s(t)}{s(0)} \right) \tag{32}$$

(see Fig. 7 for example).

V. Conclusions

A new "direct," gradient-based method for estimating the time to contact has been described and demonstrated on synthetic sequences, stop motion sequences, as well as real video. The method does not require feature detection, feature

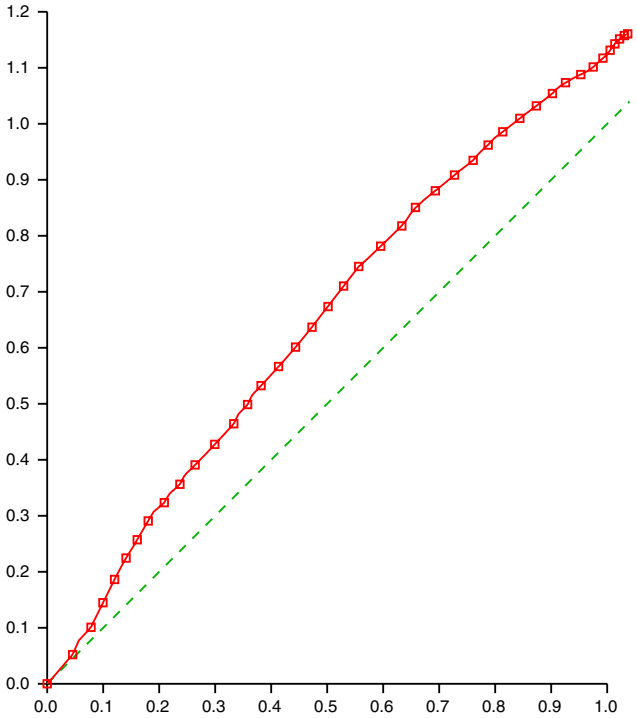

Fig. 7. Integral of 1/TTC (red dots) versus $\log_e(s(t)/s(0))$ (marked every ten frames). The diagonal dashed green line shows the ideal relationship.

tracking, or estimation of the optical flow, and has low latency. Spatial averaging and sub-sampling can extend the range of operation to small TTCs, while temporal averaging and sub-sampling can extend the range to large TTCs.

In practice, some form of image segmentation may be useful in suppressing contributions from image regions moving in ways different from those of the object of interest.

VI. Acknowledgments

Log files from MIT's DARPA Urban Challenge vehicle were kindly made available by David Moore. Measurements of image feature sizes used to estimate the TTC manually were made in some video sequences by Orçun Kurugöl.

References

[1] B.K.P. Horn & B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Vol. 16, No. 1-–3, August 1981, pp. 185-–203.

[2] A.R. Bruss & B.K.P. Horn, "Passive Navigation," *Computer Vision, Graphics, and Image Processing*, Vol. 21, No. 1, January 1983, pp. 3-–20.

[3] B.K.P. Horn & S. Negahdaripour, "Direct Passive Navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 1, January 1987, pp. 168-–176.

[4] B.K.P. Horn & E.J. Weldon, Jr., "Direct Methods for Recovering Motion," *International Journal of Computer Vision*, Vol. 2, No. 1, June 1988, pp. 51-–76.

[5] S. Negahdaripour & B.K.P. Horn, "A Direct Method for Locating the Focus of Expansion," *Computer Vision, Graphics and Image Processing*, Vol. 46, No. 3, June 1989, pp. 303-–326.

[6] B.K.P. Horn, "Parallel Analog Networks for Machine Vision," in *Artificial Intelligence at MIT: Expanding Frontiers*, edited by Patrick H. Winston and Sarah A. Shellard, MIT Press, Vol. 2, pp. 531–573, 1990.

[7] I.S. McQuirk, B.K.P. Horn, H.-S. Lee, & J.L. Wyatt "Estimating the Focus of Expansion in Analog VLSI," *International Journal of Computer Vision*, Vol. 28, No. 3, 1998, pp. 261-–277.

[8] E. De Micheli, V. Torre, & S. Uras, "The Accuracy of the Computation of Optical Flow and of the Recovery of Motion Parameters," *IEEE Transactions on PAMI*, Vol. 15, No. 5, May 1993, pp. 434-447.

[9] T.A. Camus, "Calculating Time-to-Contact Using Real-Time Quantized Optical Flow," Max-Planck-Institut fur Biologische Kybernetik, Technical Report No.14, February, 1995.

[10] P. Guermeur & E. Pissaloux, "A Qualitative Image Reconstruction from an Axial Image Sequence," *30th Applied Imagery Pattern Recognition Workshop*, AIPR 2001, IEEE Computer Society, pp. 175–181.

[11] S. Lakshmanan, N. Ramarathnam, & T.B.D. Yeo, "A Side Collision Awareness Method," *IEEE Intelligent Vehicle Symposium 2002*, Vol. 2, pp. 640–645, 17–21 June 2002.

[12] H. Hecht & G.J.P. Savelsbergh (eds.), *Time-To-Contact*, Elsevier, Advances in Psychophysics, 2004.