

Mapping Local Image Deformations into Depth

Stephen Benoit and Frank P. Ferrie

Department of Electrical and Computer Engineering
and the Centre for Intelligent Machines
McGill University, Montréal, CANADA

Abstract. The paper presents a 2 frame structure-from-motion algorithm that operates by mapping local changes (image deformations) into estimates of time-to-collision (TTC). For constant velocity motion of the camera in a stationary scene, time-to-collision amounts to coarse depth data - useful for navigation and qualitative scene understanding. The theory is supported by a set of experiments demonstrating accurate TTC recovery from video sequence data acquired by a mobile robot.

1 Introduction

Recovery of structure from motion has been examined from a variety of approaches, mainly feature point extraction and correspondence[5, 7] or computing dense optical flow[6, 8, 1]. Typically, the Fundamental Matrix framework or a global motion model is used to solve for global motion after which the relative 3-D positions of points of interest in the scene can be computed[9, 11]. Appearance-based methods have been mostly discarded for structure from motion because much of the shape and motion information are so confounded that they cannot be recovered separately or locally[3]. Soatto proved that perspective is non-linear, therefore no coordinate system will linearize perspective effects[10]. However, in [2] we showed that some useful structure and motion information could indeed be directly recovered, namely time-to-collision (TTC) and heading information. In this paper we present a practical, two-frame algorithm for recovering TTC and experimental results showing how it can be used to recover a qualitative 3-D scene description from video sequences acquired by a mobile robot.

2 Theory

The key result from [2] is that useful shape and motion information can be extracted from the analysis of local image deformations along 1-D neighbourhoods. The set-up is shown in Figure 1. Each oriented, rectangular window corresponds to the image of a cross-section of a 3-D surface, essentially a *normal section* in the context of differential geometry[4]. There is a precise relationship between the structure and motion of this cross-section and deformations of two corresponding 1-D windows, $(x_a, y_a, \theta_i, t_0)$ and $(x_a, y_a, \theta_i, t_1)$ (Figure 1), that is made explicit for particular choice of image formation model.

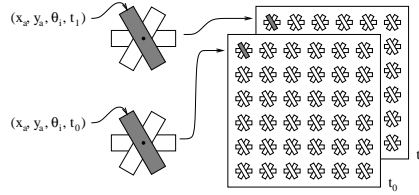


Fig. 1. Oriented slits at image coordinates (x_a, y_a) at multiple orientations θ_i cover the image plane at instants t_0 and t_1

The image formation model we use, i.e., the *forward-model*, is shown schematically in Figure 2. The mapping from cross-section to image is defined by the perspective camera model shown in Figure 2a, and the motion and structure model relating 3-D change to appearance is shown in Figure 2b, the latter comprising 5 parameters, $\mathbf{m} = (\Omega, \delta, \eta, \beta, k)$. Referring to Figure 2b, the 3-D cross-section is characterized by a curvature K , a normal vector \mathbf{N} and distance from the viewpoint, d . Distance d scales all lengths of the diagram, so it is factored out to a canonical representation with unit distance between first viewpoint VP and the fixation point on the surface O . The surface normal vector \mathbf{N} at O is encoded by the angle η with respect to the first view axis $VP - O$. The curvature of the canonical surface, the reciprocal of the radius of the circular approximation to the surface, becomes $k = Kd$.

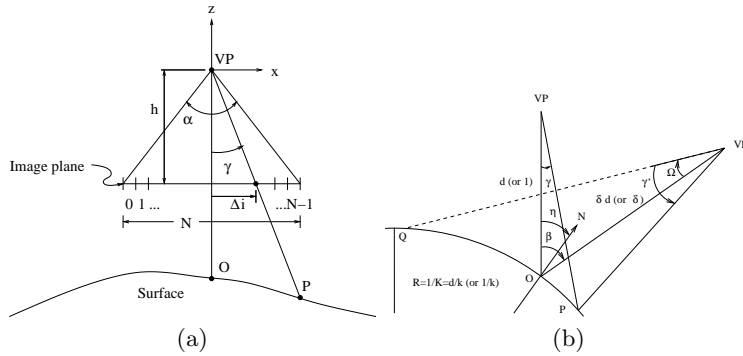


Fig. 2. (a) The 1-D camera model. (b) Motion and structure model for a surface cross-section.

The motion model is chosen to minimize image deformation due to translation, defining the second viewpoint VP' at a given distance δ at an angle β from the first view axis $VP - O$. VP' is fixated on point Q on the surface, a view rotation Ω away from the first fixation point O . Collectively, the camera and shape and motion models are sufficient to describe the forward mapping of

a 3-D contour, parameterized by orientation, η , and curvature, k , onto a 1-D image slit, \mathcal{I} , and then onto a corresponding image slit, \mathcal{I}' via translation, Ω , change in distance to viewer, δ , and change in viewpoint direction, β . Details are given in [2].

Solution of the inverse problem involves recovery of \mathbf{m} given two 1-D windows, \mathcal{I} and \mathcal{I}' . Here an appearance-based approach is used. For the experiments presented in this paper, the parameter space \mathbf{m} is quantized into 21 levels for Ω and δ respectively, and 5 levels each for η , β , and k . This follows from [2] - only Ω and δ are recoverable, but fortunately these parameters are sufficient to recover TTC. For each of the 55,125 instances of \mathbf{m}_i , we create corresponding window pairs, \mathcal{I}_i and \mathcal{I}'_i , by applying the forward model shown earlier in Figure 2.

Let \mathcal{I}_i and \mathcal{I}'_i be represented by $n \times 1$ vectors such that $\mathcal{I}_i = \mathbf{H}_i \mathcal{I}'_i$, where \mathbf{H}_i is an $n \times n$ matrix that encodes the bi-directional mapping from \mathcal{I}_i to \mathcal{I}'_i and vice-versa. We refer to this as a *correspondence matrix*, and it is relatively straightforward to determine given \mathcal{I}_i and \mathcal{I}'_i . A practical procedure for computing \mathbf{H}_i is given in [2]. To minimize the effects of intensity variations between frames, before computing \mathbf{H}_i , \mathcal{I}_i and \mathcal{I}'_i are first normalized as $\tilde{\mathcal{I}}_i, \tilde{\mathcal{I}}'_i$ for a zero mean intensity and a contrast of 1 by finding the image's brightness $\mu_{\mathcal{I}}$ and contrast $\Delta_{\mathcal{I}}$.

$$\begin{aligned} \mu_{\mathcal{I}} &\triangleq \frac{\sum_i \mathcal{I}_i + \sum_i \mathcal{I}'_i}{2N}, \\ \Delta_{\mathcal{I}} &\triangleq \frac{\max_i (|\mathcal{I}_i - \mu_{\mathcal{I}}|, |\mathcal{I}'_i - \mu_{\mathcal{I}}|)}{\mu_{\mathcal{I}}} \in (0, 1), \end{aligned}$$

$$\tilde{\mathcal{I}} = \frac{\mathcal{I} - \mu_{\mathcal{I}}}{\mu_{\mathcal{I}} \Delta_{\mathcal{I}}}, \quad \tilde{\mathcal{I}}' = \frac{\mathcal{I}' - \mu_{\mathcal{I}}}{\mu_{\mathcal{I}} \Delta_{\mathcal{I}}}. \quad (1)$$

Another key result from [2] concerns the singular value decomposition (SVD) of \mathbf{H}_i . Let \mathbf{U}_i and \mathbf{V}_i be left and right matrices respectively of the SVD of \mathbf{H}_i , and let $\mathbf{U}_{\mathbf{k}_i}$ and $\mathbf{V}_{\mathbf{k}_i}$ be their corresponding k^{th} order approximations. The latter correspond to the first k columns of \mathbf{U}_i and \mathbf{V}_i respectively sorted by singular values. Now let feature vector \hat{w}_i represent the image vectors $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{I}}'$ as follows:

$$\hat{w}_i = \left[\mathbf{U}_{\mathbf{k}_i}^T / \sqrt{2} : \mathbf{V}_{\mathbf{k}_i}^T / \sqrt{2} \right] \begin{bmatrix} \tilde{\mathcal{I}} \\ \dots \\ \tilde{\mathcal{I}}' \end{bmatrix}, \quad (2)$$

where $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{I}}'$ are a pair of inputs for which we wish to test \mathbf{H}_i . The feature vector \hat{w}_i is now the best parameterization for the image pair assuming deformation \mathbf{H}_i . The residual error can be computed by projecting the feature vector back into the image space. If the assumed deformation \mathbf{H}_i is sufficiently close to the scene geometry, then residual signal error \mathbf{r}_i , the difference between the

original image signal and the reconstructed image signal will be low,

$$\mathbf{r}_i = \left(\begin{bmatrix} \tilde{\mathcal{I}} \\ \dots \\ \tilde{\mathcal{I}}' \end{bmatrix} - \begin{bmatrix} \mathbf{U}_{\mathbf{k}_i} \\ \dots \\ \mathbf{V}_{\mathbf{k}_i} \end{bmatrix} \hat{w}_i \right) / \left\| \begin{bmatrix} \tilde{\mathcal{I}} \\ \dots \\ \tilde{\mathcal{I}}' \end{bmatrix} \right\|. \quad (3)$$

The likelihood of correspondence \mathbf{H}_i given evidence $(\tilde{\mathcal{I}}, \tilde{\mathcal{I}}')$ can be expressed as a function $\mathcal{L}(\mathbf{H}_i | \tilde{\mathcal{I}}, \tilde{\mathcal{I}}')$,

$$\mathcal{L}(\mathbf{H}_i | \tilde{\mathcal{I}}, \tilde{\mathcal{I}}') \triangleq e^{-\|\mathbf{r}_i\|} \in (0, 1]. \quad (4)$$

The uncertainty of the maximum likelihood choice can be expressed as the entropy h of the likelihoods for all the different hypotheses,

$$h_i = \frac{-\sum_{i=1}^n \mathcal{L}(\mathbf{H}_i | \tilde{\mathcal{I}}, \tilde{\mathcal{I}}') \log(\mathcal{L}(\mathbf{H}_i | \tilde{\mathcal{I}}, \tilde{\mathcal{I}}'))}{\log(n)} \in (0, 1]. \quad (5)$$

3 Implementation

In practice, the computational complexity of searching for \mathbf{H}_i is quite manageable [2]. Only Ω and δ are observable, so η , β , and k can be marginalized out by averaging the 125 matrices associated with each Ω, δ pair. This reduces the search space to 441 distinct \mathbf{H}_i . Ω can be found independently by marginalizing δ , but δ must be determined jointly with Ω . The net result is that \mathbf{H}_i can be found with a maximum of $21 + 21 = 42$ matches, in each of n image orientations ($n = 6$ in this paper), for each $i \times j$ neighbourhood of the input image pair.

Time-to-collision is carried by the δ parameter, which indicates the ratio of the distance between the new viewpoint to the surface over the distance between the old viewpoint to the surface,

$$\delta = \frac{\|VP' - O\|}{\|VP - O\|}. \quad (6)$$

The time between observations, Δt , is known beforehand. Assuming that the camera's motion relative to the scene will continue at constant velocity, one can estimate how much time will elapse before the camera reaches the point on the surface it is looking at and heading toward,

$$T = \Delta t \left(\frac{\delta}{1 - \delta} \right). \quad (7)$$

Recovering Ω and δ locally in forward time can be augmented by recovering Ω' and δ' by reversing the sequence of the images. A direct method of computing the time to collision \tilde{T} between two images, using both forward and reverse information, separated by a delay of Δt is to average the two contributions,

$$\tilde{T} = \frac{\Delta t}{2} \left(\frac{\delta}{1 - \delta} + \frac{1}{\delta' - 1} \right). \quad (8)$$

Taking contributions from different orientations into account, and weighting by their respective uncertainties (5), we obtain a more robust estimate of TTC,

$$\tilde{T} = \frac{\Delta t}{2n} \sum_{\theta=1}^n \left(\frac{\delta_{\theta} h_{\theta}}{1 - \delta_{\theta} h_{\theta}} + \frac{1}{\delta'_{\theta} h'_{\theta} - 1} \right). \quad (9)$$

4 Experiments

The forward model shown earlier in Figure 2 is used to produce correspondence matrices \mathbf{H}_i indexed by Ω and δ as outlined in Section 2. The range of values for each parameter in experiments are summarized in Table 1.

Symbol	Values	Description
Ω	-4.0°	translate left 32/64 pixels
	0°	no change
	$+4.0^{\circ}$	translate right 32/64 pixels
δ	0.80	zoom in 20%
	1.0	no change
	1.25	zoom out 25%
η	-45°	normal pointing 45° left of VP
	0°	normal pointing toward VP
	$+45^{\circ}$	normal pointing 45° right of VP
β	-10°	VP' moves to left of VP
	0°	VP' stays in line with VP
	$+10^{\circ}$	VP' moves to right of VP
k	-4	concave surface
	0	flat surface
	+4	convex surface

Table 1. Parameters of structure from motion model. N and α are known constants

Applying SVD to each of the \mathbf{H}_i yields corresponding $\mathbf{U}_{\mathbf{k}_i}, \mathbf{V}_{\mathbf{k}_i}$ pairs. These detectors, some of which are shown in Figure 3, are automatically synthesized to optimally recognise the distance-to-viewer change while remaining insensitive to other surface motions.

The one-time offline training, i.e. constructing the 441 64×64 correspondence matrices and their detectors by Singular Value Decomposition, required less than 90 seconds on an Intel Pentium 4 2660 MHz workstation.

The video test sequence was obtained by a video camera on a mobile robot as it travelled along a linear trajectory through a room, taking images at known positions in a fixed direction, looking in the direction of travel. The first and last images of the 11 frame sequence are shown in Figure 4.

The robot's position advances 20cm between each image, hence a velocity of 20cm per unit of time Δt . The map of maximum likelihood $\hat{\delta}$ and the map of time to collision T were computed over a grid of 48×36 slits at 6 orientations and are rendered in Figure 5. Computation time on an Intel Pentium 4 2660 MHz

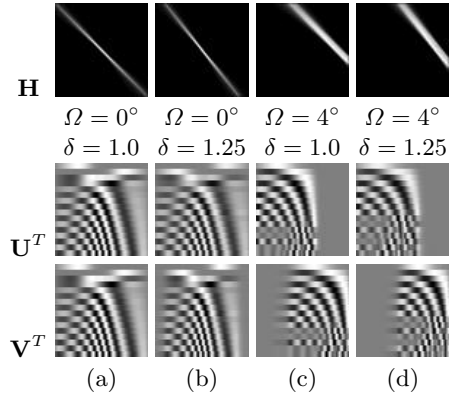


Fig. 3. Correspondence Matrices for some motions. For each \mathbf{H} , black indicates 0, white represents 1. \mathbf{H}_{ij} (row i , column j) indicates the amount of correspondence between \mathcal{I}_i and \mathcal{I}'_j . For \mathbf{U}^T and \mathbf{V}^T , black indicates -1 and white represents +1. Each row of \mathbf{U}^T is a distorted sinusoid to be applied to \mathcal{I} and the same row in \mathbf{V}^T is the corresponding distorted sinusoid for \mathcal{I}' .

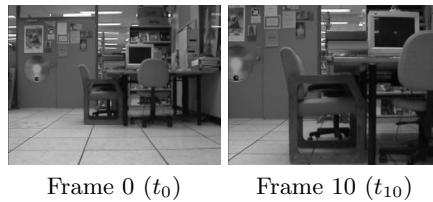


Fig. 4. Lab1 sequence, source images.

workstation was approximately 90 seconds per image frame pair using an exhaustive search. In a practical implementation, the redundancy in the detectors would be reduced using linear combinations of Principal Components, potentially reducing computation time by a factor of 10. Using a sparser sampling of the image plane would further reduce the computation time closer to real time.

One significant observation is that although the floor tile pattern expands closer to the camera due to perspective and the texture of the floor is moving toward the robot, the floor is heading *underneath* the camera and does not appear to be on a collision course with the camera. Because the camera line of sight is parallel to the floor, the algorithm has effectively classified the floor motion as maintaining constant distance from the camera, and thus not an obstacle. The algorithm performs a literal figure-ground separation, and the obstacle blobs are at least qualitatively useful for identifying the location of the nearest obstacles in the image. Next, the quantitative estimates are examined.

Note that in Figure 5, there are holes in the time map between the table and chair legs, where the back wall is beyond the detector's range. The chair near the center of the image started 400cm from the camera, and by the eleventh frame is 200cm from the camera. The chair has a large opening in it, letting

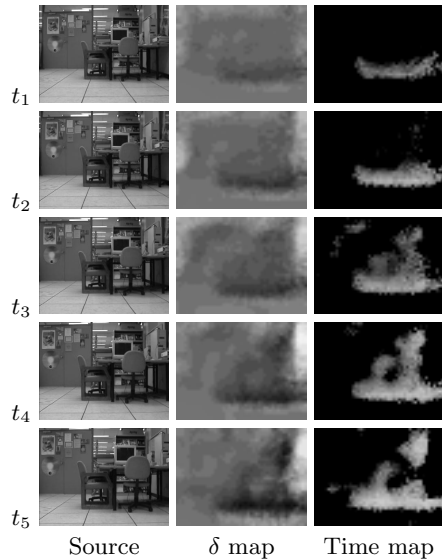


Fig. 5. Results from Lab1 sequence. For the δ map, $\delta = 0.8$ (rapid approach) is rendered as black, $\delta = 1.0$ (no depth change) is middle gray and $\delta = 1.25$ (rapid retreat) is white. The time map indicates proximity (either about to touch or recently touched) as brightness. Dark patches are more than 25 units of time away, either in the future or the past.

a view of the background through. The image slits are based on a model of a continuous surface, so some uncertainty in the maximum likelihood estimates in this situation is unavoidable. The usefulness of the estimates can be shown in Figure 6, comparing the mean of the estimated time to collision of the region around that chair to ground truth from actual measurements.

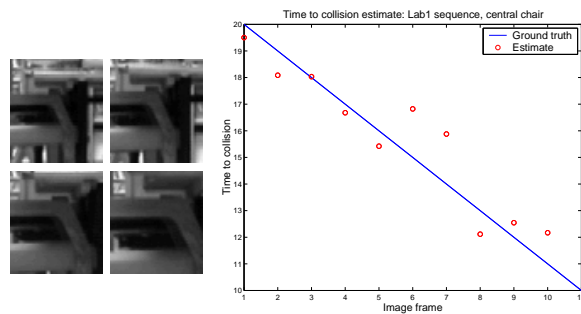


Fig. 6. Neighborhood of central chair in Lab1 sequence. Image patches from frames 0, 3, 6, and 9 (left to right, top to bottom) are shown at left. At right, the mean time to collision with this patch, shown as small circles are compared with the ground truth (line).

5 Conclusions

A time to collision or contact map provides a form of figure-ground separation that may be more informative to mobile robotics than instantaneous range images of similar resolution: it not only provides the instantaneous location of obstacles in the image plane, it also offers a prediction of their future locations. Distance to an obstacle is not the only factor to consider when ranking its importance to navigation. For example, an obstacle 1m away but maintaining its distance from the mobile robot is not significant, but an obstacle 20m away approaching at 2m/sec is critical.

As an added feature of the method proposed in this paper, the floor is naturally ignored in the case when the camera's line of sight is parallel to the floor, a task that is more difficult to achieve using optical flow methods.

The 1-D image slit surface model is often violated along various orientations at different locations in the image plane during the experiment (narrow features such as table legs, poor texture), and as a result, the maximum likelihood estimate at those orientations and locations are given higher uncertainties. Pooling together estimates from other, more confident orientations in the neighborhood leads to group estimates that are more robust to gauge time to collision.

References

1. J. Barron and R. Eagleon. Motion and structure from time-varying optical flow. In *Vision Interface*, pages 104–111, May 1995.
2. S. Benoit and F. P. Ferrie. Towards direct recovery of shape and motion parameters from image sequences. In *Proceedings of ICCV*, pages 1395–1402, Nice, France, October 2003.
3. D. DiFranco and S. Kang. Is appearance-based structure from motion viable? In *2nd International Conference on 3-D Digital Imaging and Modeling*, Ottawa, Canada, Oct. 1999.
4. M. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.
5. B. Krse, N. Vlassis, R. Bunschoten, and Y. Motomura. Feature selection for appearance-based robot localization. In *Proceedings 2000 RWC Symposium*, 2000.
6. H. Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. In *Intl. Conf. Pattern Recognition 2002*, pages I: 291–294, 2002.
7. J. Oliensis. Direct multi-frame structure from motion for hand-held cameras. In *ICPR Vol. I*, pages 889–895, 2000.
8. S. Roy and I. J. Cox. Motion without structure. In *13th Int. Conference on Pattern Recognition, Vol. I*, pages 728–734, Vienna, Austria, August 1996. IEEE.
9. S. Soatto and P. Perona. Dynamic visual motion estimation from subspace constraints. Technical Report CIT-CDS 94-006, California Institute of Technology, Pasadena, CA, Jan. 1994.
10. S. Soatto and P. Perona. On the exact linearization of structure from motion. Technical Report CIT-CDS 94-011, California Institute of Technology, Pasadena, CA, May 1994.
11. C. Tomasi. Input redundancy and output observability in the analysis of visual motion. In *Proc. Sixth Symposium on Robotics Research*, pages 213–222. MIT Press, 1993.