IEEE Conference on Computer Vision
and Pattern Recognition, June 1993

Direction of Heading from Image Deformations

Carlo Tomasi Jianbo Shi

Department of Computer Science
Cornell University, Ithaca, NY 14853

Abstract

We propose a method to compute the direction of heading from the differential changes in the angles between the projection rays of pairs of point features. These angles, the image deformations*, do not depend on viewer rotation, so the key problem of separating the effects of rotation from those of translation is solved at the input. Experiments show both the feasibility of the method on real images and the advantages of using deformations rather than optical flow.*

1 Introduction

As we walk down a hallway, the moving images on our retinas convey enough information to determine our direction of heading. Several researchers have investigated how this direction could be computed, either in the human visual system or by a computer processing images from a moving camera.

The main difficulty of this computation is to separate the effects of viewer rotation from those of viewer translation. In fact, with only translation the task would be quite simple: features in the image move toward or away from a single point, the focus of expansion, which points toward the direction of heading. If the viewer also rotates, however, the focus of expansion vanishes. Unfortunately, the rotational component is often dominant: the effects of translation are small, being inversely proportional to the usually large distance to the scene, while the effects of rotation are independent of distance. This has always been known to movie directors, who use expensive dollies to move their cameras with as little vibration as possible.

In this paper, we propose a method that eliminates the effects of rotation right at the input of the computation. Specifically, we observe that the angle between the projection rays of two features in the scene does not depend on the viewer's rotation, and we use the changes in these angles as the input to our algorithm. In other words, rather than measuring how the image points *move* in the field of view, the traditional flow-based approach, we measure how the image *deforms* over time.

Of course, our method uses the same data as the flow-based methods do, since we compute deformations from feature positions in successive images. However, our method differs in how those data are used: computing deformations removes rotation at the outset, rather than at some later stage of the computation. With noisy data, canceling rotation by computing deformations has two advantages.

1. The direction of heading is computed from deformations by minimizing a residual function with a deeper minimum than the one based on optical flow. Even very small rotations flatten the minimum of the flow-based residual function considerably, leading to a minimization that is more sensitive to noise.

2. The magnitude of deformations with respect to image noise is a direct indication of their reliability for the estimation of the direction of heading. This is not true for optical flow: a large flow may be just caused by viewer rotation, and the differences between flow vectors may be too small with respect to noise to be used for the computation of heading. Deformations, on the other hand, can be monitored as the viewer moves, and heading computed only once they are large enough.

With n feature points there are $n(n-1)/2$ pairs of features, and therefore $O(n^2)$ angles between features. However, only $2n-3$ of these angles are independent. We propose a method for selecting a sufficient set of $O(n)$ points in time $O(n \log n)$.

In the next section, we discuss previous work on the subject. Then, in section 3, we define image deformations and derive an equation that links them to the

This research was supported under NSF Grant IRI-9201751.

direction of heading. In section 4 we combine these equations into a single system of equations for a sufficient set of feature pairs. Then, in section 4, we show how the distances to the feature points in space can be eliminated from the equations, leading to a minimization problem in two variables, which we solve in section 5. Finally, we test our solution on both simulated and real images (sections 6 and 7).

2 Relation with previous work

We now compare our method with others regarding how the effects of rotation are eliminated from the images. We do not discuss methods that require continuous velocity fields or second derivatives of image motion [9] [20] [19], since we use as input the instantaneous image velocities of a set of discrete image points.

The methods presented in [10] [18] [21] are based on the following observation: if a point $\mathbf{p}$ in the first image moves to $\mathbf{q}$ in the second, then the vectors $\mathbf{q}, R\mathbf{p}, \mathbf{T}$, where R is rotation and $\mathbf{T}$ is translation, are coplanar:

$$\mathbf{q} \cdot (R\mathbf{p} \times \mathbf{T}) = 0$$

The three methods above enforce this constraint over several points to compute R and $\mathbf{T}$. The effect of noise on these nonlinear equations depends on R. For instance, if the two vectors $R\mathbf{p}$ and $\mathbf{T}$ happen to be nearly parallel to each other, small perturbations of $\mathbf{p}$ can change the direction of $R\mathbf{p} \times \mathbf{T}$ considerably.

When the translation of the viewer is large, the three methods just cited are preferable to ours, since they make no assumption as to the distance traveled by the viewer. In contrast, we measure differential changes in the angles between features. On the other hand, when the viewer moves little, the methods above can fail altogether when a certain quadratic equation has no real solution [10] [21]. Our method degrades more gracefully: as image deformations become smaller and smaller relative to noise, the uncertainty in the direction of heading grows, but no outright failure occurs.

An observation by Helmholtz [7] has been used in [11] [14] [3] [8]: the vector difference in velocity between two points that are nearby in the image but at different depths is nearly independent of rotation (and exactly so when the points are at the same image location). Changes in this difference, called *motion parallax*, supply sufficient constraints to recover the direction of heading. Our method is also based on motion parallax: our image deformations are the magnitude (in degrees of visual angle) of the vector difference used in those papers. However, by ignoring the direction of the parallax and only considering its magnitude we make parallax independent of rotations exactly, regardless of the image positions of the two feature points. The two hard problems of determining pairs of image features along depth boundaries and measuring their image velocities (given the interference of the boundary) are thereby avoided.

Our approach is similar to those in [2] [1] [12] [6] in that we minimize some residual over the measurements in the least squares sense. Like Heeger and Jepson, we reduce minimization to that of a function of two variables (the parameters for the direction of heading), but we use rotation-independent image deformations rather than image flow, with the advantages mentioned in the introduction.

3 Image deformations

Consider two points $\mathbf{P}$ and $\mathbf{Q}$ in space, as in figure 1. As the viewer moves from $\mathbf{C}$ to $\mathbf{C}'$, the magnitude α of the angle $\mathbf{PCQ}$ formed by the projection rays changes to $\mathbf{PC'Q}$. The *image deformation* is $\dot{\alpha}$, the

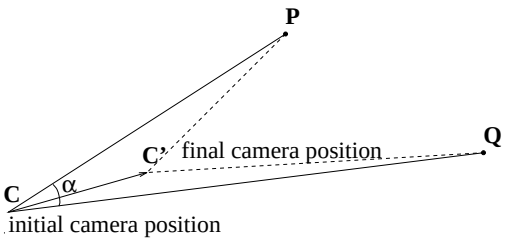

Figure 1: As the viewer moves, the angle α between projection rays $\mathbf{CP}$,$\mathbf{CQ}$ varies. The time derivative of this variation is the *image deformation*.

time derivative of α. The angle α is given by

$$\alpha = \arccos(\mathbf{p}^T\mathbf{q}) \qquad (1)$$

where $\mathbf{p}$ and $\mathbf{q}$ are two unit vectors from the viewer center to the points $\mathbf{P}$ and $\mathbf{Q}$.

To find $\dot{\alpha}$, we first determine the derivative of α with respect to viewer position when $\mathbf{C}$ moves along three special directions: $\mathbf{p}$, $\mathbf{q}$, and the direction

$$\mathbf{r} = \frac{\mathbf{p} \times \mathbf{q}}{|\mathbf{p} \times \mathbf{q}|}$$

orthogonal to $\mathbf{p}$ and $\mathbf{q}$. If p, q, r are the amounts of viewer motion along $\mathbf{p}, \mathbf{q}, \mathbf{r}$, we find that

$$\frac{\partial\alpha}{\partial p} = \frac{\sin\alpha}{|\mathbf{Q}|} \qquad \frac{\partial\alpha}{\partial q} = \frac{\sin\alpha}{|\mathbf{P}|} \qquad \frac{\partial\alpha}{\partial r} = 0\,.$$

If the viewer motion is expressed instead in an orthogonal reference system through its components x, y, z, the chain rule for differentiation yields

$$\begin{bmatrix} \frac{\partial \alpha}{\partial x} \\ \frac{\partial \alpha}{\partial y} \\ \frac{\partial \alpha}{\partial z} \end{bmatrix} = J^T \begin{bmatrix} \frac{\partial \alpha}{\partial p} \\ \frac{\partial \alpha}{\partial q} \\ \frac{\partial \alpha}{\partial r} \end{bmatrix}$$

where the Jacobian J is given by

$$J = \begin{bmatrix} \mathbf{p} & \mathbf{q} & \mathbf{r} \end{bmatrix}^{-1} . \tag{2}$$

Finally, we apply the chain rule once more to compute the derivative of α with respect to time, given the three time derivatives $\mathbf{t}$ of the viewer position $\mathbf{C}$:

$$\dot{\alpha} = sin\alpha \, \mathbf{t}^T J^T \begin{bmatrix} |\mathbf{Q}|^{-1} \\ |\mathbf{P}|^{-1} \\ 0 \end{bmatrix} . \tag{3}$$

We can rewrite equation (3) in a more compact form to obtain the following fundamental measurement equation:

$$b = \mathbf{t}^T A \mathbf{d} \tag{4}$$

where the scalar

$$b = \frac{\dot{\alpha}}{\sin \alpha} \tag{5}$$

is a quantity that can be measured from two or more images, the vector

$$\mathbf{d} = \begin{bmatrix} d_P \\ d_Q \end{bmatrix} = \begin{bmatrix} |\mathbf{P}|^{-1} \\ |\mathbf{Q}|^{-1} \end{bmatrix}$$

collects the reciprocals of the unknown depth values, the columns of the 3×2 matrix A are the known second and first row of J, and the vector $\mathbf{t}$ is the unknown viewer velocity.

4 Combining multiple measurements

With n features instead of two, we can write $n(n-1)/2$ equations like equation (4), one for every pair of features. However, only $2n-3$ of these equations provide independent measurements.

In fact, if two points $\mathbf{p}_1$, $\mathbf{p}_2$ are picked as reference, the positions of all n points is identified up to a mirror flip by the distance between $\mathbf{p}_1$ and $\mathbf{p}_2$ and by the $2(n-2)$ pairs of angles that $\mathbf{p}_1$ and $\mathbf{p}_2$ form with the other points $\mathbf{p}_i$ for $i = 3, \ldots, n$, for a total of $1 + 2(n-2) = 2n-3$ parameters. This construction, however, is only useful as a way to count independent equations. In fact, an arbitrary choice of $\mathbf{p}_1$ and $\mathbf{p}_2$ can lead to long and narrow triangles $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_i)$ yielding a numerically poor system of equations.

Instead, we measure angles α for pairs of image points that are connected by the edges of a Delaunay triangulation [13] of all the points (see figure 5). This triangulation is a planar graph, and thus has $O(n)$ edges. Every vertex is connected to at least two others, leading to a sufficient set of measurement equations. Edges connect nearest neighbors, yielding large ratios in equation (5) for reduced noise sensitivity. Finally, points strictly inside the convex hull belong to triangles that are as close as possible to equilateral, yielding maximally independent measurement equations.

We can then let

$$\begin{aligned} \mathbf{d} &= \begin{bmatrix} d_1 & \cdots & d_n \end{bmatrix}^T \\ \mathbf{b} &= \begin{bmatrix} b_1 & \cdots & b_m \end{bmatrix}^T \end{aligned}$$

where b_k is the left-hand side of equation (4) for edge k, and m is the number of edges in the Delaunay triangulation. If edge k connects points i and j, the matrix A for the k-th measurement equation (4) is formed from the Jacobian $[\mathbf{p}_i \ \mathbf{p}_j \ \mathbf{r}_{ij}]^{-1}$ of equation (2). Because the matrix A for pair (i,j) is equal to that for pair (j,i), but with its columns switched, we can consistently define

$$A_k = \begin{bmatrix} \mathbf{a}_{ji} & \mathbf{a}_{ij} \end{bmatrix}$$

and define the $m \times n$ matrix $A(\mathbf{t})$ whose entry k, l is

$$t_{kl} = \begin{cases} \mathbf{t}^T \mathbf{a}_{ij} & \text{for } l = i \\ \mathbf{t}^T \mathbf{a}_{ji} & \text{for } l = j \\ 0 & \text{otherwise} \end{cases} .$$

With these definitions, the m equations (4) can be collected into the following bilinear system:

$$\mathbf{b} = A(\mathbf{t})\mathbf{d} . \tag{6}$$

If $\mathbf{t}, \mathbf{d}$ is a solution to equation (6), so is $c\mathbf{t}, \mathbf{d}/c$ for any nonzero c, consistently with the fact that absolute scale cannot be recovered from images alone [16]. With the additional constraint that the translation $\mathbf{t}$ have unit norm, $\mathbf{t}$ is the viewer's direction of heading.

5 Solving for the direction of heading

In the presence of noise, equation (6) will only be satisfied approximately, so we need a measure for how close $\mathbf{b}$ is to the column space of $(A(\mathbf{t}))$. A measure for this distance is obtained by replacing $\mathbf{d}$ in equation (6) by its solution $\mathbf{d}^+$ in terms of the pseudoinverse,

$$\mathbf{d}^+ = (A(\mathbf{t})^T A(\mathbf{t}))^{-1} A(\mathbf{t})^T \mathbf{b}$$

and then measuring the residual

$$\rho(\mathbf{t}) = |A(\mathbf{t})\mathbf{d}^{+} - \mathbf{b}| \,. \tag{7}$$

This residual does not depend on $\mathbf{d}$, and can be minimized with respect to $\mathbf{t}$. Efficient algorithms for computing (7) are given in [5]. The minimization problem can be solved by variable projection methods [4] [15] on the unit sphere $|\mathbf{t}| = 1$. Local minima do exist, but the more points are available in the field of view, the smoother the residual (7) turns out to be. In our experiments, we usually find one or two local minima, with the correct minimum considerably deeper than the other (see sections 6 and 7 for examples). One can then use local minimization methods starting at a few random points on the unit sphere and choose the convergence point of smallest residual as the solution. We are studying minimization methods that give hard convergence guarantees.

6 A simulation experiment

Figure 2 shows a contour plot of the residual $\rho(\mathbf{t})$ on the hemisphere of heading directions $\mathbf{t}$ corresponding to a forward moving viewer. Because $A(\mathbf{t})\mathbf{d} = A(-\mathbf{t})(-\mathbf{d})$ (see equation (6)), the residual function is the same on the opposite hemisphere ($\rho(\mathbf{t}) = \rho(-\mathbf{t})$). For this simulation we use thirty feature points, spanning a visual angle of about 120 degrees and distributed in depth between one and ten units away from the camera. Absolute depths are irrelevant because scale does not influence the results. The camera translation is one hundredth of the average distance to the scene, and the camera rotation is 3 degrees around a vertical axis. Random uniform noise is added to the second image. The width of the distribution is half a pixel for a 500×500 pixel image.

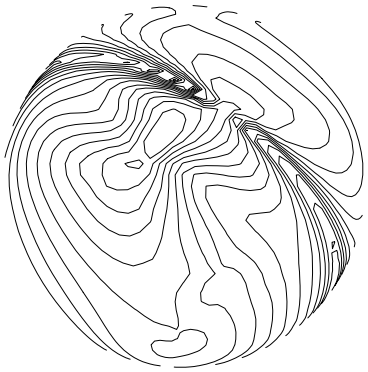

Figure 2: Contour plot of the residual $\rho(\mathbf{t})$ on the forward hemisphere. The two loops slightly above and to the left of the center are minima; the bigger loop is around the global minimum. Other loops are maxima.

The true direction, randomly generated, was the unit vector $(-0.306, -0.066, 0.950)^T$. Our method computed $\mathbf{t} = (-0.302, -0.126, 0.945)^T$, corresponding to a heading error of 3.45 degrees.

Figure 3 shows the direction error versus feature position uncertainty for the same situation. Each point on the graph is the average heading error for ten runs with the same noise distribution but different noise samples. For increasing errors the results become more and more erratic. However, the algorithm never fails completely, but degrades gracefully.

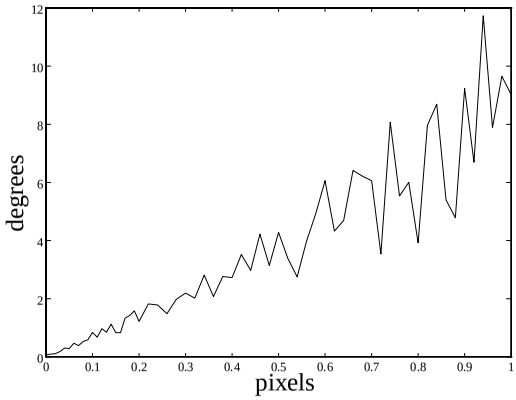

Figure 3: Direction of heading error versus feature position uncertainty.

Figure 4 compares our deformation-based residual,

$$|b - \mathbf{t}^T A\mathbf{d}|$$

(see equation (4)) with the more traditional residual based on optical flow,

$$|\mathbf{u} - (\delta A_1 \mathbf{t} + A_2 \omega)|$$

where $\mathbf{u}$ is the optical flow, ω is a vector representing the camera rotation, A_1 and A_2 are matrices that depend only on image position, δ is inverse depth along the optical axis, and $\mathbf{t}$ is the direction of heading (see for instance [6]). In both cases, the overall residual is the root mean square of the residuals for the individual features. All plots were obtained for a viewer translating exactly forward and no image noise. The two top diagrams show the residuals for pure translation: the widths and shapes of the two minima are essentially the same. When the camera rotates even by just one degree, the minimum of the flow-based residual (bottom right plot) becomes elongated and

shallow, leading to a more noise-sensitive minimization. The deformation residual, on the other hand, remains unaltered (bottom left).

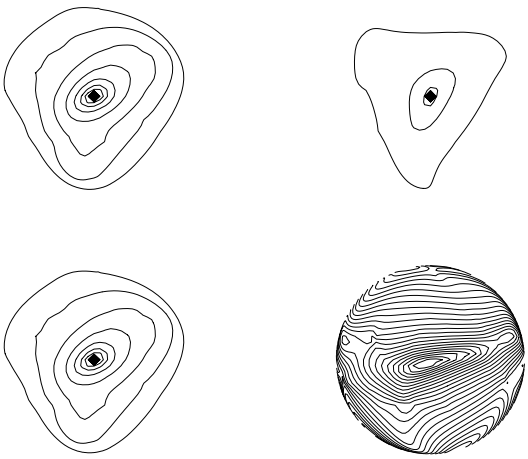

Figure 4: Contour plots of the deformation-based residual $\rho(\mathbf{t})$ (left plots) and the traditional flow-based residual (right) for pure translation (top) and with an added rotation of one degree (bottom).

To summarize, deformations are preferable to direct optical flow when the viewer rotates by even a small amount. On the other hand, regardless of whether flow or deformations are used, figure 3 shows that recovering the direction of heading from a pair of images requires cameras and tracking systems of good quality. In the next section we show that our method can be used with real images.

7 An experiment on real images

Figure 5 shows the first of a sequence of images taken in our lab, with the Delaunay triangulation of the tracked features superimposed. The viewing angle is about thirty degrees and the objects in the scene are between about 50 and 100 cm away.

About 230 features were automatically selected in the first frame and tracked using the algorithm described in [17]. Of those, 44 were handpicked to provide a roughly uniform distribution over the image. The camera was moved by a Puma arm proceeding in small steps, first along a constant reference direction (roughly towards the pencils), then along a direction at an angle of 30 degrees from the reference direction. For every new frame, features were tracked and the root mean square image deformation from start to current frame was determined. As soon as the rms deformation exceeded one pixel, the direction of heading was computed from the deformations between the start frame and the current frame. The current frame then became the new start frame for the next measurement. This procedure guarantees that the deformations are substantially greater than the feature position uncertainty (about 0.1 pixels), leading to a reliable computation of the direction of heading. Three sufficiently distant images were obtained with this procedure, with the second image at the turning point of the camera path. The residual functions for the two image pairs are shown in figure 6.

Figure 5: The first frame used in the experiment, with the Delaunay triangulation of the selected features.

The error on the angle between the two directions of heading, as computed by our method, was about 8 degrees. Since we do not know the accuracy of the Puma arm for very small motions and our camera was not calibrated, this error is only a rough indication of the accuracy of our method. We are planning more accurate experiments. Each residual function has a clean and deep global minimum, even with the small motion (1cm) and narrow field of view (30 degrees) of our experiment. There is also one local minimum in each residual, but this is much more shallow and created no problem for our minimization procedure.

8 Conclusion

Computing the direction of heading from image deformations is an interesting alternative to using the image flow field directly, because it removes the effects of rotation right at the outset in a clearly understandable and straightforward way. The minimum of the residual function computed from deformations is

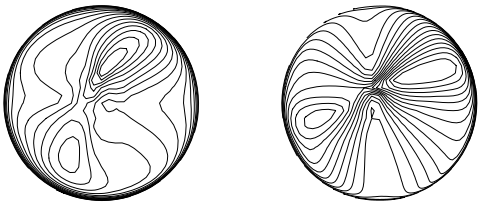

Figure 6: Contour plots of $\rho(\mathbf{t})$ on the positive hemisphere for the two real image pairs. Global minima are in the first quadrant, local minima in the third.

deeper than that of the flow-based residual, leading to a more reliable solution. Furthermore, the magnitudes of the deformations, when compared to the noise level in the images, are a direct indication of their reliability for the computation of the direction of heading. Finally, our minimization method degrades gracefully with feature position uncertainty.

Simulations confirm that a wide angle of view and accurate image measurements are necessary for good heading estimates. Our experiment with real images indicates that these requirements are realistic.

We have not yet fully explored the idea presented here, its weaknesses and strengths in comparison with competing methods, and its computational implications. On the contrary, we have left open several problems: how does the residual function behave as a function of point position, camera motion, camera calibration errors, and noise? Can the computation be made efficient enough to work in real time? Can multiple frames and incremental estimation techniques be used to improve the results over time? We are addressing these questions in our current research.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE PAMI*, 7:384–401, 1985.

[2] A. R. Bruss and B. K. P. Horn. Passive navigation. *CVGIP*, 21:3–20, 1983.

[3] J. E. Cutting. *Perception with an Eye Towards Motion.* MIT Press, 1986.

[4] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numerical Analysis*, 10(2):413-432, 1973.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, 1989.

[6] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *IJCV*, 7(2):95–118, 1992.

[7] H. Helmholtz. *Treatise on Physiological Optics.* Dover, 1925 (original published in German in 1896).

[8] E. C. Hildreth. Recovering heading for visually-guided navigation. *Vision Research*, 32(6):1177–1192, 1992.

[9] J. J. Koenderink and A. J. Van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791, 1975.

[10] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[11] H. C. Longuet-Higgins and K. Pradzny. The interpretation of a moving retinal image. *Proc. R. Soc. London B*, 208:385–397, 1980.

[12] S. J. Maybank. *A Theoretical Study of Optical Flow.* PhD thesis, University of London, 1987.

[13] F P. Preparata and M. I. Shamos. *Computational Geometry.* Springer-Verlag, 1985.

[14] J. H. Rieger and D. T. Lawton. Processing differential image motion. *JOSA*, A(2):254–360, 1985.

[15] A. Ruhe and P. Å. Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22(3):318-337, 1980.

[16] E. H. Thompson. A rational algebraic formulation of the problem of relative orientation. *Photogrammetric Record*, 3(14):152–159, 1959.

[17] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 3. detection and tracking of point features. Tech. Rep. CMU-CS-91-132, Carnegie Mellon, 1991.

[18] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE PAMI*, 6(1):13–27, 1984.

[19] A. Verri, F. Girosi, and V. Torre. Mathematical properties of the two-dimensional motion field: from singular points to motion parameters. *JOSA A*, 6(5):698–712, 1989.

[20] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao. Closed-form solution to image flow equations for 3d structure and motion. *IJCV*, 1:239–258, 1987.

[21] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE PAMI*, 11(5):451–476, 1989.