UC-405

LBL-29654

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

Submitted to Computer Physics Communications

Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition

M. Gyulassy and M. Harlander

October 1990

For Reference

Not to be taken from this room

LBL-29654 Copy 1 Bldg. 50 Library.

Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition*

Miklos Gyulassy and Magnus Harlander[1]

Nuclear Science Division
Lawrence Berkeley Laboratory
Berkeley, CA 94720 USA

Abstract:
A new Elastic Tracking (ET) algorithm is proposed for finding tracks in very high multiplicity and noisy environments. It is based on a dynamical reinterpretation and generalization of the Radon transform and is related to elastic net algorithms for geometrical optimization. ET performs an adaptive nonlinear fit to noisy data with a variable number of tracks and is more efficient numerically than the traditional Radon or Hough transform method because it avoids binning of phase space and the costly search for valid minima. Spurious local minima are avoided in ET by introducing an iteration time dependent effective potential. The method is shown to be very robust to noise and measurement error and extends tracking capabilities to much higher track densities than possible via local road finding or even the novel Denby-Peterson (DP) neural network tracking algorithms. A possible neural network implementation of ET is also discussed.

PACS: 6.50.-x, 29.80.-j, 89.90.+c

1 Introduction

Detecting curves within a complex pattern of points is a classic problem in pattern recognition and computer vision with many important practical applications[1]. In the context of high energy and nuclear physics, a common problem is the detection of the ionization paths of many charged particles in a device such as a bubble, streamer, or time projection chamber and the determination of particle momenta by fitting physical trajectories consistent with known electric and magnetic fields (see

*This work was supported by the Director, Office of Energy Research, Division of Nuclear Physics of the Office of High Energy and Nuclear Physics of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

[1]Permanent address: Technische Univ. München, Physics Department T30, James Franck Strasse 1, 8046 Garching, Germany.

e.g. [2]). Of course, many local and global methods have been developed to solve this problem[1, 3, 4]. Thusfar, most experiments had to cope only with rather low multiplicities and track densities, and conventional tracking methods have proven adequate. However, there is a need to develop more powerful methods to cope with the increasingly complex pattern recognition tasks that future high energy and nuclear experiments may face. For example, future heavy ion experiments[5] at RHIC/BNL and LHC/CERN could be confronted with trying to track up to 10^4 charged particles per event. High density tracking in noisy environments could also be of interest for identifying close secondary vertices in future B factory and SSC experiments. Clearly the design (size, segmentation, and cost) of detectors for tracking is not only limited by the available technology but also by the availability of tracking algorithms capable of extracting the desired physics information in the presence of inevitable noise and measurement error. Our aim here is to explore theoretical limitations of present tracking methods and to propose a new elastic tracking (ET) method that extends present tracking capabilities to much higher track densities. By track density, ρ_{track}, we mean here the average ratio of the distance between measured points along a track to the distance between points belonging to different tracks or random noise points.

This work is also motivated by the pioneering work of Denby[6] and Peterson[7] on applications of Hopfield type neural networks[8, 9] to tracking and other pattern recognition tasks in high energy physics (see also [10]). Neural network methods offer the advantage of highly parallel and competitive computational capabilities and of potential hardware implementation via analog VLSI technology[11]. Since the performance of such neural network tracking methods has not yet been compared to that of more conventional algorithms, one aim of the present work is to carry out such a comparison. In particular, we will use as a benchmark the conventional local Road Finder (RF) algorithm[4]. For $\rho_{track} \lesssim 1$ both tracking methods are found to perform equally well. For $1 \lesssim \rho_{track} \lesssim 4$ we show that the DP net performs significantly better than the RF method. However, we also show that the DP net performance rapidly deteriorates beyond $\rho_{track} \gtrsim 5$. Therefore, for tracking problems at higher densities even more powerful methods are required.

The ET method proposed here combines the dynamical approach of elastic net algorithms[12] to solve complex geometrical optimization problems with the classic global tracking methods based on the Radon or Hough transform[1]. ET is based on a reinterpretation and generalization of the Radon transform[1] as an effective interaction energy of an elastic line of charge with the measured ionization charge density. The tracking and fitting are performed simultaneously by solving the equations of motion of the analogous dynamical system. It differs from elastic nets[12] in that nodes on a track move coherently on a set of trajectories constrained by external physical equations of motion and in that ET dynamics involves iteration time dependent interactions to avoid problems with local minima. It also eliminates in principle the need for extensive preprocessing of the input data as it can deal directly with a continuous charge distribution.

An example of a typical "hard" tracking problem that we consider in this paper is shown in Fig. 1. In this example, the problem is to identify the 15 straight-line tracks in the upper left panel from the "measured" data points on the right upper

panel. The noisy "detector" consists of 16 equidistant rows. Each row introduces an equal number of random noise points as there are real track intersection points. The points are measured with a rms relative error of 3%. The median track density in this example is around $\rho_{track} \approx 7$. For comparison the typical track density expected in the HISS/TPC heavy ion experiment[2] is more than an order of magnitude smaller. In the left bottom panel the solution of the DP net algorithm is shown, while on the right the solution of the ET algorithm is shown. In this example the neural net got very confused, while ET resolved all the correct tracks.

The discussion is organized into the following sections: In section 2, we review the Radon/Hough transform tracking method. Section 3 introduces the elastic tracking method. In section 3.1 the effective track interaction energy and track dynamics are defined. In section 3.2 we compare the adaptive nonlinear fitting performed by ET to conventional χ^2 fitting. In section 3.3, we discuss how ET dynamics avoids getting caught in local minima of the interaction energy. Section 3.4 addresses the problem of multiple tracking from the point of view of parallel vs sequential algorithms. Section 3.5 summarizes the advantages and limitations of the ET method. In section 4 we turn to neural network tracking methods and review the Denby-Peterson net in section 4.1. In 4.2 we discuss how the ET algorithm could be implemented in principle on a Hopfield type network. In section 5 we review the conventional local road finder algorithm used as a benchmark for our numerical simulations. In section 6.1 we describe our model detector and define several local and global measures of performance used in our simulations. In 6.2 the performance of the road finder, DP net, and ET algorithms are compared for finding helices in 3D at low track densities ($\rho_{track} \lesssim 1$). To compare algorithms at much higher track densities, we turn in section 6.3 to a comparison of tracking methods for finding many straight lines in 2D. An outlook is presented in section 7. Appendix A provides a brief description of the code developed to implement the algorithms.

2 Radon Transform Tracking Method

The classical Radon transform of a density, $\rho(\mathbf{x})$, is defined by line integrals over a special class of trajectories[1]

$$R(\mathbf{x}, \mathbf{p}) = \int d\tau \rho(\mathbf{r_p}(\tau) + \mathbf{x}) \ , \tag{1}$$

where $\mathbf{r_p}(\tau)$ is the trajectory specified by parameters $\mathbf{p}$ such that $\mathbf{r_p}(0) = 0$, and $\mathbf{x}$ is a point on the track. For tracking problems in a homogeneous magnetic field $\mathbf{r_p}(\tau)$ describes a simple helix, while in a field free region $\mathbf{r_p}(\tau) \propto \mathbf{p}\tau$ would be a straight line pointing in the direction of $\mathbf{p}$. Its inverse is easily obtained by a Fourier transform in the variable $\mathbf{x}$. For tracking applications the phase space is discretized and the Radon transform converts into the Hough transform[1]. If the output of each pixel is restricted to binary values, the Hough transform simply counts the number of pixels which are on and which intersect the trajectory.

Curve detection then involves scanning the phase space bins for maxima or ridges. By summing over the feature space pixels, $\mathbf{x}_i$, the accumulated sum has isolated

maxima in momentum space. This method is thus basically a template matching method. The templates in this case correspond to the family of curves specified by $\mathbf{r_p}(\tau) + \mathbf{x}$.

The most obvious disadvantage of this method of track detection is that its numerical cost increases very quickly with the desired resolution, $(\delta x, \delta p)$, and the dimension d of the feature space. Given a range of typical momenta ΔP and a range of coordinates ΔX in each dimension, $(\Delta X \Delta P / \delta x \delta p)^d$ line integrals must be computed followed by a lengthy search for maxima. In addition, care must be taken to avoid spurious local maxima associated with accidental alignment of points on different tracks.

For straight line tracking in 2D at low track densities, VLSI hardware devices have been developed to implement this method at least on a coarse grid in phase space (see review and references in Ref.[1]). However, for higher dimensions, high track densities, and nonlinear trajectories this straight forward application of the Radon transform is impractical. To overcome the numerical limitation of having to bin phase space while retaining the pattern recognition power derived from the global, long range information provided by the line integral, we introduce in the next section the ET method for tracking.

3 Elastic Tracking

3.1 Track Energy and Dynamics

Our starting point is a reinterpretation and generalization of the Radon transform (1) from the point of view of dynamical systems. We interpret R as the interaction energy of the external measured (negative) charge density, $\rho(\mathbf{x})$, with the (positive) charge density of a track, T, defined by

$$\rho_T(\mathbf{x}) = \int d\tau \delta(\mathbf{x} - \mathbf{r}_{\mathbf{p}_T}(\tau) - \mathbf{x}_T) \ , \tag{2}$$

By restricting the range of the integral to $\tau \geq 0$, $\mathbf{x}_T$ can be interpreted as the origin of the trajectory of a test particle produced with initial momentum $\mathbf{p}_T$. The effective potential in the case of the Radon transform is just a zero range delta function.

Generalizing to a finite range and possibly *iteration* time dependent potential, $V(\mathbf{x}, t)$, the track interaction energy is defined by

$$R_V(\mathbf{x}_T, \mathbf{p}_T, t) = \int dx dx' \rho(\mathbf{x}) V(\mathbf{x} - \mathbf{x}', t) \rho_T(\mathbf{x}') \ . \tag{3}$$

In the limit $V(\mathbf{x}, t) = \delta(\mathbf{x})$, eq.(3) reduces to the ordinary Radon transform. For a static square well potential R_V reduces essentially to the Hough transform. As with the Radon transform the inverse of R_V is easily computed via a Fourier transform in the variable $\mathbf{x}_T$.

By assigning a negative charge to the ionization charge density, ρ, and a positive charge to the track density, ρ_T, we convert the tracking problem into a problem of finding minima of the effective track interaction energy. The standard gradient descent

method provides a means to find such minima by solving the following dynamical equation:

$$\begin{aligned} d\mathbf{p}_T/dt &= -\eta \nabla_{\mathbf{p}_T} R_V(\mathbf{x}_T, \mathbf{p}_T, t) \ , \\ d\mathbf{x}_T/dt &= -\eta \nabla_{\mathbf{x}_T} R_V(\mathbf{x}_T, \mathbf{p}_T, t) \ , \end{aligned} \tag{4}$$

where η is a small rate parameter. Starting from a random (or more educated) initial guess for $(\mathbf{x}_T, \mathbf{p}_T)$, the above equations evolve the phase space point specifying the trajectory into a local minimum of the interaction energy.

There is of course considerable freedom in the choice of the effective potential. We adopt for convenience a simple Lorentzian form,

$$V(x,t) = w^2(t)/(x^2 + w^2(t)) \ , \tag{5}$$

with a time dependent range

$$w(t) = b + (a-b)\exp(-t/c) \ . \tag{6}$$

A slow iteration time dependence of the range is introduced in order to avoid getting caught in local spurious minima as explained in section 3.3. The natural scale for the asymptotic range is $b \sim \Delta x$, where Δx is the rms measurement error of the ionization coordinates as we shall see in the next section.

3.2 ET as Adaptive Nonlinear Fitting

The dynamical equations (4) leads to an adaptive nonlinear fit of one track to the data. To see the relation between that fit and the conventional χ^2 fit, consider tracking by a detector composed of N detector rows or planes such as in Fig. 1. Let $\mathbf{x}_{ij}$ denote the centroid of the ionization charge measured in detector i for track j with $1 \le j \le M$. Here M is the multiplicity of tracks entering the detector. In addition spurious noise points, $\mathbf{x}_{ij}$ with $M+1 \le j \le M_i \equiv M + \tilde{M}_i$, are introduced by each imperfect detector i. The measured (negative) charge density can then be approximated by

$$\rho(x) = -\sum_{i=1}^{N}\sum_{j=1}^{M_i} \theta_{ij}\delta(\mathbf{x} - \mathbf{x}_{ij}) \ , \tag{7}$$

where for $j \le M$, $\theta_{ij} = 1$ if track j intersects detector i and zero otherwise, and for noise points $(M < j \le M_i)$ $\theta_{ij} = 1$.

Let $\mathbf{r}_i(\mathbf{x}, \mathbf{p})$ denote the intersection point of the trajectory T specified by $(\mathbf{x}, \mathbf{p})$ with detector i. The (positive) trajectory charge density is then

$$\rho_T(\mathbf{x}') = \sum_i \theta_{iT}\delta(\mathbf{x}' - \mathbf{r}_i(\mathbf{x}, \mathbf{p})) \ , \tag{8}$$

where $\theta_{iT} = 1$ if trajectory T intersects detector i and zero otherwise. In this case, the (negative) track interaction energy (3) is simply given by

$$R_V(\mathbf{x}, \mathbf{p}, t) = -\sum_{i=1}^{N}\sum_{j=1}^{M_i} \theta_{ij}\theta_{iT} \frac{w^2(t)}{(\mathbf{x}_{ij} - \mathbf{r}_i(\mathbf{x}, \mathbf{p}))^2 + w^2(t)} \ . \tag{9}$$

For well separated tracks and noise points, in the sense that $(\mathbf{x}_{im} - \mathbf{x}_{in})^2 \gg \Delta x^2$ for all pairs $m \neq n$, only one term, say J, in the sum over j will survive in the limit that the elastic trajectory is close to track J. In the very idealized case, with $|\mathbf{x}_{iJ} - \mathbf{r}_i(\mathbf{x}, \mathbf{p})| \ll \Delta x$, the track interaction energy reduces to

$$R_V(\mathbf{x}, \mathbf{p}, t) \approx \sum_{i=1}^{N} \theta_{ij}\theta_{iT}(-1 + (\mathbf{x}_{iJ} - \mathbf{r}_i(\mathbf{x}, \mathbf{p}))^2/\Delta x^2) \tag{10}$$

when the Lorentzian width is taken to be the rms measurement error Δx. Eq.(4) then insures that the sum of squares of the tracking errors becomes a minimum with respect to variations in $(\mathbf{x}, \mathbf{p})$. We see that in this idealized case the ET fit is similar to the ordinary χ^2 fit. However, in the usual case, this limit does not apply and the nonlinearity of the Lorentzian cannot be neglected.

The nonlinearity in eq.(9) is essential and acts as a selective filter that reduces greatly the sensitivity of the fit to outlier points, such as those from other tracks and noise. Furthermore, unlike conventional χ^2 fits, the ET fit is adaptive in the sense that it does not work with a fixed number of points but rather selects a subset of points leading to a good fit. Note also that for a single track with N measured points with true Gaussian error, $\langle R_V \rangle \approx -0.758N$ for a perfect fit with $b = \Delta x$, whereas the expectation of the right hand side in (10) vanishes. This again emphasizes the difference between χ^2 and the ET fitting method.

3.3 Local vs Global Minima

As with any optimization or minimization problem care must be taken to ensure that the system does not get caught in a spurious local minimum. One of the classic ways to avoid this problem is to couple the system to a heat bath with a temperature large enough so that the system can fluctuate out of shallow minima. This is basically the strategy used by the neural network algorithm[7] discussed in section 4. For elastic tracking, however, there is another natural approach to this problem.

Note that for $w \to \infty$, $R_V \to$ constant independent of $(\mathbf{x}, \mathbf{p})$. Therefore all minima are washed out in the long range limit. This suggests that during the initial evolution of the trajectory, the range $w(0) = a$ should be taken significantly larger than the final range set by $b \approx \Delta x$. In addition the range relaxation time, c, in (6), should be larger than the mean convergence time of the algorithm so that the system has time to pass over the spurious local minima that emerge when w is small.

This method of finding the true track minima is illustrated in Fig. 2. We consider here the energy surface for a 2D problem with 5 straight-line tracks. Two of the tracks intersect the origin with angles $\theta = 26.6$ and 45 degrees relative to the x axis. The other three lines are randomly oriented and do not pass through the origin. The interaction energy $R_V(0, \theta)$ is plotted as a function of θ for different ranges $w = 0.1, 0.01, 0.001$. For very small ranges, two deep, narrow minima are seen near the correct angles. However, many shallow spurious minima from accidental alignment points from different tracks can also be seen. If we start at some random guess for θ, then there is a high probability that the system evolves into one of the spurious minima.

On the other hand, choosing a much longer range, $w = 0.1$, washes out the spurious minima and only one deep and very broad minimum near $\theta \approx 30$ remains. For this width, any initial guess for θ will converge to that global minimum. If after convergence within some tolerance, $|d\mathbf{p}_T/dt|\Delta t < \delta p, |d\mathbf{x}_T/dt|\Delta t < \delta x$, the width is slowly decreased to 0.01 the trajectory will begin to slide down on one of the sides of the hill that emerges between the true minima as the width gets smaller. As the width is decreased even further, the true energy surface becomes better and better resolved. By varying the width slowly enough most initial guesses will move in the evolving energy surface into one of the valid deep minima. In this way, ET dynamics with a time dependent width (6) can often avoid getting trapped by spurious minima.

As a check on whether the elastic track has converged to a true track, the magnitude of the track interaction energy is also monitored. For a valid track, a good fit will give a value of R_V close to $-0.75N_T$ for $b = \Delta x$ and Gaussian errors as noted above, where $N_T = \sum_i \theta_{iT}$ is the expected number intersections of the trajectory with the elements of the detector. In our actual numerical simulation, we check that the Hough like transform for the trajectory obtained by replacing the Lorentzian by $V(x) = \theta(3\Delta x - x)$ is close to $-N_T$. For that transform however only one point is counted per detector row. Other points within $3\Delta x$ in the same row are ignored.

3.4 Multiple Elastic Tracking

Once convergence is achieved for one track, another one could be looked for by starting with a new random initial template. and waiting until a new valid minimum is found. However, convergence for new tracks can be speeded up by introducing multiple elastic track dynamics. Since two trajectories should not converge to the same track, an effective repulsive interaction between trajectories should be introduced. The obvious generalization for the interaction energy between trajectories T_i and T_j is

$$R_V(T_i, T_j, t) = \int dx dx' \rho_{T_i}(\mathbf{x}) V(\mathbf{x} - \mathbf{x}', t) \rho_{T_j}(\mathbf{x}') \ . \tag{11}$$

The total interaction energy of the system is then

$$E(t) = \sum_i R_V(T_i, t) + \tfrac{1}{2} \sum_{i \neq j} R_V(T_i, T_j, t) \ , \tag{12}$$

where the negative interaction energy of a trajectory with the measured ionization density is denoted by $R_V(T_i, t) = R_V(\mathbf{x}_{T_i}, \mathbf{p}_{T_i}, t)$ given by eq.(3).

There are at least two ways to implement the multiple track dynamics: in parallel or sequentially. The parallel method involves the usual many body dynamics in which all trajectories move simultaneously according to the obvious generalization of eq.(4)

$$\begin{aligned} d\mathbf{p}_{T_i}/dt &= -\eta \nabla_{\mathbf{p}_{T_i}} E(t) \ , \\ d\mathbf{x}_{T_i}/dt &= -\eta \nabla_{\mathbf{x}_{T_i}} E(t) \ . \end{aligned} \tag{13}$$

We refer to (13) as competitive elastic dynamics, since the trajectories move not only in the field of the external charge density but are repelled by each other as they

evolve. One numerical difficulty of this method, however, is that it could be sensitive to initial conditions.

To avoid that problem we chose instead a sequential algorithm that is well suited to numerical simulation on sequential digital computers. In this method only one trajectory is evolved at a time to convergence. Subsequent trajectories are evolved according to the external field screened by the additional (static) charge densities of previously found trajectories. In this approach then, the jth trajectory evolves according to the screened interaction energy

$$E_j(t) = \int dx dx' (\rho(\mathbf{x}) + \rho_{screen}(\mathbf{x})) V(\mathbf{x} - \mathbf{x}', t) \rho_{T_j}(\mathbf{x}') \ , \tag{14}$$

where the screening density is given by

$$\rho_{screen}(\mathbf{x}) = \sum_{i=1}^{j-1} \rho_{T_i}(x) \ . \tag{15}$$

Because of screening, it becomes easier for subsequent trajectories to converge to new tracks. One advantage of this approach is that many different initial conditions can be tried for each new trajectory, thereby reducing the sensitivity to initial conditions. Typically we try up to $N_{stop} \sim 100$ random initial conditions for each new track. If no convergence is achieved for N_{stop} trials, it is assumed that no more tracks can be found and the algorithm is terminated.

We adopted the sequential method in our numerical simulations, and screening was implemented by simply removing the external charge point closest to the found trajectories within a radius $3\Delta x$ on each detector. The numerical implementation of this ET algorithm could be further improved by making more educated initial condition guesses and terminating the search at an earlier stage where the remaining charge density is sufficiently low that simpler and cheaper methods, such as the road finder discussed in section 5, perform adequately. We will however not consider such refinements further here.

3.5 Remarks

ET is a dynamical generalization of the Radon transform which solves both the phase space discretization problem and track recognition minimization problem. It can deal in principle with the continuous ionization charge data, $\rho(x)$ and performs track fitting at the same time. It is important to emphasize again, however, that ET can only be applied to tracking problems where trajectories obey known physical equations of motion. Therefore either the phase space dependence of the trajectory $\mathbf{r}_{\mathbf{p}_T}(\tau) + \mathbf{x}_T$ must be known analytically, as in the case of straight lines or helices, or at least it must be computable numerically from known equations of motion.

This method is therefore less general than for example the local road finding algorithm[4] or the Denby-Peterson (DP) net approach[6, 7] discussed in the two next sections, which can find tracks of arbitrary shape at sufficient low track densities. However, when the class of allowed trajectories is known, ET is bound to be more powerful at high ρ_{track} because it takes into account that information.

4 Neural Network Tracking

4.1 Denby-Peterson Net

For general tracking problems, Denby[6] and Peterson[7] proposed a novel application of Hopfield neural networks[8]. It assumes that ionization density can be approximated by $\rho(x) = \sum_i \delta(\mathbf{x} - \mathbf{x}_i)$. The DP net associates neurons with all possible track segments, $\vec{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, between measured points. The output, s_{ij}, of each neuron varies in the range $[0, 1]$. A valid track segment is signified by a value of s_{ij} close to one, whereas a value close to zero signifies that $\vec{r}_{ij}$ is an unlikely track segment. From a random initial configuration for s_{ij}, the network dynamics evolves the neurons to a state minimizing the network free energy.

The energy or cost function for the network is assumed to be of the standard Hopfield form[8]

$$E = \tfrac{1}{2} \sum_{ijk} s_{ij} s_{jk} W_{ijk} \quad , \tag{16}$$

where W_{ijk} is the interaction energy or cost associated with connecting segments $\vec{r}_{ij}$ and $\vec{r}_{jk}$. The network dynamics evolves the net to a solution of the mean field equations[7]

$$s_{ij} = \tfrac{1}{2}(1 + \tanh\left(-\frac{1}{T}\frac{\partial E}{\partial s_{ij}}\right)) \quad , \tag{17}$$

where T is a temperature related formally to the nonlinear sigmoid response (tanh) of a neuron. The temperature is a crucial parameter of the algorithm and controls the ability of the network to avoid getting caught in local minima of its free energy. Choosing it too small leads to "premature crystalization" into local minima. Choosing T too high leads to random results since all states are equally likely. The magnitude of T must be adjusted near the mean single particle energy, $\langle \partial E / \partial s_{ij} \rangle$.

The choice of W_{ijk} proposed by Denby and Peterson was a *local* function of $\mathbf{x}_i, \mathbf{x}_j$, and $\mathbf{x}_k$ that penalizes connections with large angles (kinky tracks) and also those with long track segments. The specific parameterization that we adopted for convenience was

$$W_{ijk} = (A - (A + B) cos^{2n}(\theta_{ijk}/2))/(r_{ij} + r_{jk}) \quad , \tag{18}$$

where we took $A = 4, B = 0.5$, and $n = 16$. Here θ_{ijk} is the angle between segments $\vec{r}_{ij}$ and $\vec{r}_{jk}$. By choosing $A \gg B$, only those connections are favored that correspond to small changes in the direction of the track. Kinks are heavily penalized including those unnatural connections that look like two segments coming out or flowing into the same point. Therefore, the global constraints in refs.[6, 7] need not be explicitly added with this choice. The form of our weight function is illustrated in Fig. 3.

In practice, we solve eq.(17) by iteration with an asynchronous update algorithm. An initial random guess for the s_{ij} is made. The right hand side of (17) is computed for one of the neurons with $\mathbf{x}_i$ furthest from the interaction region (on row N of our model detector and $\mathbf{x}_j$ on row $N - 1$). Taking into account the new value of the output of that neuron, a second neuron with $\mathbf{x}_i$ on the furthest row is updated. When all neurons beginning on row N are updated, the neurons beginning on rows

$N-1$ are updated and so on until the whole network is swept over. What makes this algorithm asynchronous of course is that information gained at each single neuron update is propagated to the next neuron. In contrast in synchronous dynamics all neurons are updated independently at the same time step. The asynchronous method here is similar to the sequential ET dynamics discussed in section 3.4. This iterative network algorithm is continued until the maximum change of the output signal of all neurons is less than 0.01. At that time all neurons with $s_{ij} > 0.5$ are identified as valid track segments and linked to form the identified tracks. The connected lines in Fig. 1 show the network solution for our hard (high ρ_{track}) example.

As noted before, the real power of this method is its ability to find arbitrary curves. Its limitation to moderate track densities only stems from its neglect of important information contained in long range correlations in cases where the class of allowed trajectories is known a priori. We show in the next section, however, that a Hopfield type network could be constructed which makes full use of such additional information.

4.2 ET Net

The steepest descent dynamics proposed for ET in section 2 is of course only one way to perform the minimization task. We now show that it is possible to map the ET method itself onto a Hopfield net. An extension of the DP net that incorporates the long range correlations of ET can be made simply by amplifying the DP weight, W_{ijk}^{DP}, in eq.(18) with the value of the line integral eq.(3). This can be achieved for example by taking

$$W_{ijk} = W_{ijk}^{DP} \sum_n{}' V(\mathbf{x}_n - \mathbf{x}_{ijk}(n)) \ , \tag{19}$$

where $\mathbf{x}_{ijk}(n)$ is the point nearest to $\mathbf{x}_n$ that lies on a trajectory fitted through the sequence of points $\mathbf{x}_i, \mathbf{x}_j$ and $\mathbf{x}_k$

The prime on the sum indicates that the sum does not have to be taken over all the points in the detector, but rather could be restricted to a finite domain near $\mathbf{x}_j$. In effect, this involves evaluating the ET line integral over a finite range whose scale could be adjusted to match the complexity of the problem. In the limit that this range includes only points i, j, k, W would reduce effectively to the local DP weight. By including additional points on rows above $\mathbf{x}_i$ and below $\mathbf{x}_k$ more information on the long range correlations can be incorporated into the net dynamics. The numerical cost of this extension increases linearly with the range of the line integral and is proportional to N^3 as for the DP net. In contrast, the steepest descent ET method involves typically calculating only ~ 100 interaction integrals ($\propto N$) for each of the M trajectories.

5 Road Finding

The simplest and numerically fastest tracking method that provides a fair benchmark for comparing tracking algorithms is the local Road Finding (RF) algorithm[4]. RF

is basically an intuitive "follow your nose " method. To describe this algorithm, we consider as in section 3.2 the problem of tracking in a TPC type detector[2] with N rows ordered with increasing distance from the interaction vertex. In each row, i, the intersection points, $\mathbf{x}_{ij}$, for tracks, j with $1 \leq j \leq M$, are measured. In addition, we again assume that spurious random noise points, $\mathbf{x}_{ij}$, with $M+1 \leq j \leq M_i$ are introduced by each detector.

The algorithm begins by looking for points on the three furthest detectors, $N, N-1, N-2$, that line up within some small initial tolerance angle $\Delta\theta_0$. That line defines a candidate track direction. A road or "nose" is defined by a cone of angular width, $\Delta\theta$ (the nose thickness), oriented in the direction of the candidate track and with a vertex on the point nearest the target. The road is projected several detectors forward to test for the presence of measured points within that cone. If no points are found, then the initial guess is rejected and another three points lining up on the furthest detectors are taken as an initial guess for a candidate track.

If one point is found within the projected nose, it is chained up with the first three and a new local orientation of the road is defined using this point. The search for new points in subsequent rows (within the nose thickness around the locally updated nose orientation) is continued until no further points are found. All the points found are then linked up and identified with one track.

New tracks are searched for by repeating this algorithm with new combinations of three collinear points. However, in subsequent road searches points already linked with previous tracks are ignored. Once all three-point collinear combinations beginning in row N have been tried, all three-point collinear combinations beginning on row $N-1$ are tried, and so on. Only tracks with at least four linked points are regarded as valid tracks or track segments.

For sufficiently low track densities, the nose thickness can always be adjusted so that all the tracks are correctly found. However, for high densities the accidental occurrence of collinear points from different tracks and noise points is certain to confuse this algorithm. In fact, an important strategy of this algorithm is to reject all points within the search road if there is more than one on a given detector road. In that way local crossing points and noisy confusion points are simply skipped over. As long as the confusion regions occupy a small region of the feature space, the projection of the road can be extended far enough that additional track points can be found further on down the road where the track density is low again.

While the above strategy solves the local confusion problem, it sometimes segments tracks into multiple "tracklets", i.e. tracks that are disconnected segments of the same track. The same problem also arises in the DP net tracking method. Thus in both those methods post-processing is required to try to link disconnected tracklets. Also since both methods only link the coordinate space points, they require an additional post-processing fitting stage, where each tracklet is fit to a trajectory specified by parameters $\mathbf{p}$. The necessity of a pre-processing stage to define the location, $\mathbf{x}_{ij}$, of the maxima of the measured charge distributions and two post-processing stages introduces of course additional sources of tracking errors.

The simplest reconnection criterion is one linking tracklet i and j with each other

if

$$|\mathbf{p}_i - \mathbf{p}_j| < \Delta p_{join} \quad . \tag{20}$$

While this works in many cases of moderate densities, such a relinking criterion certainly will distort for example the physically interesting two track correlation function, $C(\mathbf{p}_1, \mathbf{p}_2)$, on the scale Δp_{join}. On the other hand, choosing p_{join} too small will lead to an overestimate of the track multiplicity distribution. The ET method is immune to this segmentation problem because it deals only with complete trajectories.

Finally, we note that many variants and extensions of the road finder methods have been developed (see review in [4]). For example, a χ^2 fit of the previously linked points could serve to predict better the orientation and width of the road ahead. In addition, competition between adjacent roads could be implemented. For our benchmark studies we use only the simplest version of RF.

6 Numerical Simulations

6.1 Performance Measures

The performance of the tracking methods was tested with simulations of TPC like detectors in 2D and 3D. Our idealized rectangular detector covers a region $|x| < 1, |z| < 1$ and $|y| < 1.5$. The "beam" axis is taken as the y axis. The detector consists of N_y rows or planes spaced at regular intervals ($\Delta y = 3/N_y$) along the y axis. The detectors in each row are used to determine the mean (x, z) location of the ionization charge densities with an rms Gaussian error $\delta x = 0.03\Delta y$ to simulate detection errors similar to those expected in the Hiss/TPC[2] as an example. Random noise was introduced on each layer such that the number of noise points in each detector was taken to be a fixed fraction f of the number of true track intersections.

For a local measure of the tracking complexity we use the dimensionless local track density

$$\rho_{ij} = \Delta y / \min_{k \neq i}((x_{ij} - x_{ik})^2 + (z_{ij} - z_{ik})^2)^{1/2} \quad . \tag{21}$$

The mean track density, ρ_{track}, is defined here as the median of this local density. To evaluate the performance of the methods, we monitored four quantities:

1. The distribution of the number of valid track points found by the algorithm as a function of the local track density.

2. The fractional pixel error distribution, measuring the fraction of the identified points at a track density ρ_{ij} which were incorrect either because they actually belonged to different tracks or were noise points.

3. The relative rms error, $\Delta n/n$, of the track multiplicity found relative to the input multiplicity n, after tracklets with similar momenta were linked.

4. The rms momentum resolution, Δp, of the identified tracks, where all components of $\mathbf{p}$ are normalized between -1 and 1.

Note that the first two measures look at detailed local tracking performance, while the last two are measures of the final physics output performance. The third is sensitive to multiplicity errors, while the fourth is a measure of the overall quality of the fit to the points found.

6.2 3D Low Density Helices

An example of the tracking problem in a device such as discussed in [2] is shown in Fig. 4. The multiplicity, $M = 15$ and the number of rows $N = 32$ have been scaled down for numerical simplification in a way as to preserve approximately the distribution of track densities of the full scale problem with $M = 200$ and $N = 128$. The first 8 rows nearest the target were turned off to reduce the number of points at high density. A great simplification in this problem is that the interaction origin of all helices is assumed to be known and fixed at one point just outside the detector. Thus only the momenta and the multiplicity are variables.

The three momenta for the trajectories were chosen from a Gaussian distribution with widths taken to simulate the helix curvatures expected in the HISS/TPC[2]. A twenty percent noise fraction $f = 0.2$ was introduced for these simulated events.

The distribution of local track densities is shown in Fig. 5. Note that for this class of events and detector configuration the distribution of correctly identified points peaks below unit density indicating that these events are "easy". Indeed, all three methods correctly identify the vast majority of points. In the lower part of Fig. 5, we see that in fact the fractional pixel error is zero until densities > 2. Pixel errors begin to show up only in the physically insignificant far tails of the distribution beyond that density. While the statistics is very low, there appears to be a trend that the DP net performs better than the RF and ET methods in those tails. From a global perspective however RF is more than adequate for tracking such events. This problem is simply too easy.

6.3 2D High Density Lines

Unfortunately, to study more interesting problems with track densities approaching 10 requires in 3D an increase of the multiplicity by a factor of 100 or reduction of the number of tracking detector by a factor of ten. The first option was computationally prohibitive while the second leads to too few measurements per track. Instead, we chose the simpler route by reducing the dimension to two and the number of tracking detectors by a factor of two.

Sample events were generated by choosing M random straight lines constrained such that they enter the lower half of the detector, exit the upper half (row $N_y/2 - 1$ or above), and intersect the line $y = -1.5$ at $|x| < 2$. Given N_i measured points in row i, noise was simulated by adding $f \times N_i$ random points with $|x| < 1$ in each row. The noise to signal fraction, f, was varied between 0 and 1.

An example of a hard problem was already shown in Fig.1 corresponding to $M = 15$, $f = 1.0, N_y = 16$, and $\delta x = 0.03\Delta y$. In Fig. 6 examples of tracking problems with increasing complexity are shown for $f = 0.2$, corresponding to only 20% noise.

As the multiplicity increases from 3,5,10,15 the median track density increases from $\rho_{track} \approx 0.5, 1, 2, 3$. For multiplicity 3, all three methods find the correct tracks. Note, however, that the median density is similar in this case to the 3D helix problem shown in Figs. 4,5. Doubling the density to $\rho_{track} \approx 1$ for N=5 already leads to difficulties for the RF method since one of the real tracks is misidentified as two distinct tracks with different slopes. At N=10 and 15, the RF method performs poorly.

The DP net in these examples is able to resolve tracks correctly up to N=10 and only begins to get confused at $N = 15$. The competitive and reinforcement dynamics of the net make it possible for the net to "see" the global picture much better than the local RF method. However, as in Fig. 1 the neural net also gets confused when the track density gets too high. Recall that the DP net was not constructed to look for long range linear correlations among the points. We checked that the performance of the net could not be improved by variations of the network parameters (A, B, T).

The ET method on the other hand with its built in a priori knowledge of straight line trajectories has no difficulty finding the tracks in all the examples including the hard example in Fig. 1.

In Fig. 7 we compare more quantitatively the detailed performance of three tracking methods as a function of the local track density. Identical sets of 100 simulated events were used with all three methods to compare the performance at different N and f. In Fig. 7 results for $N = 10, 15$ and $f = 0.2$ are shown. The upper graphs show the distribution of points identified with valid tracks as a function of the local track density. In the lower graphs the fractional pixel misidentification error is shown.

The results demonstrate that both the DP net and ET methods perform better than RF at these moderately high densities. More points are picked up with fewer errors. For $N = 10$ with $\rho_{track} \approx 2$ there is virtually no difference between the DP and ET results. However, Fig. 7 reveals a confusion threshold of the DP net for multiplicities between 10 and 15 ($\rho_{track} \approx 2 - 3$). Both the pixel identification distribution and the fractional pixel error distribution for the DP net approaches that of the RF method at $N = 15$. At the same time, the performance of ET remains virtually unchanged as a function of multiplicity.

Finally, the global comparison of the methods is shown in Fig. 8. The performance is compared for $f = 0$ and 0.2 as a function of the median track density for $N = 3, 5, 10, 15, 20$. The solid points at the highest density in all panels correspond to conditions of the hard problem in Fig. 1 with 100% noise.

Without noise, the momentum resolution of the DP net and ET are similar, up to $\rho_{track} \approx 3$ where the segmentation problem causes the DP to overestimate the multiplicity more often than does ET. Already with 20% noise, however, both the momentum resolution and multiplicity error of DP becomes comparable to that of RF. Surprisingly, the multiplicity error of DP even exceeds that of RF at density 4. At the highest density studied, the RF method is useless, whereas the ET method still does a fair job of extracting the physics. Unfortunately, finite computer resources prevented us from accumulating statistics on the DP net under these extreme conditions, but examples such as in Fig. 1 together with the results for density $\rho_{track} \approx 4$ point to the need to incorporate information on long range correlation in order to avoid confusing the neural net.

7 Outlook

In summary, we have shown that the elastic tracking method extends tracking capabilities up to very high track densities. The power of the Radon transform when put into a dynamical context is the key to its success. ET is basically an Elastic *Template* method, where the templates have fuzzy edges. The fuzzy edges play a very important role in the dynamics by smoothing over local minima. At first those edges should be wide and then adiabatically reduced as the template homes in on a true minimum. Of course, the method could be generalized to arbitrary dimensions with arbitrary templates as long as those templates can be parametrized. Multiple Elastic Template dynamics could be defined similar to the multiple tracking dynamics introduced in section 3.4.

The ET method is well suited to complex pattern recognition tasks where a priori knowledge constrains strongly the possible class of patterns. Obviously for complex problems all the information known a priori must be used to extract the patterns from the data. We saw that even powerful neural network methods fail already at moderate complexity if that information is not utilized. On the other hand, we also indicated in section 4.2 how the ET strategy can be implemented in principle on a Hopfield type net. In that way ET could take advantage of analog VLSI technology. In the future, it would be interesting to study the performance of ET type nets. In particular, the optimal range of the fuzzy templates needs to be further investigated.

On a practical side, elastic template dynamics offers the advantage of being able to deal directly with continuous data distributions and of performing automatically an adaptive nonlinear fit to that data. There are obviously many possible applications of ET that could be further investigated, e.g., extracting rotation bands of high spin nuclei from complex spectra or extracting multiple overlapping circles in noisy Ring Imaging Cherenkov Counters, tracking in future ultra-compact high energy and nuclear physics detectors, and so on. ET may have useful applications in other areas, e.g. signal processing or medical image analysis, as well.

Acknowledgements: Valuable discussions on neural network algorithms with B. Denby, B. Schürmann, W. Keupper, and K. Frankel are gratefully acknowledged. Discussions on tracking methods and TPC detectors with H. Matis, G. Rai, P. Jacobs, J. Carroll, H. H. Wieman, H.G. Ritter, J. Harris, and A. Poskanzer are also gratefully acknowledged.

A Code Description

The code used for the simulations is written in C++. It runs on a Sparcstation 1, SunOS 4.0.3 and SunOS 4.1. We used the g++ (Version 1.37.0) compiler from the GNU Project[2], which is available for no charge via ftp from prep.ai.mit.edu with login anonymous. The code uses utility functions from the g++ library which is also available from GNU. These could be replaced without much effort if a different C++ compiler were to be used.

The code provides a graphical user interface based on the MIT X11 (Release 4) window system and the corresponding library (libX11). However it can be run also in batch mode. The program is controlled by a set of flags and parameters, which has to be provided in a file which is read at startup time. The control structure of the program consists of an outer loop controlling the mouse and keyboard events and an inner loop, which allows the accumulation of statistics.

At first data samples are either generated randomly or can be read from a file. A data point is realized by a structure holding the x-y-z coordinate, its distance to its next neighbour for calculating the density, a label for the track it is coming form and a label to assign a number for the track it is associated with by one of the algorithms.

A flag selects which algorithm is used for the analysis. Each algorithm fills an array of "Chains" being variable sized arrays of pointers pointing to the original addresses of the points. For the RF and the DP net, postprocessing is only implemented for straight lines. For helices no fit procedure is provided.

There is a choice of three different outputs:

- Postscript output of the distribution of the data points in real space.
- The number of found valid points, the number of found wrong points, the fractional error, and the rms error of the multiplicity and the momentum for straight lines accumulated over a sample of events.
- Original versus the fitted momentum distribution and the coordinates of the chained up points.

A copy of the image or the source code can be obtained by sending mail to **harlan@ds0.cip.physik.tu-muenchen.de** or **gyulassy@lbl.bitnet**.

[2]For information write to Free Software Foundation, 675 Mass. Ave, Cambridge, Mass 02139, USA

References

[1] J. Illingworth and J. Kitter, Computer Vision, Graphics, and Image Processing 44 (1988) 87.

[2] G. Rai et al., IEEE Trans. on Nucl. Sci., 37 (1990) 56.

[3] H. Grote, Rep. Prog. Phys. 50 (1987) 473.

[4] M. Regler and R. Frühwirth, in Proc. Advanced Study Institute on Techniques and Concepts in High Energy Physics, St. Croix 1988 (Plenum Publ. Corp., Rochester).

[5] Proc. Second Workshop on Experiments and Detectors for RHIC, LBL-24604 (1988), CONF-870543; Proc. Third Workshop on Experiments and Detectors for RHIC, BNL (1990).

[6] B. Denby, Comput. Phys. Commmun. 49 (1988) 429;
B. Denby and S. L. Linn, Comput. Phys. Commun. 56 (1990) 293;
B. Denby et al., IEEE Transactions on Nuclear Science 37 (1990) 248.

[7] C. Peterson, NIM A279 (1989) 537; Neural Networks 2 (1989) 475; LU-TP-89-1 (1989) Lund preprint.

[8] J.J. Hopfield and D.W. Tank, Biological Cybernetics, 52 (1985) 141; J.J Hopfield, Proc. Natl. Acad. Sci. USA, 79 (1982) 2554.

[9] Proc. IJCNN International Joint Conference on Neural Networks, Washington, D.C. 1989 (IEEE Cat.no. 89CH2765-6) ; San Diego 1990 (IEEE Cat.no. 90CH2879-5).

[10] B. Humpert, Comput. Phys. Commun. 56 (1990) 299.

[11] C. Mead, Analog VLSI and Neural Systems (Addison-Wesley Pub. Co., 1989).

[12] R. Durbin and D. J. Willshaw, Nature 326 (1987) 689; W. S. Wong and C. A. Funka-Lea, IJCNN 1990 op cit, Vol III, p.799.

Figure Captions

Fig. 1 A hard problem of tracking 15 lines on the top left given the measured points on top right including 100% noise. The confusion of the neural net algorithm[6, 7] on the bottom left is apparent. The long range correlations are however easily identified with the Elastic Tracking method on the bottom right.

Fig. 2 The energy surface for elastic tracking in a 2D example with 5 straight tracks discussed in the text illustrating the problem of local minima at high resolution (small w).

Fig.3 The weight function used for the Denby-Peterson net as a function of the relative angle between segments ij and jk.

Fig. 4 A typical 3D event with helical tracks simulating conditions in the HISS/TPC[2] with 20% noise. The solution of the road finding method is shown illustrating the problem of track segmentation and unidentified track points.

Fig. 5 Performance comparison of Road Finder[4], Neural Net[6, 7], and Elastic Tracking algorithms for tracking helices as in Fig. 4. The distribution of found valid points and the fractional error as a function of the local track density (21) shows that all three methods work well for this low track density ($\rho_{ij} \lesssim 1$). While the statistics in the high density tails are very poor there is a tendency for the DP net to give the best performance.

Fig. 6 Examples of straight-line tracking problems in 2D with increasing track multiplicity. The input configuration for each multiplicity is shown in the top panels. The data with 20% noise and 3% relative Δx measurement error together with the tracks found by the three methods are shown. Note that the Road Finder method breaks down between $N = 5 - 10$, the DP net breaks down between $N = 10 - 15$, while ET finds the correct solution in all cases.

Fig. 7 Local performance measures for 2D straight-line tracking illustrated in Fig. 6 for multiplicity $N = 10, 15$. For this high track density the ET method performs the best. Note the rapid increase of the error produced by the Road Finder and DP net methods with increasing multiplicity.

Fig. 8 Global performance measures for the 2D tracking problem in Figs. 6,7 as a function of the median track density for $N = 3, 5, 10, 15, 20$. The left side shows the performance with no noise. The right side shows results with 20% noise. The solid points in all panel correspond to $N = 15$ with 100% noise. The upper panels show the rms multiplicity error, the lower panels show the momentum resolution as defined in the text.

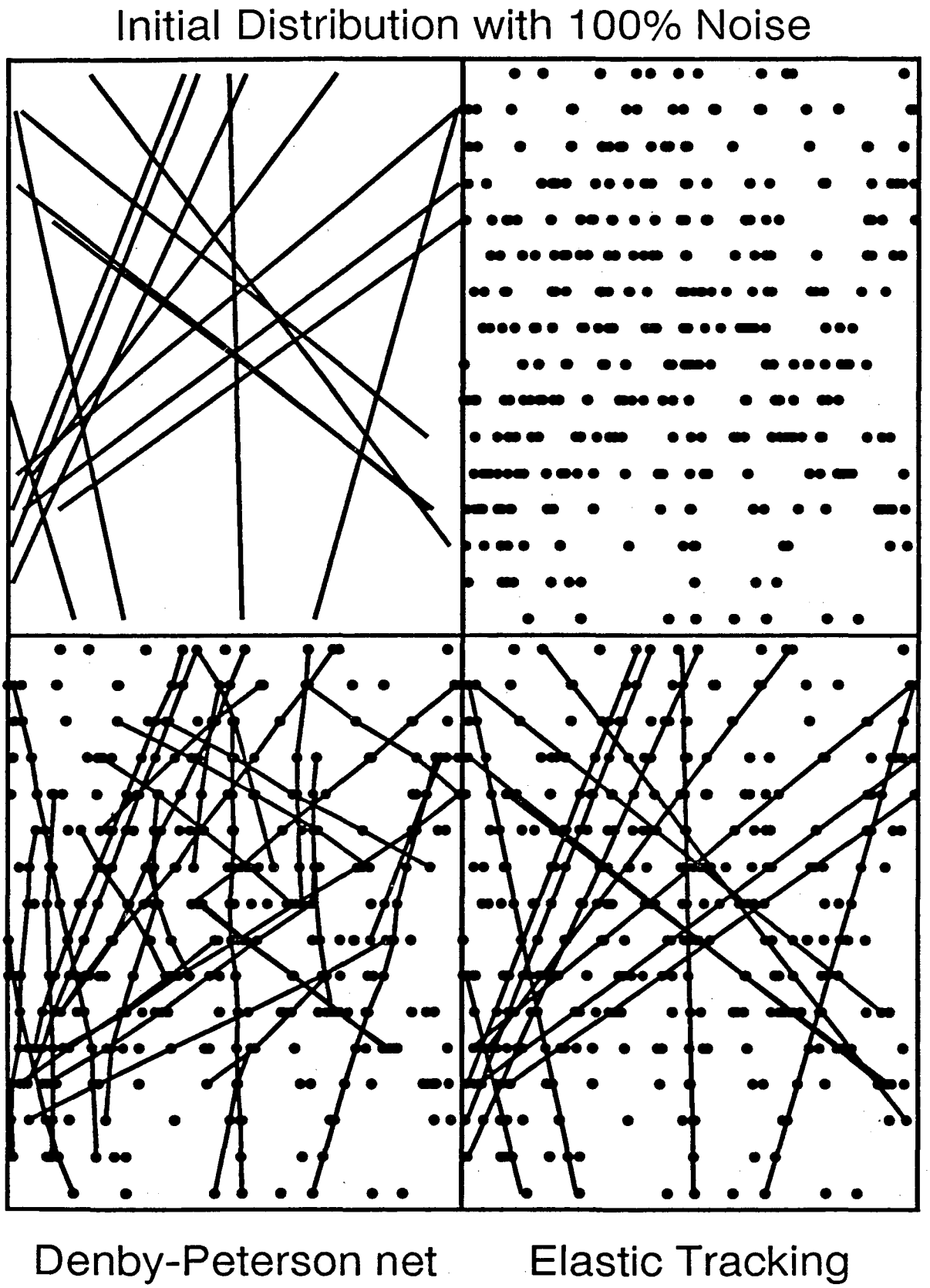
Initial Distribution with 100% Noise
Denby-Peterson net
Elastic Tracking

Fig. 1

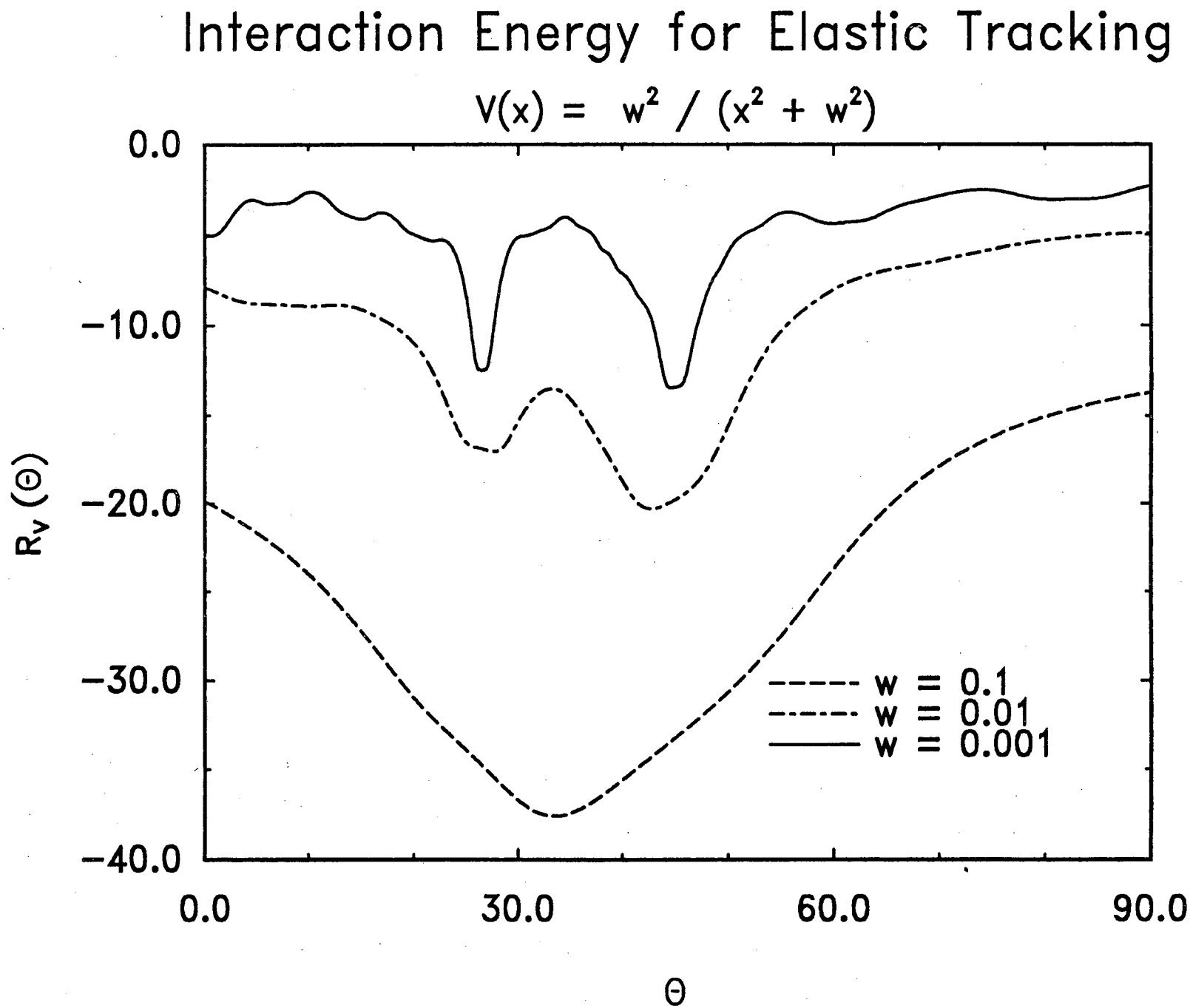
Interaction Energy for Elastic Tracking
V(x) = w² / (x² + w²)
0.0
−10.0
−20.0
−30.0
−40.0
R_V (Θ)
0.0
30.0
60.0
90.0
Θ
w = 0.1
w = 0.01
w = 0.001

Fig. 2

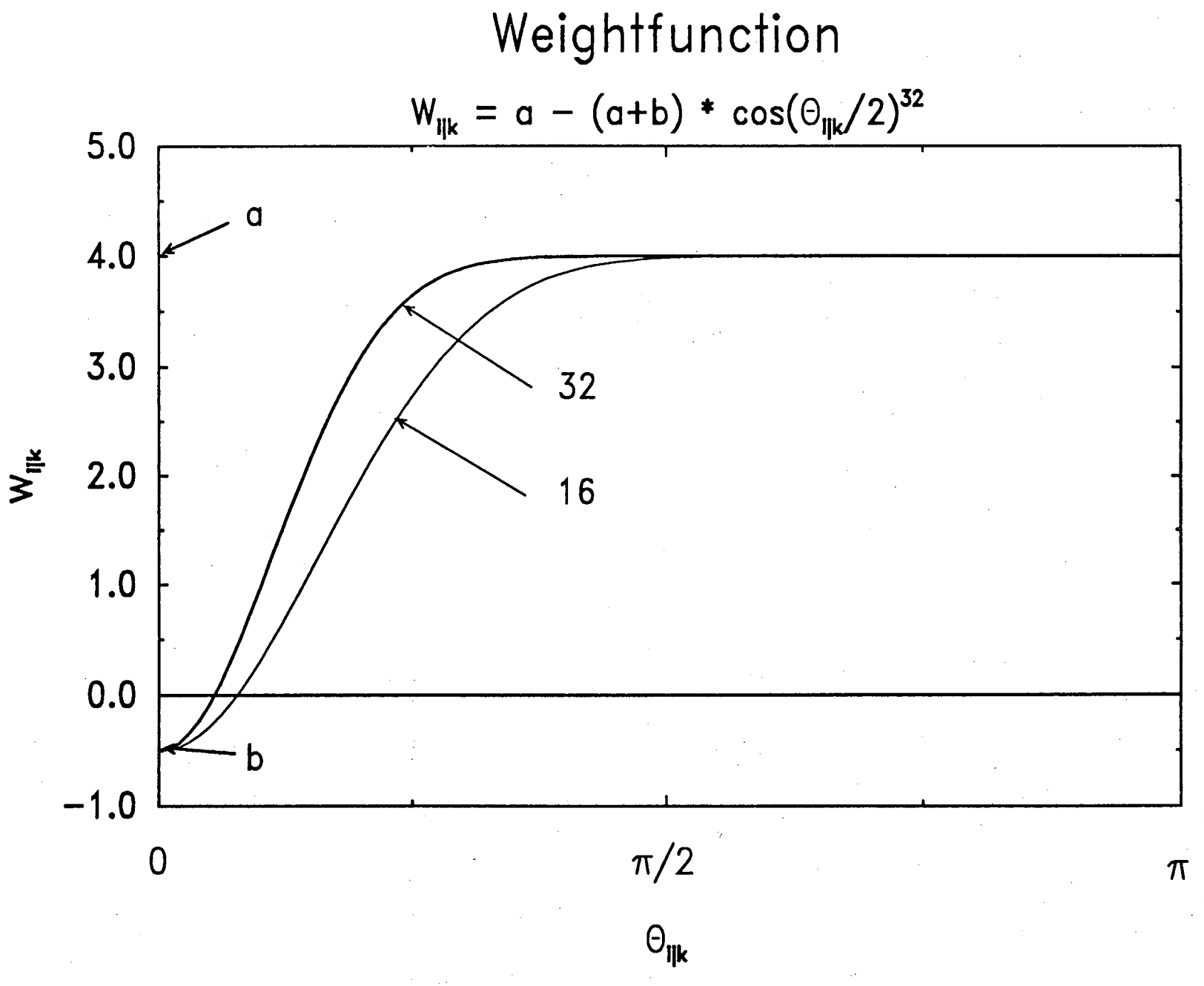
Weightfunction
$W_{l|k} = a - (a+b) * \cos(\Theta_{l|k}/2)^{32}$
5.0
4.0
3.0
2.0
1.0
0.0
−1.0
$W_{l|k}$
a
b
32
16
0
π/2
π
$\Theta_{l|k}$

Fig. 3

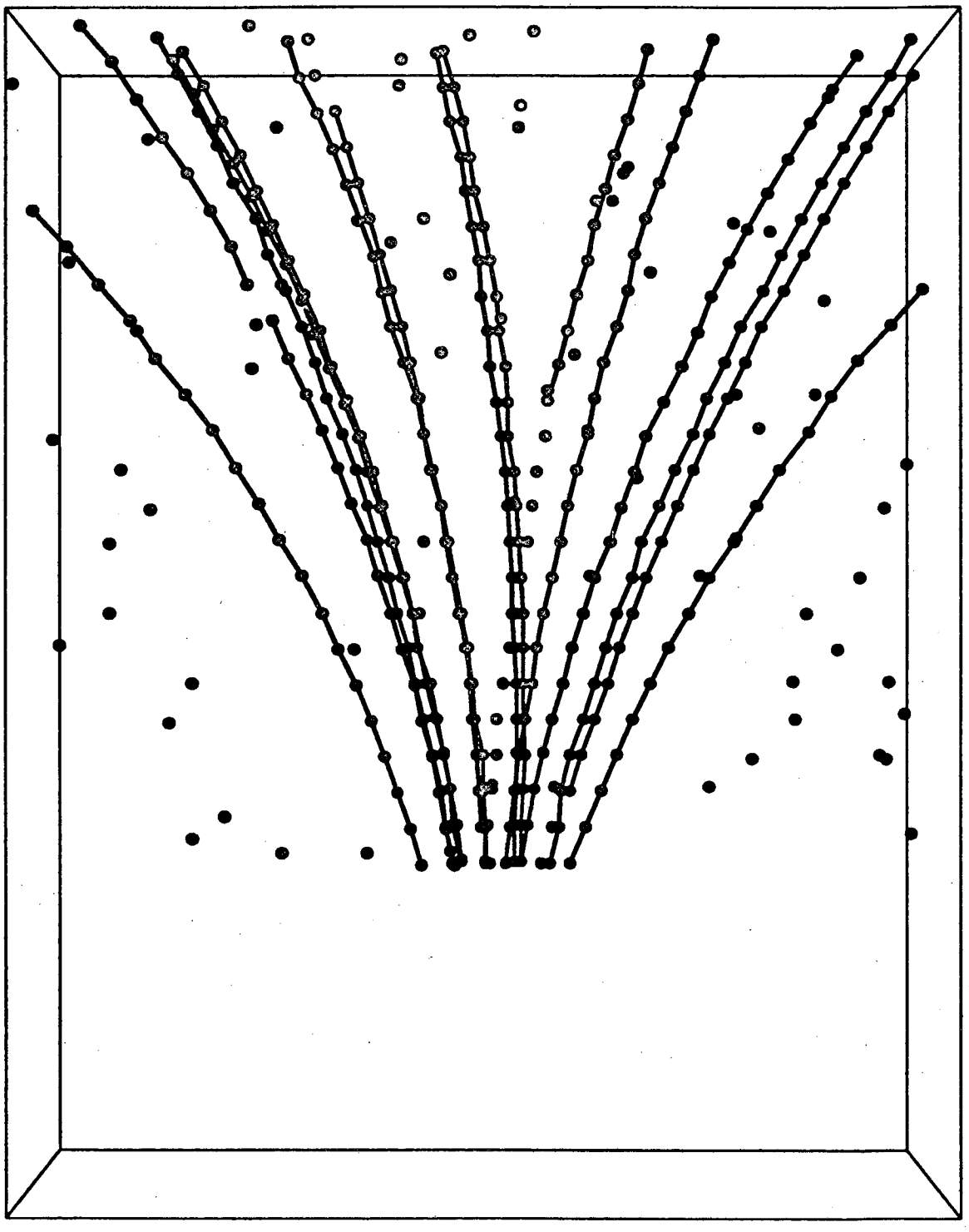

Fig. 4

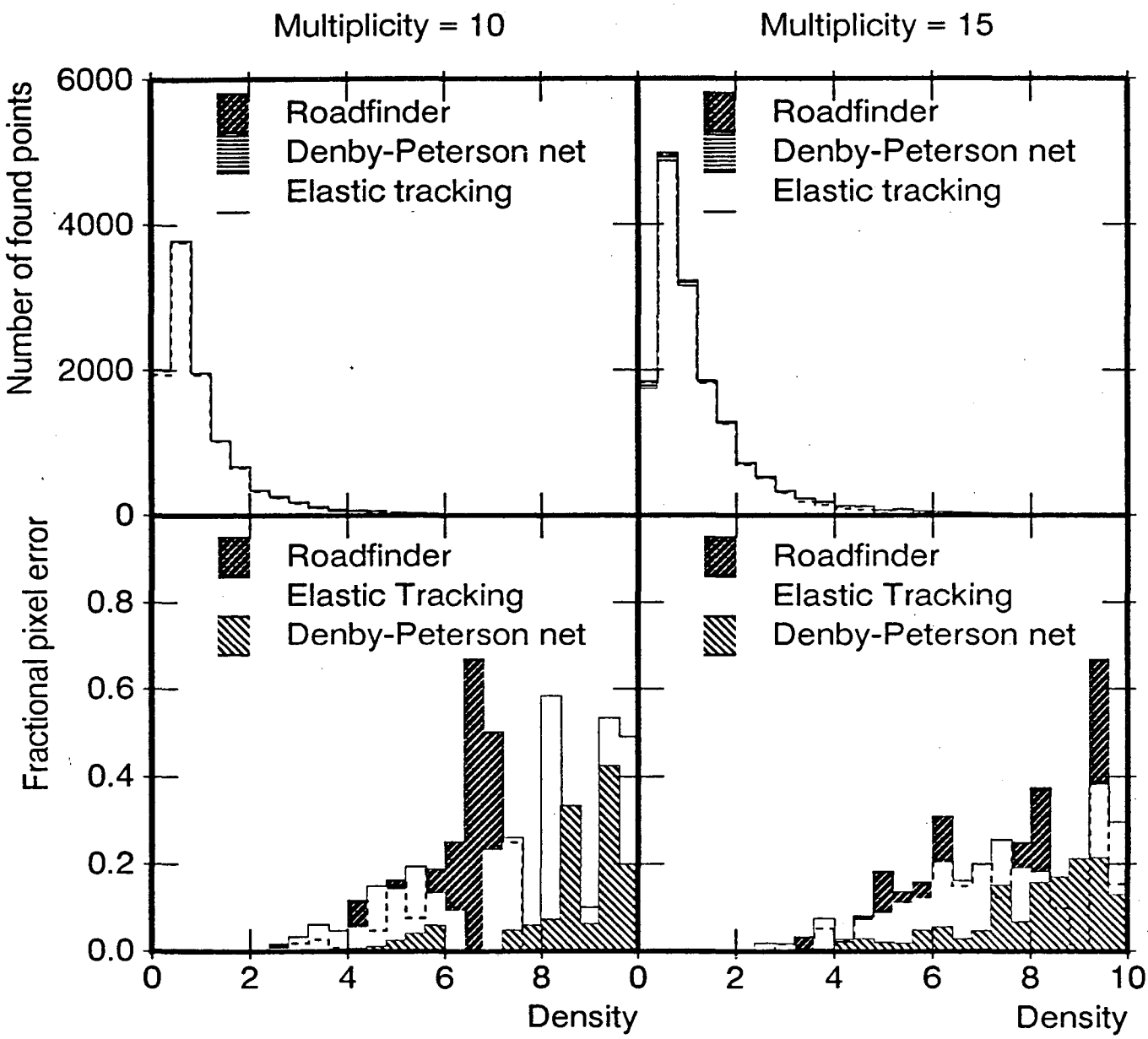
Multiplicity = 10
Multiplicity = 15
Number of found points
6000
4000
2000
0
Roadfinder
Denby-Peterson net
Elastic tracking
Roadfinder
Denby-Peterson net
Elastic tracking
Fractional pixel error
0.8
0.6
0.4
0.2
0.0
Roadfinder
Elastic Tracking
Denby-Peterson net
Roadfinder
Elastic Tracking
Denby-Peterson net
0
2
4
6
8
0
2
4
6
8
10
Density
Density

Fig. 5

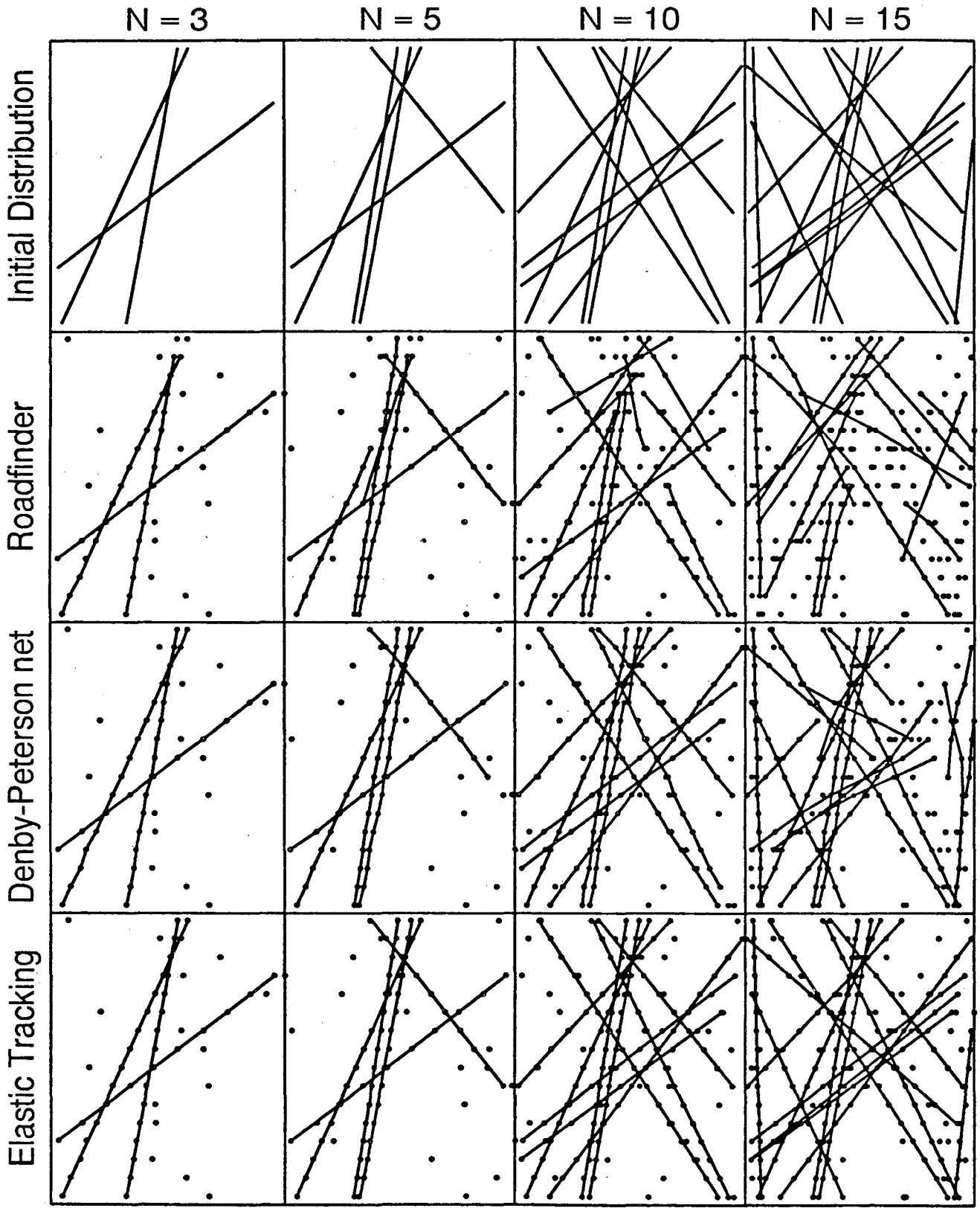
N = 3
N = 5
N = 10
N = 15
Initial Distribution
Roadfinder
Denby-Peterson net
Elastic Tracking

Fig. 6

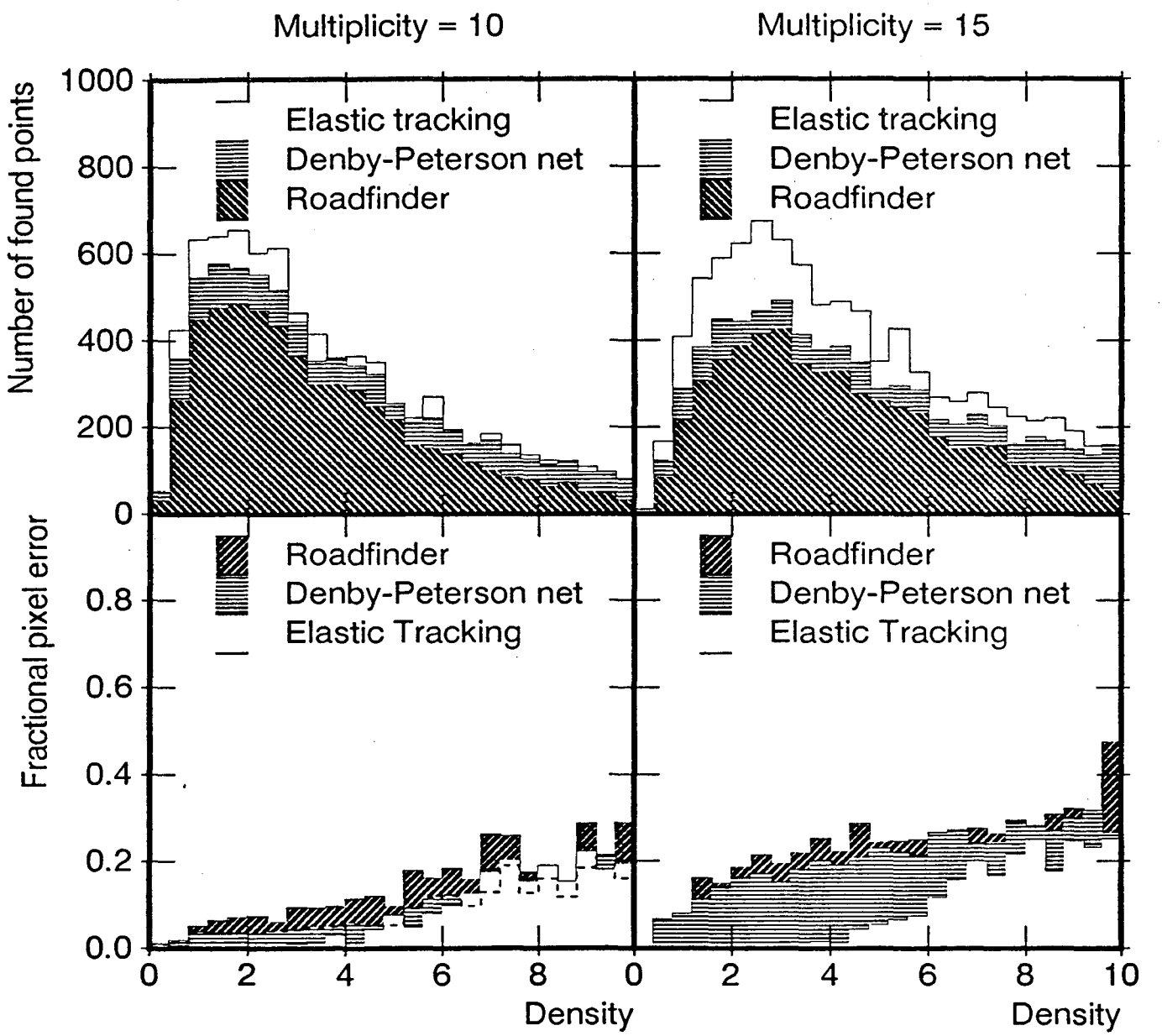
Multiplicity = 10
Multiplicity = 15
Number of found points
1000
800
600
400
200
0
Elastic tracking
Denby-Peterson net
Roadfinder
Fractional pixel error
0.8
0.6
0.4
0.2
0.0
Roadfinder
Denby-Peterson net
Elastic Tracking
0
2
4
6
8
10
Density

Fig. 7

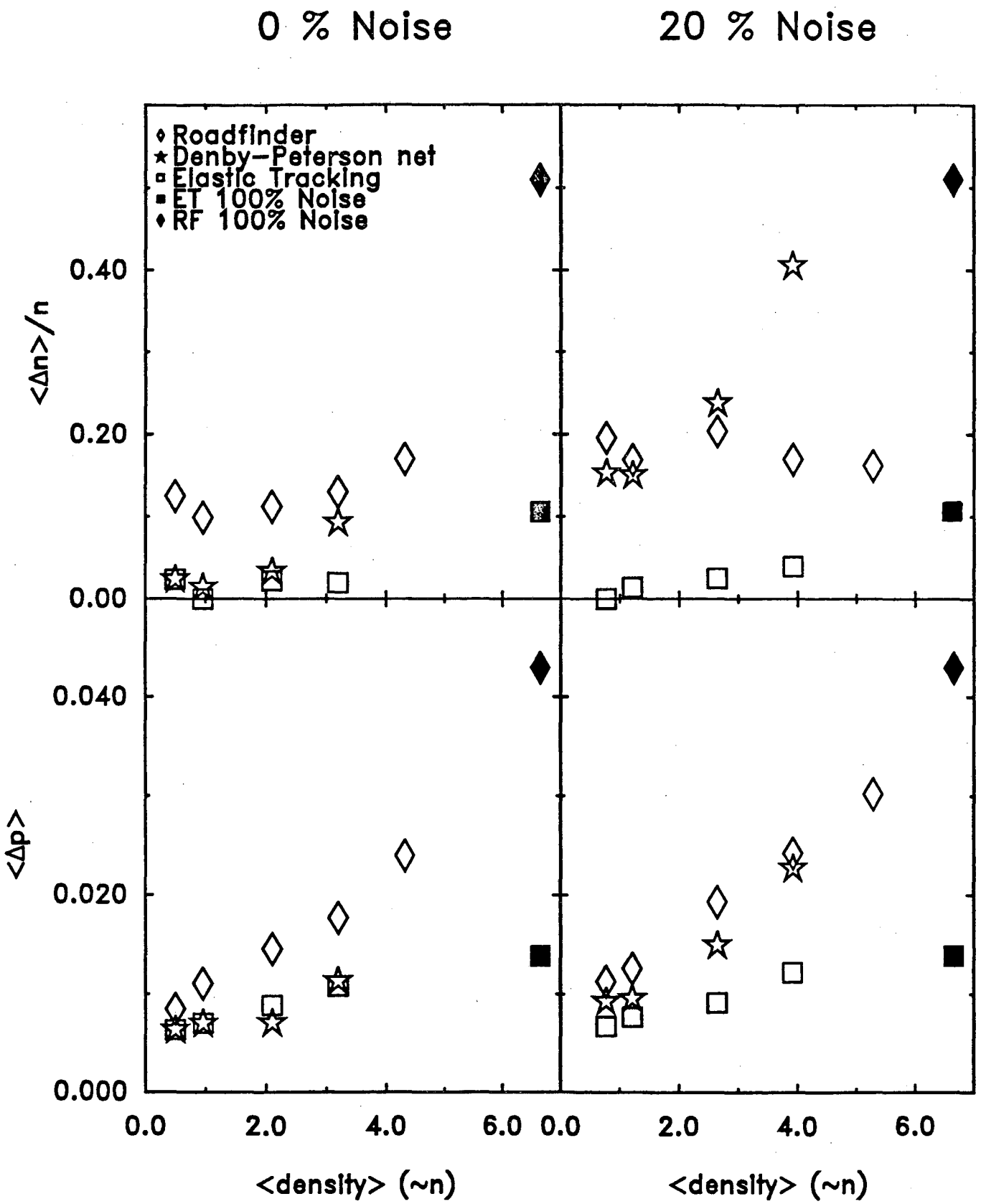
0 % Noise
20 % Noise
Roadfinder
Denby–Peterson net
Elastic Tracking
ET 100% Noise
RF 100% Noise
<Δn>/n
0.40
0.20
0.00
<Δp>
0.040
0.020
0.000
0.0
2.0
4.0
6.0
<density> (~n)

Fig. 8

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
INFORMATION RESOURCES DEPARTMENT
BERKELEY, CALIFORNIA 94720