Mean Field Point Matching by Vernier Network and by Generalized Hough Transform

Chien-Ping Lu
Eric Mjolsness

Research Report YALEU/DCS/RR-974
May 1993

Mean Field Point Matching by Vernier Network and by Generalized Hough Transform

Chien-Ping Lu and Eric Mjolsness*

May 25, 1993

Abstract

We introduce neural network architectures for solving certain point matching problems that commonly arise in computer vision. The neural networks arise from a new application of mean field theory (MFT) techniques, in which a hierarchical representation of continuous variables is used to eliminate some of the spurious local minima which would remain in a more conventional MFT neural net for the same problem. The resulting *vernier network* algorithm is related to the more conventional generalized Hough transform (with fixed bins) for solving the same problem. The vernier network can also be elaborated to a *filtered vernier network* which is more efficient. All are improvements on the conventional MFT neural network. We compare performance and cost of these four algorithms (conventional MFT, generalized Hough, vernier network, and filtered vernier network) under various noise conditions. The vernier network algorithm can be easily extended to more general model-based object recognition problems.

1 Introduction

We consider the problem of matching or registering two sets of point features in which all of the points in one set (the model) undergo a common geometric transformation made up of a rotation and a translation, following which each transformed point is independently translated by a small random vector, resulting exactly in the other set of points (the scene). The *point matching problem* is to recover the actual transformation and correspondence that relates the two sets of points.

Such problems arise naturally in computer vision, usually with further elaborations most of which we will not consider in this paper. For example, more difficult versions of the problem delete some of the model points randomly, introduce a global scale change, and/or add noisy labels to the points. One complication we will explore is the addition of many spurious scene points according to a background probability distribution. The point matching problem may also be generalized to three dimensions in several ways that are important to computer vision. A good algorithm for solving such point matching problems is essential to object recognition and two-frame rigid motion estimation problems, assuming the relevant images have each been preprocessed into a sparse set of significant features.

*This work was supported by AFOSR grant AFOSR-90-0224 and DARPA/ONR grant N00014-92-J-4048. The authors are with the Department of Computer Science, Yale University, P.O.Box 2158 Yale Station, New Haven CT 06520-2158.

Each of these point matching problems have two components. One is to establish the correct correspondence between scene and model points. The other is to estimate the position and orientation of the scene points relative to the model points (which we refer as the "pose" of the model in the scene) assuming a known correspondence. These two components are tightly coupled. If the point correspondence is known, the pose can be determined easily by least squares procedures. Similarly, for known pose, the problem reduces to an assignment problem.

The search for the optimal solution to the problem can be performed either in correspondence space or in pose space. The correspondence space search requires exploring a potentially exponential subset of all possible correspondences. For example *tree pruning methods* [Bai84, GLP87, AF86] searches over a tree in which each node represents a partial match, and evaluates each partial match through the pose that best fits it. An appropriate noise model is essential for prunning away large portions of the search space [Bai84]. However, it has been shown that even with such constrained search, the expected search time is still exponential [Gri90]. Examples of techniques searching pose space are the generalized Hough transform [Sto87, GLP87]. and transformation sampling [Cas88], in both of which all possible pairings of one model point and one scene point "vote" for the best candidate among a set of discretized poses. The two methods differ in their algorithms for collecting votes.

In this paper we consider the simplest case of 2D-2D point matching, in which we are given $\hat{N}$ 2D scene points extracted from the observed image: $X = \mathbf{x}_1, \ldots, \mathbf{x}_{\hat{N}}$, some of which correspond to N 2D model points: $Y = \mathbf{y}_1, \ldots, \mathbf{y}_N$. We can formulate the problem as minimization of the following objective function (see e.g. [Mjo91])

$$E_{\text{match}}(\mathbf{M}, \theta, \mathbf{t}) = \sum_{ia} M_{ia} \|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2 = \sum_{ia} M_{ia} C_{ia}(\theta, \mathbf{t}), \tag{1}$$

where $\{M_{ia}\} = \mathbf{M}$ is a match matrix representing the unknown correspondence, $\{C_{ia}(\theta, \mathbf{t})\} = \mathbf{C}(\theta, \mathbf{t})$ is the parametric cost matrix, $\mathbf{R}_\theta$ is a rotation matrix with rotation angle θ, and $\mathbf{t}$ is a translation vector. With this formulation, we assume a gaussian noise model on point locations instead of the polygonally bounded error model used in [Bai84, Cas88].

By contrast with the relatively standard methods mentioned above, we propose to solve the point matching problem by optimizing (1) directly over correspondences, rotations and translations. This is a *parametric assignment problem* that differs from other purely discrete assignment problems in that the cost matrix is determined by continuous variables also subject to optimization. We turn this combined discrete and continuous optimization problem into a purely continous one by using Mean Field Theory (MFT) techniques from statistical physics, adapted as necessary to produce effective algorithms in the form of analog neural networks. One simplifying characteristic of the resulting algorithms is that they are essentially described by objective functions rather than by abstract computer programs. For example the conventional MFT approach to point matching would yield an effective objective function for continuous-valued $M_{ia} \in [0, 1]$ elements such as [KY91, Yui90]

$$\begin{aligned} E_{\text{WTA}}(\mathbf{M}, \theta, \mathbf{t}) \;=\; & \textstyle\sum_{ia} M_{ia} \|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2 + (A/2) \sum_a (\sum_i M_{ia} - 1)^2 \\ & + (1/\beta) \textstyle\sum_{ia} M_{ia} U_{ia} - (1/\beta) \sum_i \log \left(\sum_a \exp U_{ia} \right) \end{aligned} \tag{2}$$

(which arises from recent progress in Mean Field Theory neural networks [PS89, Sim90, GY91])

or the low-noise approximation [Mjo91, MG90]

$$E_{\text{eff}}(\theta, \mathbf{t}) = -\frac{1}{\beta} \log \sum_{ia} e^{-\beta \|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2}. \tag{3}$$

Unfortunately, even with continuation in β, straightforward descent algorithms for both of these objectives suffer from the presence of spurious local minima, particularly in the determination of the rotation parameter θ.

We propose a new *vernier network* algorithm arising from a novel application of MFT to a hierarchical representation of the continuous geometric variables, which in effect can be thought of as a binning transformation that turns the original optimization problem over a single pose space into several optimization problems over smaller intervals or Cartesian products of intervals (the bins). Initially, the centers of each bin are designated as principle poses, which are fine-tuned by associated vernier variables. We show that in a certain approximation, the vernier network algorithm reduces to the generalized Hough transform. Our experiments show that the vernier network can use many fewer bins than the Hough transformation while achieving much better performance. It is however quite expensive, so we also consider a further modification (the *filtered vernier network*) which introduces a two-level multiscale search and is significantly more efficient. We do not however explore true self-similar multiscale algorithms, either for the Hough transform or for the vernier network. With some modifications to the objective function (1), the vernier network algorithm can be easily extended to more general model-based object recognition problems.

2 The Theory

2.1 The Generalized Hough Transform

The generalized Hough transform (GHT) finds a solution to the point matching problem by searching for large clusters of evidence in a three-dimensional array of bins (the Hough space) which results from quantizing each dimension of the three-dimensional pose space. Each possible pair of scene point and model point $(\mathbf{x}_i, \mathbf{y}_a)$ is consistent with the pose hypotheses $(\theta, \mathbf{t} = \mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a)$ for all θ. In the Hough space, there will be several bins intersecting with the pose hypotheses, and each such bin will receive a "vote" that the correct pose falls into it. Given a model and scene pair, GHT records the number of votes each bin receives from all possible matches. The bin that receives the maximum number of votes will be chosen as the putative solution to the problem. Assume that $2\epsilon_\theta$, $2\epsilon_{\mathbf{t}}$ and $2\epsilon_{\mathbf{t}}$ are the bin widths along the θ, x and y dimensions, respectively. Adopting the "pose clustering" approach in [Sto87], we can define the GHT for the 2D point matching problem as

$$\text{GHT}(j,k,l) = \sum_{ia} \text{Indicator}\left((\mathbf{x}_i - \mathbf{R}_{\hat{\theta}_j} \mathbf{y}_a - \hat{\mathbf{t}}_{kl}) \in [-\epsilon_{\mathbf{t}}, \epsilon_{\mathbf{t}}] \times [-\epsilon_{\mathbf{t}}, \epsilon_{\mathbf{t}}]\right), \tag{4}$$

where $(\hat{\theta}_j, \hat{\mathbf{t}}_{kl})$ is the center of bin (j,k,l). In the limiting case where there are infinite number of bins and the data is noiseless, the generalized Hough transform is simply

$$\text{GHT}(\theta, \mathbf{t}) = \sum_{ia} \text{Indicator}\left(\mathbf{x}_i = \mathbf{R}_\theta \mathbf{y}_a + \mathbf{t}\right). \tag{5}$$

Note that

$$\text{Indicator } (\mathbf{x}_i = \mathbf{R}_\theta \mathbf{y}_a + \mathbf{t}) = \lim_{\beta \to \infty} e^{-\beta \|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2}. \tag{6}$$

Though the discrete voting scheme utilized in GHT looks different from continuous energy minimization approach, we will show in next section that the two formulations are related if a specific constraint on match variables is used with the matching objective function (1).

2.2 The Mean Field Theory Algorithm

To solve a point matching problem by minimizing (1), we need to enforce some constraints on match variables M_{ia}; otherwise the objective function can always be minimized with all M_{ia} set to zero. A standard constraint on $\mathbf{M}$ for assignment problems is $\sum_i M_{ia} = 1, \forall a$ and $\sum_a M_{ia} = 1, \forall i$, i.e. that $\mathbf{M}$ is a permutation matrix. In more general cases where there are spurious scene points, we may impose a generalized matching constraint $\sum_i M_{ia} = 0$ or $1, \forall a$ and $\sum_a M_{ia} = 0$ or $1, \forall i$, or we can use the weaker constraint

$$\sum_{ia} M_{ia} = N, \tag{7}$$

which implies that there are exactly N matches among all possible matches. An advantage of using (7) is that we do not have to deal with spurious or missing features explicitly; they are modeled by empty rows or columns of the $\mathbf{M}$ matrix. While this constraint is weaker, we can argue that it is a good approximation to real matching constraints. An entropy argument in favor of this constraint is that among matrices satisfying (7), the vast majority have low occupancy for most rows and columns. There is also an energy argument: multiple assignments are allowed but discouraged by the effective energy term $\beta \sum_{ia} M_{ia} \|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2$ unless two values of $\mathbf{x}_i$ or two values of $\mathbf{y}_a$ happen to be within $\frac{1}{2\sqrt{\beta}}$ of each other. So (7) is plausible as the sole constraint on $\mathbf{M}$.

In the following, we designate the allowed set of matrices $\mathbf{M}$ that satisfy the given constraints (whatever they are) by $\mathcal{M}$. Assume a Gibbs distribution for $\mathbf{M}$

$$P_\beta(\mathbf{M_{ia}}|\theta, \mathbf{t}) \;=\; \frac{1}{Z_\beta} e^{-\beta E_{\text{match}}(\mathbf{M_{ia}}, \theta, \mathbf{t})} \tag{8}$$

where $Z_\beta = \sum_{\mathbf{M} \in \mathcal{M}} e^{-\beta E_{\text{match}}(\mathbf{M_{ia}}, \theta, \mathbf{t})}$ is the normalization factor known as *partition function* in statistical physics, and β is the inverse temperature. The contribution of $\mathbf{M}$ to the partition function can be exactly or approximately evaluated, leaving an effective objective function $E_{\text{eff}} = -\frac{1}{\beta} \log Z_\beta$ depending on the pose parameters only. By introducing a source field $\{H_{ia}\}$ and defining $Z_\beta[\mathbf{H}] = \sum_{\mathbf{M} \in \mathcal{M}} e^{-\beta E_{\text{match}}(\mathbf{M}, \theta, \mathbf{t}) + \sum_{ia} H_{ia} M_{ia}}$, the mean field $\langle \mathbf{M} \rangle_\beta$ can be calculated as

$$\langle M_{ia} \rangle_\beta = -\frac{1}{\beta} \left. \frac{\partial \log Z_\beta[\mathbf{H}]}{\partial H_{ia}} \right|_{H_{ia}=0}. \tag{9}$$

The basic premises of the Mean Field Theory (MFT) approach are that at very high temperature (small β), the effective objective function is convex, and as $\beta \to \infty$, the mean field $\langle \mathbf{M} \rangle_\beta$ will

usually approach the optimal $\mathbf{M}^*$. Therefore, by tracking the local minimum of the effective objective function from high temperature down to low temperature, we may be able to find the optimal pose as well as the optimal correspondence that supports it.

The form of E_{eff} depends on the constraints $\mathcal{M}$ with which the partition function is evaluated. [Mjo91] (Section 2.5) provides a way to approximate the summation over $\mathbf{M}$ subject to the constraint (7) as

$$Z_\beta \approx \frac{1}{N!}[\sum_{ia} e^{-\beta\|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2}]^N \tag{10}$$

from which we find the effective objective function equivalent to (3)

$$-\frac{1}{\beta}[N \log \sum_{ia} e^{-\beta\|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2} - \log N!]. \tag{11}$$

The mean field $m_{ia}(\theta, \mathbf{t}, \beta) = \langle M_{ia} \rangle_\beta$ for given pose $(\theta, \mathbf{t})$ can be calculated using (9) as

$$m_{ia}(\theta, \mathbf{t}, \beta) = \frac{N e^{-\beta\|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2}}{\sum_{jb} e^{-\beta\|\mathbf{x}_j - \mathbf{R}_\theta \mathbf{y}_b - \mathbf{t}\|^2}}. \tag{12}$$

We note that the constraint (7) is satisfied for the mean field since $\sum_{ia} M_{ia} \equiv N$. Because the logarithm function is monotonic, minimizing (11) is equivalent to maximizing

$$\sum_{ia} e^{-\beta\|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2}. \tag{13}$$

Applying the fixed-point-preserving algebraic transformation rule [MG90]

$$e^X \to (X + 1)\sigma - \sigma \log \sigma \tag{14}$$

to (13), we get the following equivalent objective function

$$E_\sigma(\{\sigma_{ia}\}, \theta, \mathbf{t}) = \sum_{ia} \sigma_{ia}\|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2 + \frac{1}{\beta}\sum_{ia} \sigma_{ia}(\log \sigma_{ia} - 1). \tag{15}$$

By comparing σ_{ia} to M_{ia}, this objective function is simply the original objective function E_{match} with match variable σ_{ia} subject to a penalty function $\sigma_{ia}(\log \sigma_{ia} - 1)$. This penalty function is plotted in Figure 1.

It prohibits negative values of σ_{ia} by means of an infinite slope at the origin, and discourages large positive value of σ_{ia}. Optimizing with respect to $\{\sigma_{ia}\}$, we obtain a simplified objective function

$$F(\theta, \mathbf{t}, \beta) \equiv E_\sigma(\{\sigma^*_{ia}\}, \theta, \mathbf{t}) = -\frac{1}{\beta}\sum_{ia} \sigma^*_{ia}(\theta, \mathbf{t}) \tag{16}$$

where

$$\sigma^*_{ia}(\theta, \mathbf{t}) = e^{-\beta\|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2}, \tag{17}$$

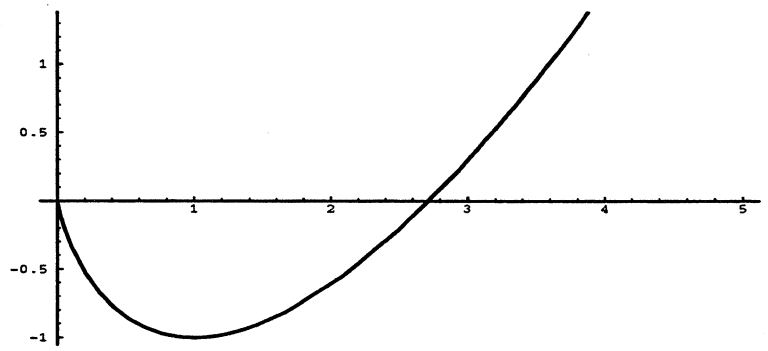

Figure 1: Plot of the penalty function $x(\log x - 1)$. Note that the minimum of this function is at $x = 1$.

and a gradient descent dynamics for finding the saddle point of F (16) is

$$\begin{aligned} \dot{\mathbf{t}} &= -\kappa \sum_{ia} \sigma_{ia}(\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}) \\ \dot{\theta} &= -\kappa \sum_{ia} \sigma_{ia}(\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t})^t (\mathbf{R}_{\theta+\frac{\pi}{2}} \mathbf{y}_a). \end{aligned} \tag{18}$$

Since such local descent methods are easily trapped in local minima, we seek the optimal solution through continuation over β which controls the roughness of the energy surface (deterministic annealing). Using (9), (16), and (17), one can calculate that $\langle M_{ia} \rangle_\beta = \sigma_{ia}(\theta, \mathbf{t})$ under the Gibbs distribution $P_\beta(\mathbf{M}|\theta, \mathbf{t}) = \frac{1}{Z_\beta} e^{-\beta E_\sigma(\{M_{ia}\}, \theta, \mathbf{t})}$. Note that the most favorable value of each mean field match variable σ_{ia} is 1. In the beginning of the annealing process, β is small and all σ_{ia} are close to 1. This means that when no information is available yet, each possible match is considered equally probable. As β increases, the values of some σ_{ia} change as the corresponding matches become less probable.

An alternative way to find the global minimum of the objective function is to sample the negative of the energy surface at some set of discrete points, and then pick the point having the highest value. The required density of the sample points depends on the roughness of the energy surface, which is controlled by β. Considering a partition of the translation space into bins of width $2\epsilon_\mathbf{t} \times 2\epsilon_\mathbf{t}, \beta^{-1} \approx 2\epsilon_\mathbf{t}^2$, we obtain a set of discrete samples of $F(\theta, \mathbf{t}, \beta)$ for a rotation angle θ:

$$F(k, l; \theta, \beta) = F(\theta, \hat{\mathbf{t}}_{kl}, \beta), \tag{19}$$

where $\hat{\mathbf{t}}_{kl}$ is the center of the translation bin (k, l). Furthermore, (16) and (17) show that

$$-\beta F(\theta, \hat{\mathbf{t}}_{kl}, \beta) \approx \sum_{ia} \text{Indicator } ((\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \hat{\mathbf{t}}_{kl}) \in [-\epsilon_\mathbf{t}, \epsilon_\mathbf{t}] \times [-\epsilon_\mathbf{t}, \epsilon_\mathbf{t}]), \tag{20}$$

from which we see that for a *fixed rotation angle*, the GHT with bin width along x and y translation equal to $2\epsilon_{\mathbf{t}}$ can be thought of as an approximate method for sampling the energy surface of MFT objective function at $\beta = (2\epsilon_{\mathbf{t}})^{-1}$. Figure 2 demonstrates this connection quantitatively. Increasing β in MFT algorithm corresponds to finer binning in the GHT. In both cases, the objective functions reveal more detailed information as β is increased, and their corresponding energy surfaces become bumpier. We note that the bin widths along the rotation axis are not related to the inverse temperature β. In the next section. we will show a complete correspondence between the GHT and MFT algorithms, including rotations, by introducing a hierarchical representation of rotation.

2.3 The Vernier Network

Though the effective objective function (16) is non-convex over translation at low temperatures, its dependence on rotation is non-convex even at relatively high temperature. We propose to overcome this problem by applying MFT to a hierarchical representation of rotation

$$\theta = \sum_{j=0}^{J-1} \chi_j(\hat{\theta}_j + \theta_j),\ \theta_j \in [-\epsilon_\theta, \epsilon_\theta], \qquad \left(\text{and } \mathbf{t} = \sum_{j=0}^{J-1} \chi_j \mathbf{t}_j\right) \tag{21}$$

where $\epsilon_\theta = \pi/2J$, $\hat{\theta}_j = (j + \frac{1}{2})\frac{\pi}{J}$ are the centers of each interval, and θ_j are "vernier" fine-tuning variables. The χ_j's are binary variables (so $\chi_j \in \{0, 1\}$) that satisfy the winner-take-all constraint $\sum_j \chi_j = 1$, and therefore pick a best bin. The essential reason that this hierarchical representation of θ has fewer spurious local minima than the conventional analog representation is that the change of variables also increases the *connectivity of the network's state space*: big jumps in θ can be achieved by local variations of the 0/1 χ variables.

A full Mean Field Theory derivation for the resulting network will be worked out in the next section. In this section we show informally how the MFT objective function and neural network for the rotation parameters θ_j arise.

Changing the representation for θ gives the new partition function (as detailed in the next section)

$$Z = \sum_{\{\chi | \sum_j \chi_j = 1\}} \prod_j \int_{[-\epsilon_\theta, \epsilon_\theta]} d\theta_j e^{-\beta \chi_j F(\hat{\theta}_j + \theta_j, \mathbf{t}_j)} \tag{22}$$

The contribution of each θ_j to the partition function can be evaluated as (c.f. [PS89])

$$\begin{aligned}
\textstyle\int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j e^{-\beta \chi_j F(\hat{\theta}_j + \theta_j, \mathbf{t}_j)} &= \textstyle\int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j \int dv_j e^{-\beta \chi_j F(\hat{\theta}_j + v_j, \mathbf{t}_j)} \delta(v_j - \theta_j) \\
&= \textstyle\int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j \int dv_j \int_I du_j e^{-\beta \chi_j F(\hat{\theta}_j + v_j, \mathbf{t}_j)} e^{-u_j(v_j - \theta_j)} \\
&= \textstyle\int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j e^{u_j \theta_j} \int dv_j \int_I du_j e^{-\beta \chi_j F(\hat{\theta}_j + v_j, \mathbf{t}_j)} e^{-u_j v_j} \\
&= \tfrac{2}{u_j} e^{-u\hat{\theta}_j} \sinh(\tfrac{u_j \pi}{2J}) \textstyle\int dv_j \int_I du_j e^{-\beta \chi_j F(\hat{\theta}_j + u_j, \mathbf{t}_j)} e^{-u_j v_j}
\end{aligned} \tag{23}$$

where I is the imaginary axis. Combining these results gives

$$Z = 2^N \sum_{\chi} \prod_j \int dv_j \int_I du_j e^{-\beta E'_j(\chi_j, u_j, v_j, \mathbf{t}_j)} \tag{24}$$

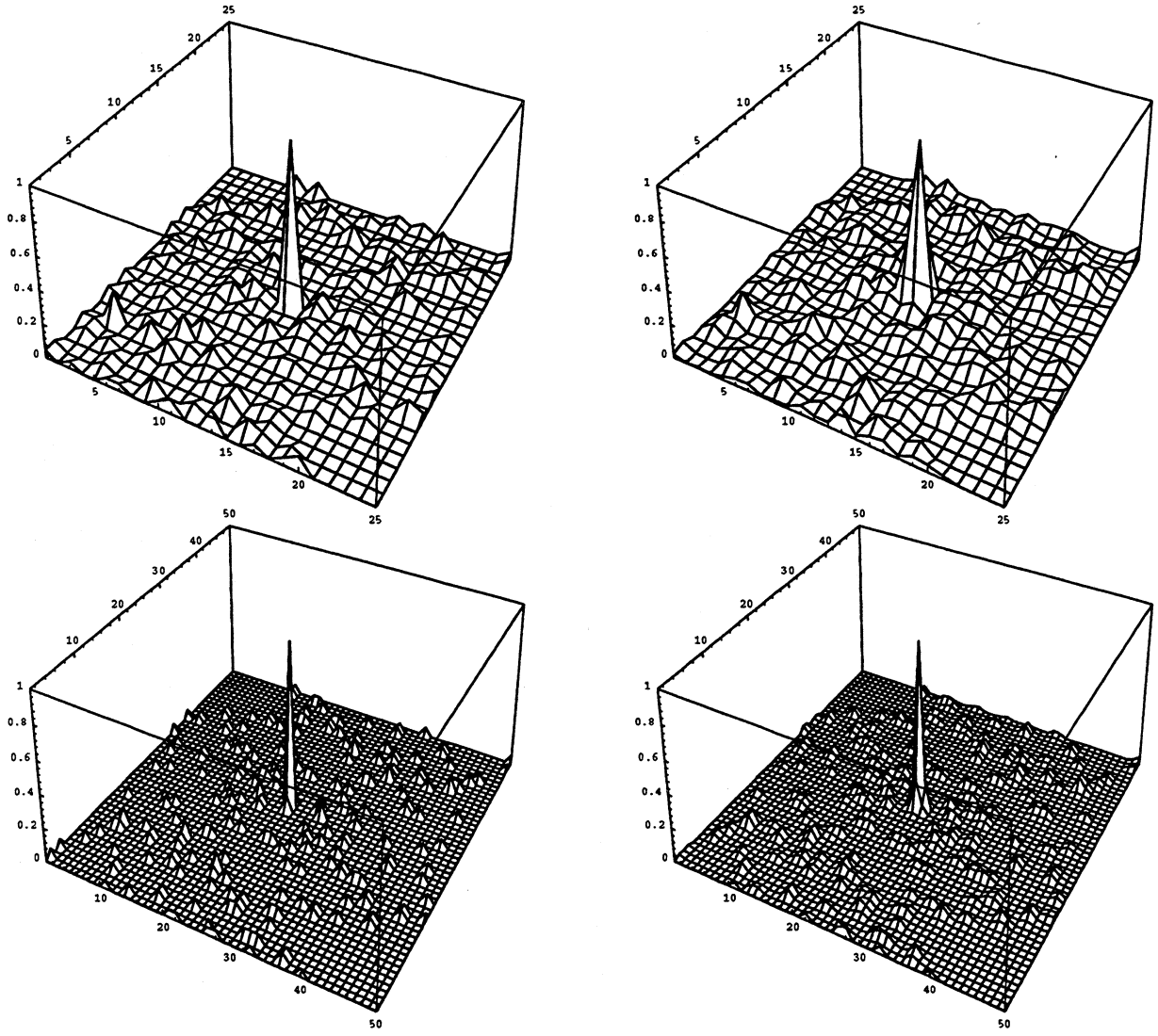

Figure 2: Comparison of GHT (*left*) with discretized objective function $-\beta F(\theta, \mathbf{t}, \beta)$ (*right*) with $\beta^{-1} = 2\epsilon_{\mathbf{t}}^2$, as a function of translation parameter $\mathbf{t}$. We only show the energy surfaces at the correct rotation angle. In (*left*), the vertical scale represents the number of votes collected at each bin, and the horizontal scales are indexes for translation bins. In (*right*), the vertical scale represents the value of $-\beta F(\theta, \mathbf{t}, \beta)$ at discrete points sampled at the center of each bin. The values are normalized so that the central spike in both plots are of height 1. Here we use 25-by-25 (*top*) bins and 50-by-50 (*bottom*) bins for translation. The point feature sets $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_a\}$ are chosen as in the experiments of section 3.

$$E'_j(\chi_j, u_j, v_j, \mathbf{t}) = \chi_j F(\hat{\theta}_j + v_j, \mathbf{t}_j) + \frac{1}{\beta}(u_j v_j - \log \frac{\sinh(u_j \epsilon_\theta)}{u_j}). \tag{25}$$

Finally, the MFT objective function (the free energy) is

$$F(\chi, \{v_j\}, \{u_j\}, \{\mathbf{t}_j\}) = \sum_j \chi_j F(\hat{\theta}_j + v_j, \mathbf{t}_j) + \frac{1}{\beta} \sum_j \left(u_j v_j - \log \frac{\sinh(u_j \epsilon_\theta)}{u_j} \right) + \mathrm{WTA}(\chi) \tag{26}$$

in which $\{u_j\}$ parameters are now real-valued. This gives us a hierarchical optimization in which an optimal rotation $\hat{\theta}_j + v_j^*$ is found for each interval $[\hat{\theta}_j - \epsilon_\theta, \hat{\theta}_j + \epsilon_\theta], j = 1, \ldots, J$, and these locally optimal rotation angles are then compared to give a globally optimal one $\hat{\theta}_{j^*} + v_{j^*}^*$. In other words, we transform a hard global optimization problem into several small local ones, and the pick the one with smallest local energy. Each bin-specific summand of F can be minimized by the following fixed point equations

$$\mathbf{t}_j = \sum_{ia} \sigma_{ia}(\mathbf{x}_i - \mathbf{R}_{\hat{\theta}_j + v_j} \mathbf{y}_a) / \sum_{ia} \sigma_{ia} \tag{27}$$

$$u_j = -\beta \sum_{ia} \sigma_{ia}(\mathbf{x}_i - \mathbf{R}_{\hat{\theta}_j + v_j} \mathbf{y}_a - \mathbf{t}_j)^t (\mathbf{R}_{\hat{\theta}_j + v_j + \frac{\pi}{2}} \mathbf{y}_a) \tag{28}$$

$$v_j = \langle \theta_j \rangle_\beta = \frac{1}{u_j} - \frac{\epsilon_\theta}{\tanh(\epsilon_\theta u_j)} = g(u_j). \tag{29}$$

Note that, in the expression for $g(u)$, the poles cancel at $u_j = 0$ and g acts like a sigmoidal transfer function that confines v_j to the interval $[-\epsilon_\theta, \epsilon_\theta]$.

2.4 Complete MFT Derivation

We start with the partition function

$$Z = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta \int_{-\infty}^{\infty} dt e^{-\beta F(\theta, \mathbf{t}, \beta) - \mathbf{t}^2 / 2\sigma_0^2}. \tag{30}$$

Let

$$\tilde{E}(\theta, \mathbf{t}) = F(\theta, \mathbf{t}, \beta) + \mathbf{t}^2 / (2\sigma_0^2 \beta) - h_\theta \theta - \mathbf{h}_t \cdot \mathbf{t} \tag{31}$$

Then Z can be used to calculate averages, e.g.

$$\langle \theta \rangle_\beta = -\frac{1}{\beta} \frac{\partial \log Z}{\partial h_\theta} \Big|_{\mathbf{h}=0} \tag{32}$$

but we will drop the $\mathbf{h}$ source term in this calculation for convenience.

Now we introduce the hierarchical representation of θ by means of a change of variables:

$$\theta = \sum_j \chi_j (\hat{\theta}_i + \theta_j) \tag{33}$$

where $\hat{\theta}_j$ is the center of bin j. Define

$$C = 2\epsilon \sum_{\{\mathrm{x}|\sum_j \chi_j=1\}} \int_{-\epsilon}^{\epsilon} (\prod_j \frac{d\theta_j}{2\epsilon}) \delta_{2\pi}(\theta - \sum_j \chi_j(\hat{\theta}_j + \theta_j)) \tag{34}$$

where $\delta_{2\pi}$ is a periodic version of Dirac δ function, which can be written as

$$\delta_{2\pi}(x) = \sum_{n=-\infty}^{\infty} \delta(x - 2\pi n) \tag{35}$$

Note that C is equal to the number of bins which overlap with θ, which we take to be an *integer constant* such as two. Now

$$\begin{aligned} Z &= \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta \frac{2\epsilon}{C} \sum_{\{\mathrm{x}|\sum_j \chi_j=1\}} \int_{-\epsilon}^{\epsilon} (\prod_j \frac{d\theta_j}{2\epsilon}) \delta_{2\pi}(\theta - \sum_j \chi_j(\hat{\theta}_j + \theta_j)) \int_{-\infty}^{\infty} dt e^{-\beta \tilde{E}(\theta_j, \mathrm{t})} \\ &= \frac{1}{2\pi C} (\frac{1}{2\epsilon})^{N-1} \sum_{\{\mathrm{x}|\sum_i \chi_j=1\}} \int_{-\epsilon}^{\epsilon} (\prod_j d\theta_j) \int_{-\infty}^{\infty} dt e^{-\beta \tilde{E}(\sum_j \chi_j(\hat{\theta}_j+\theta_j), \mathrm{t})} \end{aligned} \tag{36}$$

But

$$\begin{aligned} \int_{-\infty}^{\infty} dt e^{-\beta \tilde{E}(\sum_j \chi_j(\hat{\theta}_j+\theta_j), \mathrm{t})} &= \int_{-\infty}^{\infty} dt \sum_j \chi_j e^{-\beta \tilde{E}(\sum_j \chi_j(\hat{\theta}_j+\theta_j), \mathrm{t})} \\ &= \sum_j \chi_j \int_{-\infty}^{\infty} d\mathrm{t}_j e^{-\beta F(\hat{\theta}_j+\theta_j, \mathrm{t}_j, \beta) - \mathrm{t}_j^2/2\sigma_0^2} \\ &\quad \times (\frac{1}{2\pi\sigma_0^2})^{N-1} \int_{-\infty}^{\infty} (\prod_{k \neq i} d\mathrm{t}_k) e^{-\mathrm{t}_k^2/2\sigma_0^2} \\ &= (\frac{1}{2\pi\sigma_0^2})^{N-1} \int_{-\infty}^{\infty} (\prod_k d\mathrm{t}_k) e^{-\beta \sum_j \chi_j F(\hat{\theta}_j+\theta_j, \mathrm{t}_j, \beta) - \sum_j \mathrm{t}_j^2/2\sigma_0^2} \end{aligned} \tag{37}$$

Finally,

$$\begin{aligned} Z &= \frac{1}{2\pi C} (\frac{1}{4\pi\epsilon\sigma_0^2})^{N-1} \int_{-\epsilon}^{\epsilon} (\prod_j d\theta_j) \int_{-\infty}^{\infty} (\prod_k d\mathrm{t}_k) \sum_{\{\mathrm{x}|\sum_j \chi_j=1\}} e^{-\beta \sum_j \chi_j F(\hat{\theta}_j+\theta_j, \mathrm{t}_j, \beta) - \sum_j \mathrm{t}_j^2/2\sigma_0^2} \\ &\approx K e^{-\beta F(\{\theta_j^*, \mathrm{t}_j^*, \chi_j^*, u_j^*, w_j^*\})} \end{aligned} \tag{38}$$

where K is a constant, and the conventional Mean Field Theory (stationary phase) calculations are used to derive F (as in the previous section and [PS89, Sim90, GY91]):

$$\begin{aligned} F(\{\theta_j, \mathrm{t}_j, \chi_j, u_j, w_j\}) &= \sum_j \chi_j F(\hat{\theta}_j + \theta_j, \mathrm{t}_j, \beta) + \frac{1}{2\sigma_0^2 \beta} \sum_j \mathrm{t}_j^2 \\ &\quad + \frac{1}{\beta} \sum_j (\theta_j u_j - \log \frac{\sinh \epsilon u_j}{u_j}) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta) \end{aligned} \tag{39}$$

with

$$\mathrm{WTA}(\{\chi_j, w_j\}, \beta) \equiv \frac{1}{\beta}\Big(\sum_j \chi_j w_j - \log \sum_j e^{w_j}\Big). \tag{40}$$

As $\sigma_0 \to \infty$,

$$Z \propto e^{-\beta \tilde{F}(\{\theta_j^*, \chi_j^*, u_j^*, w_j^*\})}, \tag{41}$$

where

$$\tilde{F}(\{\theta_j, \chi_j, u_j, w_j\}) = \sum_j \chi_j F(\hat{\theta}_j + \theta_j, \mathbf{t}^*(\hat{\theta}_j + \theta_j)) + \frac{1}{\beta}\sum_j (\theta_j u_j - \log \frac{\sinh u_j}{u_j}) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta). \tag{42}$$

This is the vernier network objective function. Note that the function $\mathbf{t}^*(\hat{\theta}_j + \theta_j)$ could be replaced with independent variable $\mathbf{t}_j$, to be optimized along with θ_j. But since each term in the first summand depends quadratically on a single $\mathbf{t}_j$ (c.f. (15)), the optimal translation $\mathbf{t}_j^*$ can be solved in closed form (27).

Algorithms for optimizing (42) can be made efficient by considering the fixed-point-preserving transformation

$$\tilde{F} \;\to\; E_{\text{clocked}} \tag{43}$$

to a clocked objective function [MM93]

$$\begin{aligned} E_{\text{clocked}} \;&=\; \sum_j F(\hat{\theta}_j + \theta_j, \mathbf{t}^*(\hat{\theta}_j + \theta_j)) + \frac{1}{\beta}\sum_j (\theta_j u_j - \log \frac{\sinh u_j}{u_j}) \\ &\oplus\; \sum_j \chi_j F(\hat{\theta}_j + \bar{\theta}_j, \mathbf{t}^*(\hat{\theta}_j + \bar{\theta}_j)) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta). \end{aligned} \tag{44}$$

In this notation, the clocked sum of the form $E_1 \oplus E_2$ is equal to its first summand during phase one of an optimization cycle, and equal to its second summand during phase two. The cycle then repeats with a lower temperature. Also barred variables like $\bar{\theta}_j$ are *clamped* to constant values attained in the previous phase of the clocked objective, i.e. the previous summand of the clocked sum denoted by $\oplus$. The first clocked summand of (44) represents a set of noninteracting networks, one per rotation bin. The second term is a pure winner-take-all network with constant coefficients, and as such can be implemented by digital logic rather than by an analog neural network. We implemented the vernier network this way in the computer experiments reported in section 4.

3 The Filtered Vernier Networks

If we have enough vernier bins, the vernier variables will be close to zero and hence can be dropped from the effective objective function

$$\sum_j \chi_j F(\hat{\theta}_j + \theta_j, \mathbf{t}^*(\hat{\theta}_j + \theta_j)) + \frac{1}{\beta}\sum_j (\theta_j u_j - \log \frac{\sinh u_j}{u_j}) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta) \tag{45}$$

$$\approx\; \sum_j \chi_j F(\hat{\theta}_j, \mathbf{t}^*(\hat{\theta}_j)) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta), \tag{46}$$

and the goal is to find

$$\text{argmin}_j F(\hat{\theta}_j, \mathbf{t}^*(\hat{\theta}_j)). \tag{47}$$

If we relate vernier bins to rotation bins in GHT, the optimal translation for each principle rotation $\hat{\theta}_j$ can be found as

$$\mathbf{t}^*(\hat{\theta}_j) \;=\; \text{argmin}_{\mathbf{t}} F(\hat{\theta}_j, \mathbf{t}(\hat{\theta}_j), \beta). \tag{48}$$

As in (19), $\mathbf{t}^*(\hat{\theta}_j)$ can be approximated by $\hat{\mathbf{t}}_{k^*,l^*}$, where $(k^*, l^*) = \text{argmax}_{kl}\text{GHT}(j,k,l)$. So we see that GHT is a two-stage approximation to the vernier network. That is, GHT is obtained by first replacing $\hat{\theta}_j + \theta_j$ with $\hat{\theta}_j$, and then sampling the translation space.

The generalized Hough transform has the advantage of being able to find the optimal solution if data are noiseless and we have a infinitely fine partition of the pose space, and would fail if either of these conditions is not satisfied. Because of the speed of GHT at coarse scale, we can use the GHT as a preprocessing *filter* for the vernier network, running the vernier network only on those bins selected by the filter and starting at a much higher temperature than usual, namely twice the effective temperature of the GHT filter. As in (44), we can derive a clocked objective function for filtered vernier network

$$\begin{aligned} F(\theta, \mathbf{t}(\theta)) \;\rightarrow\; & \sum_j \chi_j F(\hat{\theta}_j, \mathbf{t}^*(\hat{\theta}_j)) + \text{kWTA}(\{\chi_j | 0 \le j \le J-1\}, k) \\ & \oplus \sum_j F(\hat{\theta}_j + \theta_j\{\chi_j\}, \mathbf{t}^*(\hat{\theta}_j + \theta_j\{\chi_j\})) + \frac{1}{\beta}\sum_j \varphi_{\text{MFT}}(\theta_j, n_j) \\ & \oplus \sum_j n_j\{\chi_j\} F(\hat{\theta}_j + \bar{\theta}_j, \mathbf{t}^*(\hat{\theta}_j + \bar{\theta}_j)) + \text{WTA}(\{n_j\{\chi_j | 0 \le j \le J-1\}\}) \end{aligned} \tag{49}$$

where the notation $\theta\{\chi\}$ stands for

$$\chi\theta + (1-\chi)\bar{\theta}. \tag{50}$$

This notation represents the "filtering" mechanism by which only those θ_j that are selected ($\chi_j = 1$) will be further updated, and those not selected will simply take clamped values. We have already shown that (49) can be approximated by the GHT. The binning transformation (49) could be applied recursively to

$$F(\hat{\theta}_j + \theta_j\{\chi_j\}, \mathbf{t}^*(\hat{\theta}_j + \theta_j\{\chi_j\})) \tag{51}$$

for each j with $\chi_j \neq 1$.

4 Experimental Results and Conclusions

A typical matching example by a vernier network with just one rotation bin is shown in Figure 3. An instance of the model is created in the scene with a rotation angle randomly chosen from the interval $[-\pi/4, \pi/4]$. The vernier network uses a bin of width π centered on zero rotation. The

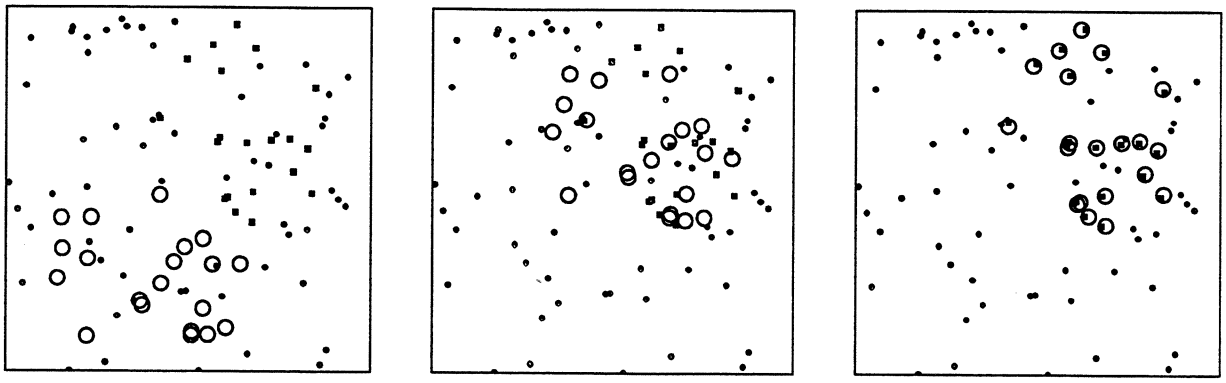

Figure 3: Matches a 20-point model to a scene with 66.7% spurious outliers. An instance of the model is created in the scene with a rotation angle randomly chosen from the interval $[-\pi/4, \pi/4]$. The vernier network uses a bin of width π centered on zero rotation. Shown in leftmost, middle, and rightmost are configurations at the first, tenth, and fifty-first annealing steps, respectively. In each plot, a projection of the model with its currently estimated pose on the scene is represented by the circles. The instance of the model in the scene is represented by squares. Other small dots are spurious scene points.

model moves through the scene as the optimization proceeds. The estimated poses converge to the correct one. We observe that the MFT network is able to filter out spurious scene points, and gradually locates those scene points that correspond to model points.

In the numerical experiments, we tested four algorithms: plain MFT (or relaxation network) without bins, GHT, vernier network, and filtered vernier network. Four setting of the bin size for the Generalized Hough transform were used: we divided the three-dimensional parameter space into 8×8×8, 16×16×16, 32×32×32, and 64×64×64 bins. GHT bin size was determined according to the noise level of the data. For higher noise, larger bins were used to account for the uncertainty in the measurement; this was done by choosing the smallest error result for each noise level. For the plain MFT, we chose the center of the whole pose space as the initial pose. Each simulation began with temperature 0.5, and decreased the temperature until $\sum_{ia} \sigma_{ia} \approx N$, by successive factors of 0.9. The vernier network had 8 bins on the rotation angle. To treat cases in which the correct pose falls near a boundary between bins, the bins were overlapped with each other by a factor of two with bin centers spaced uniformly on the unit circle. The stopping criterion was the same as that for plain MFT. In the filtered vernier network, a 32×32×32 GHT was used as a filter to select bins. The *single* bin with highest energy at its center was chosen for further processing with the vernier network, although for larger Hough parameter spaces one may need to examine more candidate bins with the vernier net. All the MFT algorithms used analog gradient descent (using Hopfield/Grossberg dynamics), simulated with the forward Euler method, to minimize their objective function. We note that other relaxation algorithms (such as conjugate gradient method) may be more efficient for use on general-purpose computers.

For each trial in our experiments, the model consists of 20 two-dimensional points generated uniformly in the square region whose coordinates are in $[-0.5, 0.5]\times[-0.5, 0.5]$. A rotation angle was chosen from the interval $[0, 2\pi]$ and a translation vector was chosen from the square $[-0.5, 0.5]\times[-0.5, 0.5]$, both according to a uniform distribution. Outliers were generated uni-

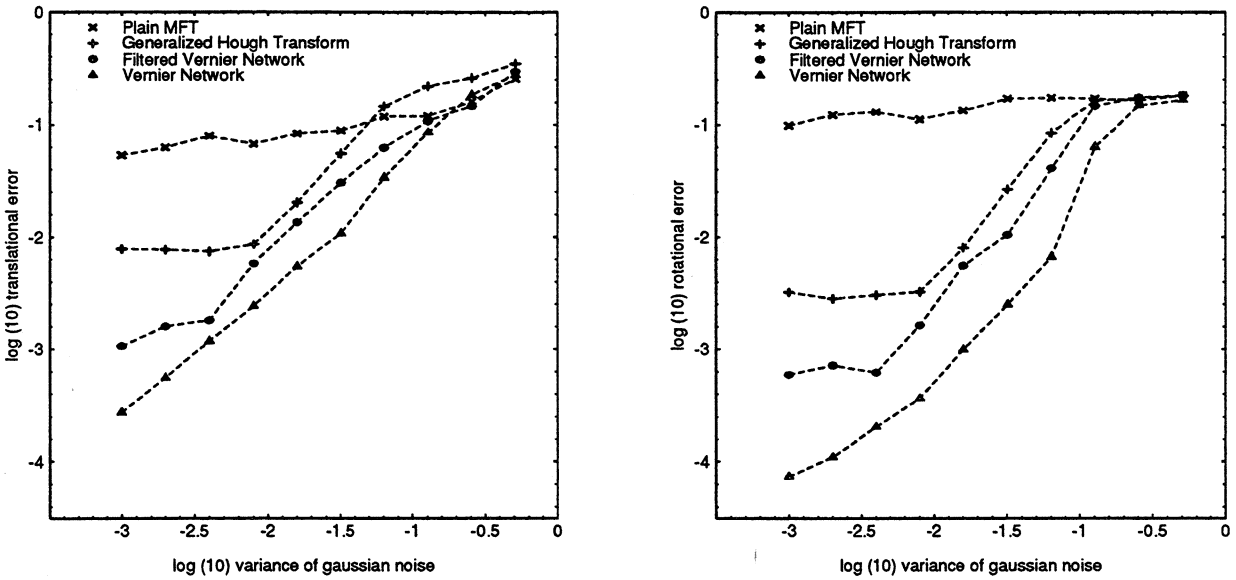

Figure 4: Comparison of the performance of plain MFT network, generalized Hough transform (GHT), vernier network, and filtered vernier network. The average of the logarithm of each algorithm's error is taken over 500 runs for GHT and 100 runs for the other algorithms. They are plotted as functions of the logarithm of the variance of the Gaussian displacement noise added to the scene. The GHT data points are each the best (minimal error) of the four binning schemes discussed in the text. 50% of the scene points are randomly generated outliers. Similar relative performance results have been obtained for 66.7% outliers (see Figure 5). The average running times on a Sparcstation 2 computer were observed to be $32 \pm .04$ sec per run for the plain MFT network, 22 ± 1.3 sec for the $64 \times 64 \times 64$ GHT, 35 ± 6 sec for the $32 \times 32 \times 32$ filtered vernier network, and 331 ± 23 sec for the 8-bin vernier network.

formly in the square $[-1,1]\times[-1,1]$, and were presented in numbers equal to, or double, the number of model points. Independent Gaussian noise was added to the rotated and translated model points, and the variance of this noise was varied by powers of 2 between 0.001 and 0.512. Running times were measured on a Sparcstation 2 computer, and averaged over ten runs of each algorithm. The results are shown on a logarithmic scale in Figure 4 and 5. We can see that (a) the plain MFT method performs with unacceptably high error by comparison with the other algorithms, particularly for the rotation parameter, (b) the vernier network has a much better performance than GHT, though at substantially higher cost, and (c) the filtered vernier network has intermediate performance at a cost comparable to that of the GHT code.

In a separate experiments (Figure 6 and 7), we compared filtered vernier networks with 1, 2, 3, and 4 vernier bins, which use the same $32 \times 32 \times 32$ GHT filter as in the previous experiment. The filtered vernier networks with more vernier bins achieve better performance at the expense of increased running time.

In the absence of further developments, the choice among these methods would depend on the desired tradeoff between accuracy and computational cost, except that the plain MFT algorithm is rarely optimal. However, our performance measurements are significantly more refined than our cost measurements because the implementation codes for the vernier networks can be sped up substantially; this seems not to be the case for the GHT code (except by making essential

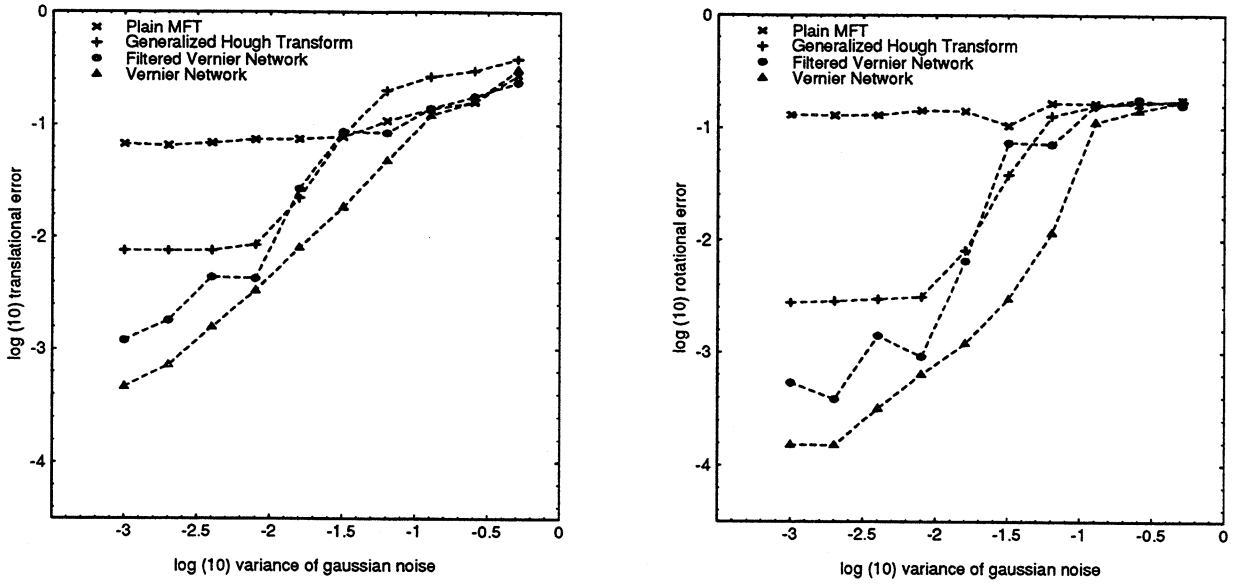

Figure 5: Comparison of the performance of plain MFT network, generalized Hough transform (GHT), vernier network, and filtered vernier network. 66.7% of the scene points are randomly generated outliers.

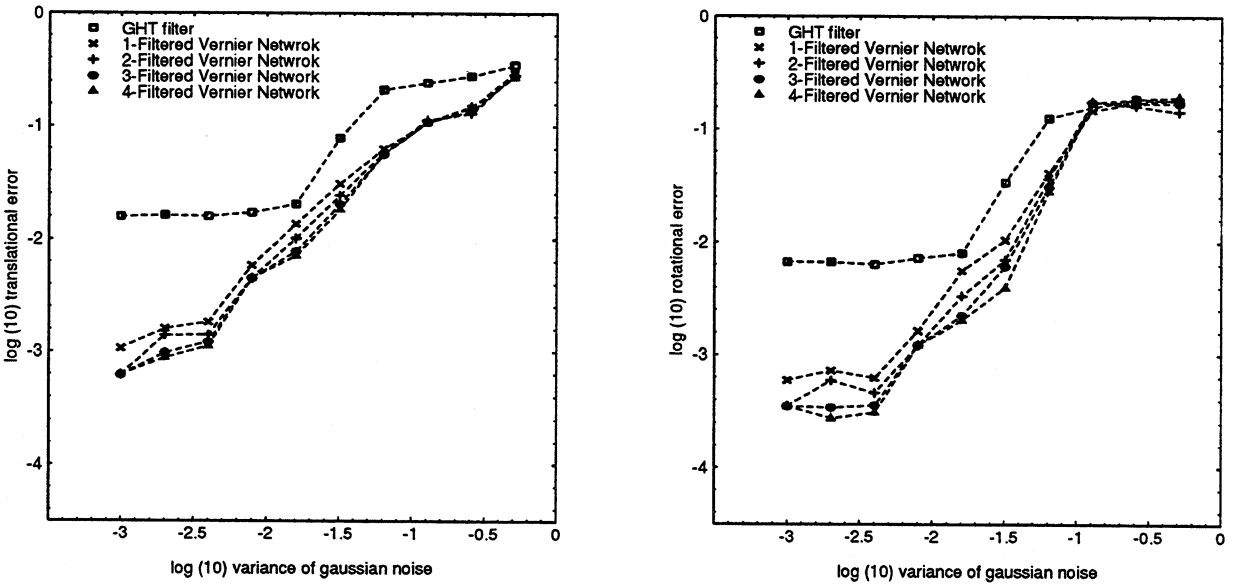

Figure 6: Comparison the performance of GHT without post-processing and the filtered vernier networks with 1, 2, 3, and 4 vernier bins. The average of the logarithm of each algorithm's error is taken over 500 runs for GHT filters without post-processing and 100 runs for filtered vernier networks. 50% of the scene points are randomly generated outliers.

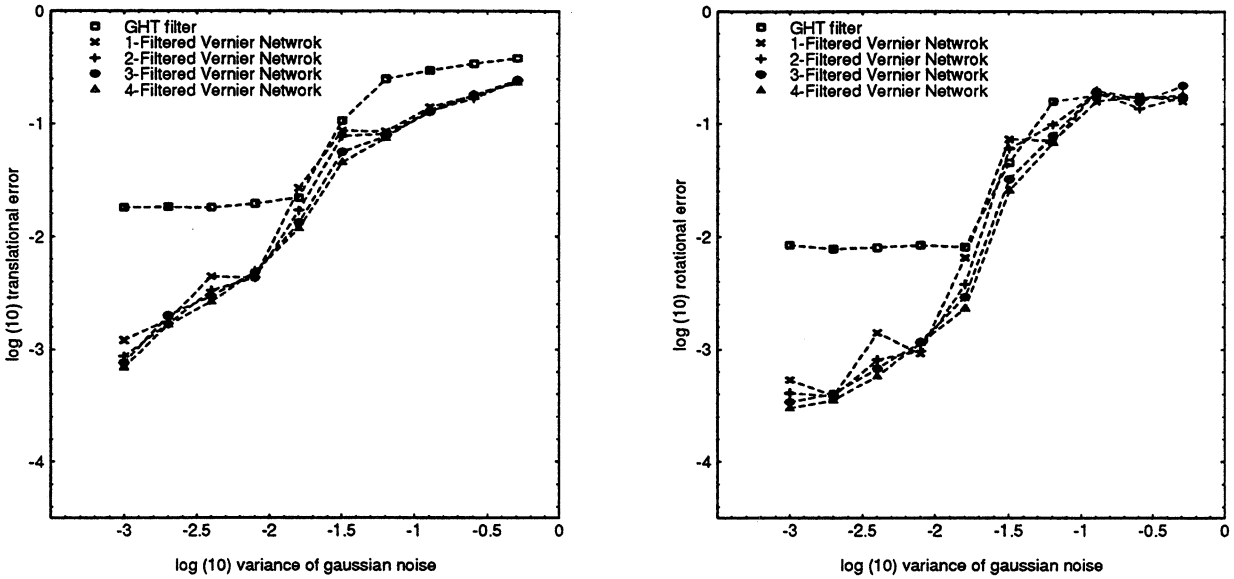

Figure 7: Comparison of the performance of GHT without post-processing and the filtered vernier networks with 1, 2, 3, and 4 vernier bins. 77.6% of the scene points are randomly generated outliers.

changes to the underlying algorithm). The likely effect of speeding up the vernier net code would be to make the vernier and filtered vernier networks (still) more competitive relative to the GHT and to increase the optimal number of vernier bins in the filtered vernier net.

5 Acknowledgment

We thank Anand Rangarajan for providing discussions regarding the MFT formulation of the problems.

References

[AF86] N. Ayache and O. D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.

[Bai84] Henry S. Baird. *Model-Based Image Matching Using Location.* The MIT Press, Cambridge, Massachusetts, first edition, 84.

[Cas88] T. A. Cass. Parallel computation in model-based recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 879–884, 1988.

[GLP87] W. E. L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9:469–482, 1987.

[Gri90] W. E. L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. *Artificial Intelligence*, (44):121–166, 1990.

[GY91] D. Geiger and A. L. Yuille. A common framework for image segmentation. *International Journal of Computer Vision*, 6(3):227–243, August 1991.

[KY91] J. J. Kosowsky and A. L. Yuille. The invisible hand algorithm: Solving the assignment problem with statistical physics. Technical Report 91-1, Havard Robotics Laboratory, 1991.

[MG90] Eric Mjolsness and Charles Garrett. Algebraic transformations of objective functions. *Neural Networks*, 3:651–669, 1990.

[Mjo91] Eric Mjolsness. Bayesian inference on visual grammars by neural nets that optimize. Technical Report YALEU/DCS/TR-854, Yale Computer Science Department, May 1991. (Temporarily available on neuroprose as "mjolsness.grammar.ps.Z").

[MM93] Eric Mjolsness and Willard L. Miranker. Greedy Lagrangians for neural networks: Three levels of optimization in relaxation dynamics. Technical Report YALEU/DCS/TR-945, Yale Computer Science Department, January 1993.

[PS89] Carsten Peterson and Bo Söderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1):3–22, 1989.

[Sim90] Petar D. Simic. Statistical mechanics as the underlying theory of 'elastic' and neural optimisations. *Network*, 1:89–103, 1990.

[Sto87] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, (40), 1987.

[Yui90] Alan L. Yuille. Generalized deformable models, statistical physics, and matching problems. *Neural Computation*, 2(1):1–24, 1990.