Mean Field Point Matching by Vernier Network and by Generalized Hough Transform: Preliminary Report

Chien-Ping Lu
Eric Mjolsness

Research Report YALEU/DCS/TR-949
February 1993

Mean Field Point Matching by Vernier Network and by Generalized Hough Transform: Preliminary Report

Chien-Ping Lu
Eric Mjolsness
Yale Department of Computer Science
P.O. Box 2158 Yale Station
New Haven, CT 06520-2158

February 2, 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.

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 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 the such point matching problems is essential to object recognition

Supported in part by ONR grant N00014-92-J-4048, and by AFOSR grant AFOSR-88-0240.

and two-frame rigid motion estimation problems, assuming the relevant images have each been preprocessed into a sparse set of significant features.

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.

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

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

where $\{M_{ia}\} = \mathbf{M}$ is a permutation matrix (called the "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. Minimizing (1) over permutations, rotations and translations is a *parametric assignment problem* which differs from other assignment problems in that the cost matrix is determined by continuous variables also subject to optimization. As an optimization problem, (1) has enough spurious local minima that many descent methods are prone to error.

Several approaches have been proposed to solve this optimization problem. They differ in the ways they handle the point correspondence components of the problem. For example *tree pruning methods* [Bai84, GLP87, AF86] make hypotheses concerning the point correspondence by searching over a tree in which each node represents a partial match. Each partial match is then evaluated through the pose that best fits it. The *parametric linear programming* approach [ZS92] instead makes hypotheses concerning the pose parameters first. For each pose hypothesis $(\theta, \mathbf{t})$, we need to solve an assignment problem with the constant cost matrix $\{C_{ia}(\theta, \mathbf{t})\}$. The solutions to these assignment problems are then compared to choose the best pose hypothesis. In the *generalized Hough transform* approach [Bal87, GLP87], a set of optimal poses is computed for each possible pairing of a model point and a scene point, and all the selected optimal poses then "vote" for the closest candidate among a set of discretized poses. An implicit hypothesis in this approach is that each model point has an equal chance to be matched to each scene point.

By contrast with these relatively standard methods, we propose to optimize (1) directly 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 [MM93]. 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]

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

(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 hierachical representation of the continuous geometric variables, which in effect can be though of as a binning transformation that turns the original optimization problem over a single pose space into several optimization problems over smaller Cartesian product sets (the bins). Initially, the centers of each Cartesian product set 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.

2 The Theory

2.1 The MFT approach

To solve point matching problem by minimizing (1), we need to enforce some constraints on M_{ia}, otherwise the objective function can always be minimized with all M_{ia} set to zero. The constraint on M for standard assignment problems is $\sum_i M_{ia} = 1, \forall a$ and $\sum_a M_{ia} = 1, \forall i$, i.e. that M is a permutation matrix. In more general cases where there are spurious scene points, we can use the weaker *multinomial constraint* $\sum_{ia} M_{ia} = N$, which implies that there are exactly N matches among all possible matches. In the following, we designate the allowed set of matrices M that satisify the given constraints (whatever they are) by $\mathcal{M}$.

Assume a Gibbs distribution for the scene points $\mathbf{x}_i, \ldots, \mathbf{x}_N$

$$f_\beta(\{\mathbf{x}_i\} | \mathbf{M}, \theta, \mathbf{t}) \;=\; \frac{1}{Z_\beta} e^{-\beta E_{\text{eff}}(\mathbf{M}, \theta, \mathbf{t})} \tag{4}$$

where $Z_\beta = \sum_{\mathbf{M} \in \mathcal{M}} e^{-\beta E_{\text{eff}}(\mathbf{M}, \theta, \mathbf{t})}$ is the normalization factor known as *partition function* in statistical physics, and β is the inverse temperature. The contribution of M to the partition function can be exactly or approximately evaluated leaving an effective objective function $E_{eff} = -\frac{1}{\beta} \log Z_\beta$ depending on the pose parameters only. The form of E_{eff} depends on the constraints $\mathcal{M}$ with which the partition function is evaluated. [Mjo91] provides a way to approximate the summation over M subject to the multinomial constraint, and gives the effective objective function

(3). The descent dynamics for finding the saddle point of this objective is

$$\begin{aligned} \dot{\mathbf{t}} &= -\kappa \sum_{ia} m_{ia}(\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}) \\ \dot{\theta} &= -\kappa \sum_{ia} m_{ia}(\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t})^t (\mathbf{R}_{\theta+\frac{\pi}{2}} \mathbf{y}_a) \\ m_{ia} &= \exp -\beta \|\mathbf{x}_i - \mathbf{R}_\theta \mathbf{y}_a - \mathbf{t}\|^2. \end{aligned} \tag{5}$$

The basic premises of the 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 approach the optimal $\mathbf{M}^*$. Therefore, by tracking the saddle point 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.

2.2 The Vernier Network for Rotations

Given the constraints on match variables and the corresponding effective objective function, we can focus on finding the optimal pose parameters. Though the effective objective is non-convex over translation at low temperatures, its dependence on rotation is non-convex even at relatively high temperature. We propose overcoming this problem by applying MFT to a hierachical representation of rotation

$$\theta = \sum_{j=0}^{J-1} \chi_j(\hat{\theta}_j + \theta_j),\ \theta_j \in [-\epsilon_\theta, \epsilon_\theta], \qquad \left(\text{and } \mathbf{t} = \sum_{j=0}^{J-1} \chi_j \mathbf{t}_j\right) \tag{6}$$

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

The full Mean Field Theory 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)

$$Z = \sum_{\{\chi | \sum_j \chi_j = 1\}} \prod_j \int_{[-\epsilon_\theta, \epsilon_\theta]} d\theta_j e^{-\beta \chi_j E_{eff}(\hat{\theta}_j + \theta_j, \mathbf{t}_j)} \tag{7}$$

The contribution of each θ_j to the partition function can be evaluated as

$$\begin{aligned} &\int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j e^{-\beta \chi_j E_{eff}(\hat{\theta}_j + \theta_j, \mathbf{t}_j)} = \int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j \int dv_j e^{-\beta \chi_j E_{eff}(\hat{\theta}_j + v_j, \mathbf{t}_j)} \delta(v_j - \theta_j) \\ &= \int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j \int dv_j \int_I du_j e^{-\beta \chi_j E_{eff}(\hat{\theta}_j + v_j, \mathbf{t}_j)} e^{-u_j(v_j - \theta_j)} \\ &= \int_{\theta_j \in [-\epsilon_\theta, \epsilon_\theta]} d\theta_j e^{u_j \theta_j} \int dv_j \int_I du_j e^{-\beta \chi_j E_{eff}(\hat{\theta}_j + v_j, \mathbf{t}_j)} e^{-u_j v_j} \\ &= \frac{2}{u_j} e^{-u_j \hat{\theta}_j} \sinh(u_j \epsilon_\theta) \int dv_j \int_I du_j e^{-\beta \chi_j E_{eff}(\hat{\theta}_j + u_j, \mathbf{t}_j)} e^{-u_j v_j}. \end{aligned} \tag{8}$$

Combining these results gives

$$Z = 2^N \sum_{\chi} \prod_j \int dv_j \int_I du_j e^{-\beta E'_j(\chi_j, u_j, v_j, \mathbf{t}_j)} \tag{9}$$

$$E'_j(\chi_j, u_j, v_j, \mathbf{t}) = \chi_j E_{\mathit{eff}}(\hat{\theta}_j + v_j, \mathbf{t}_j) + \frac{1}{\beta}(u_j v_j - \log \frac{\sinh(u_j \epsilon_\theta)}{u_j}). \tag{10}$$

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

$$F(\chi, \{v_j\}, \{u_j\}, \{\mathbf{t}_j\}) = \sum_j \chi_j E_{\mathit{eff}}(\hat{\theta}_j + v_j, \mathbf{t}_j) + \frac{1}{\beta} \sum_j \left(u_j v_j - \log \frac{\sinh(u_j \epsilon_\theta)}{u_j}\right) + \mathrm{WTA}(\chi). \tag{11}$$

This gives us a hierachical optimization in which an opimal 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 (under the hopeful assumption that its $\chi_j \approx 1$) by the following fixed point equations

$$\begin{aligned} \mathbf{t}_j &= \sum_a m_{ia}(\mathbf{x}_i - \mathbf{R}_{v_j} \mathbf{y}_a) \\ u_j &= -\beta \sum_{ia} m_{ia}(\mathbf{x}_i - \mathbf{R}_{v_j} \mathbf{y}_a - \mathbf{t}_j)^t (\mathbf{R}_{v_j + \frac{\pi}{2}} \mathbf{y}_a) \\ v_j &= \langle \theta_j \rangle_\beta = \frac{1}{u_j} - \frac{\epsilon_\theta}{\tanh(\epsilon_\theta u_j)} = g(u_j). \end{aligned} \tag{12}$$

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

2.3 Complete MFT Derivation

We start with the partition function

$$Z = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta \int_{-\infty}^{\infty} dt e^{-\beta E_{\mathit{eff}}(\theta, \mathbf{t}, \beta) - \mathbf{t}^2 / 2\sigma_o^2} \tag{13}$$

where

$$E_{\mathit{eff}} = -\frac{1}{\beta} \log \sum_{ia} e^{-\beta \|\mathbf{x}_i - \mathbf{y}_a - \mathbf{t}\|^2}. \tag{14}$$

Let

$$\tilde{E}(\theta, \mathbf{t}) = E_{\mathit{eff}}(\theta, \mathbf{t}, \beta) + \mathbf{t}^2 / (2\sigma_0^2 \beta) - h_\theta \theta - \mathbf{h}_t \cdot \mathbf{t} \tag{15}$$

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

$$\langle \theta \rangle_\beta = -\frac{1}{\beta} \frac{\partial \log Z}{\partial h_\theta}\Big|_{\mathbf{h}=0} \tag{16}$$

but we will drop the **h** source term in this calculation for convenience.

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

$$\theta = \sum_j \chi_j(\hat{\theta}_j + \theta_j) \tag{17}$$

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

$$C = 2\epsilon \sum_{\{\chi|\sum_j \chi_j=1\}} \int_{-\epsilon}^{\epsilon} (\prod_j \frac{d\theta_j}{2\epsilon}) \delta_{2\pi}(\theta - \sum_j \chi_j(\hat{\theta}_j + \theta_j)) \tag{18}$$

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

$$\delta_{2\pi}(x) = \sum_{n=-\infty}^{\infty} \delta(x - 2\pi n) \tag{19}$$

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

$$\begin{aligned} Z &= \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta \frac{2\epsilon}{C} \sum_{\{\chi|\sum_j \chi_j=1\}} \int_{-\epsilon}^{\epsilon} (\prod_j \frac{d\theta_j}{2\epsilon}) \delta_{2\pi}(\theta - \sum_j \chi_j(\hat{\theta}_j + \theta_j)) \int_{-\infty}^{\infty} dt e^{-\beta \tilde{E}(\theta_j, t)} \\ &= \frac{1}{2\pi C} (\frac{1}{2\epsilon})^{N-1} \sum_{\{\chi|\sum_i \chi_j=1\}} \int_{-\epsilon}^{\epsilon} (\prod_j d\theta_j) \int_{-\infty}^{\infty} dt e^{-\beta \tilde{E}(\sum_j \chi_j(\hat{\theta}_j + \theta_j), t)} \end{aligned} \tag{20}$$

But

$$\begin{aligned} \int_{-\infty}^{\infty} dt e^{-\beta \tilde{E}(\sum_j \chi_j(\hat{\theta}_j + \theta_j), t)} &= \int_{-\infty}^{\infty} dt \sum_j \chi_j e^{-\beta \tilde{E}(\sum_j \chi_j(\hat{\theta}_j + \theta_j), t)} \\ &= \sum_j \chi_j \int_{-\infty}^{\infty} dt_j e^{-\beta E_{eff}(\hat{\theta}_j + \theta_j, t_j, \beta) - t_j^2/2\sigma_0^2} \\ &\quad \times (\frac{1}{2\pi\sigma_0^2})^{N-1} \int_{-\infty}^{\infty} (\prod_{k \neq j} dt_k) e^{-t_k^2/2\sigma_0^2} \\ &= (\frac{1}{2\pi\sigma_0^2})^{N-1} \int_{-\infty}^{\infty} (\prod_k dt_k) e^{-\beta \sum_j \chi_j E_{eff}(\hat{\theta}_j + \theta_j, t_j, \beta) - \sum_j t_j^2/2\sigma_0^2} \end{aligned} \tag{21}$$

Finally,

$$\begin{aligned} Z &= \frac{1}{2\pi C} (\frac{1}{4\pi\epsilon\sigma_0^2})^{N-1} \int_{-\epsilon}^{\epsilon} (\prod_j d\theta_j) \int_{-\infty}^{\infty} (\prod_k dt_k) \sum_{\{\chi|\sum_j \chi_j=1\}} e^{-\beta \sum_j \chi_j E_{eff}(\hat{\theta}_j + \theta_j, t_j, \beta) - \sum_j t_j^2/2\sigma_0^2} \\ &\approx K e^{-\beta F(\{\theta_j^*, t_j^*, \chi_j^*, u_j^*, w_j^*\})} \end{aligned} \tag{22}$$

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

$$F(\{\theta_j, \mathbf{t}_j, \chi_j, u_j, w_j\}) = \sum_j \chi_j E_{eff}(\hat{\theta}_j + \theta_j, \mathbf{t}_j, \beta) + \frac{1}{2\sigma_0^2 \beta} \sum_j \mathbf{t}_j^2$$
$$+ \frac{1}{\beta} \sum_j (\theta_j u_j - \log \frac{\sinh \epsilon u_j}{u_j}) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta) \tag{23}$$

with

$$\mathrm{WTA}(\{\chi_j, w_j\}, \beta) = \frac{1}{\beta} \Big(\sum_j \chi_j w_j - \log \sum_j e^{w_j} \Big). \tag{24}$$

As $\sigma_0 \to \infty$,

$$Z \propto e^{-\beta \tilde{F}(\{\theta_j^*, \chi_j^*, u_j^*, w_j^*\})}, \tag{25}$$

where

$$\tilde{F}(\{\theta_j, \chi_j, u_j, w_j\}) = \sum_j \chi_j E_{eff}(\hat{\theta}_j + \theta_j, t^*(\theta_j)) + \frac{1}{\beta} \sum_j (\theta_j u_j - \log \frac{\sinh u_j}{u_j}) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta). \tag{26}$$

In our simulations, the optimal value $t^*(\theta_j)$ is approximated by solving $\dot{\mathbf{t}} = 0$ in equation (5) for $\mathbf{t}$, holding $\{m_{ia}\}$ fixed. This is very easy to do, but at greater computational expense one could iteratively solve for $t^*(\theta_j)$ by updating $\{m_{ia}(t)\}$ as well.

There is a fixed-point-preserving transformation of $\tilde{F}$:

$$\tilde{F} \;\to\; E_{\mathrm{clocked}}$$

to a clocked objective function [MM93]

$$E_{\mathrm{clocked}} = \sum_j E_{eff}(\hat{\theta}_j + \theta_j, t^*(\theta_j)) + \frac{1}{\beta} \sum_j (\theta_j u_j - \log \frac{\sinh u_j}{u_j})$$
$$\oplus \; - \sum_j \chi_j E_{eff}(\hat{\theta}_j + \bar{\theta}_j, t^*(\bar{\theta}_j)) + \mathrm{WTA}(\{\chi_j, w_j\}, \beta). \tag{27}$$

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 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 it this way in the computer experiments reported in section 3.