M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 344
To appear: 1995 IEEE Workshop on Neural Networks for Signal Processing

Motion Estimation and Segmentation Using a Recurrent Mixture of Experts Architecture

Yair Weiss and Edward. H. Adelson
Dept. of Brain and Cognitive Sciences and the Media Lab
MIT E15-384, Cambridge, MA 02139, USA
{yweiss,adelson}@media.mit.edu

Abstract

Estimating motion in scenes containing multiple motions remains a difficult problem for computer vision. Here we describe a novel recurrent network architecture which solves this problem by simultaneously estimating motion and segmenting the scene. The network is comprised of locally connected units which carry out simple calculations in parallel. We present simulation results illustrating the successful motion estimation and rapid convergence of the network on real image sequences.

1 Introduction

Motion estimation is an ill-posed problem. In other words, local motion measurements are inherently ambiguous. When the scene contains only one smoothly varying motion the ill posedness can be overcome by imposing a smoothness constraint on the solution (e.g. [Poggio et al., 1985]). The smoothness assumption, however, is not valid when the scene contains multiple motions, and imposing it leads to erroneous motion estimates especially at occlusion boundaries (e.g. [Horn, 1986]). One way to modify the smoothness assumption is to estimate motion discontinuities via line processes and disable motion smoothing across the line processes [Terzopoulos, 1986, Hutchinson et al., 1988]. These algorithms are notoriously slow to converge, and more importantly they produce a representation which is ill suited for dealing with scenes containing occlusion, such as a scene showing a cat walking behind a fence. Motion discontinuities can capture the fact that the cat fragments and the fence posts are not moving together, but they can not capture the fact that the cat fragments move together. In contrast, the representation we are interested in computing explicitly groups the fragments together [Wang and Adelson, 1994, Darrell and Pentland, 1991, Black and Anandan, 1993].

2 Architecture

Our architecture is based on the "divide and conquer" modularity principle [Jordan and Jacobs, 1994]. Rather than have one network estimate motion everywhere, we have multiple *motion expert* subnetworks competing to explain the data by minimizing *motion error.* The error signal to these expert subnetworks is controlled by a *gating* subnetwork which assigns different regions of space to different experts. The advantage of this approach is that it restores the validity of the smoothness assumption: regions undergoing drastically different motions are assigned to different experts, and the motion of regions assigned to a specific expert is indeed smoothly varying. The network simultaneously estimates the motions and the assignments. The assignment is based on two factors: (1) which expert is currently doing a better job of explaining the motion data, and (2) the current assignment of nearby regions having similar intensities. As shown below this simultaneous estimation and segmentation is accomplished using simple parallel updates.

The architecture and flow of information are depicted schematically in figure 1. The motion expert subnetwork is comprised of K sheets of retinotopically organized units (K represents the maximum number of motions in the scene). Each sheet contains units tuned for a specific velocity at a particular retinal location (cf. [Bulthoff et al., 1989]). The distribution of responses of all velocity tuned units at a given location represents the velocity estimate of the motion expert. The input to the motion experts comes from the motion error subnetwork, which also contains units tuned for a specific velocity at a particular retinal location. The exact form of these motion selective units is irrelevant, as long as they represent the local deviation from coherent motion in a given velocity. The calculation can be based on correlation as in [Bulthoff et al., 1989] or motion energy as in [Simoncelli et al., 1991]. The input from the motion error to a sheet in the motion experts subnetwork is modulated by a corresponding sheet in the gating subnetwork. The gating subnetwork, in turn, receives input from the experts and motion error subnetworks as well as a *local intensity* subnetwork which modulates the cooperation of nearby gating network units.

3 Dynamics

We denote by $\theta_k(i,j,l)$ the activity of a unit in the kth sheet of the motion experts network at grid location i,j tuned to velocity l, and by $E(i,j,l)$ the activity of a unit in the motion error network. Similarly, we denote by $G_k(i,j)$ the activity of unit i,j in the kth sheet of the gating network and by $I(i,j)$ the activity of a unit in the local intensity network. To emphasize the connection to the EM algorithm we call the dynamics of the gating and expert networks the E and M dynamics respectively.

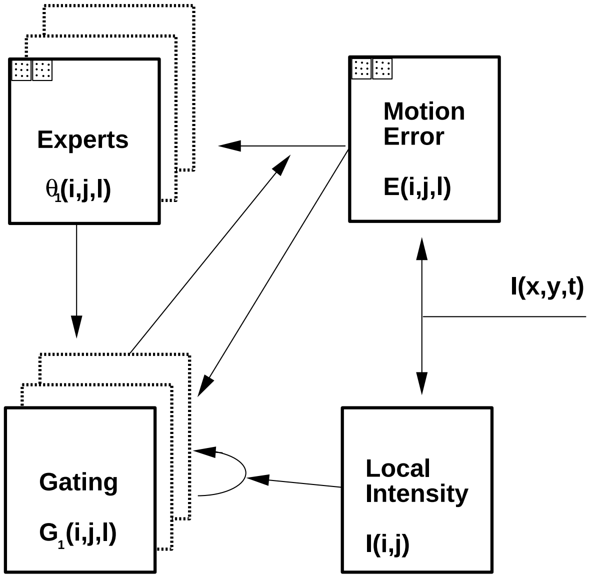

Figure 1: A schematic depiction of the information flow in the network. In accordance with the "divide and conquer" modularity principle, we have multiple motion expert subnetworks competing to explain the data by minimizing motion error. The error signal to the expert networks is modulated by a gating subnetwork which assigns different regions of space to different experts. The assignment is based on the motion error of each expert's estimate as well as the current assignment of neighboring regions with similar intensities. The simultaneous estimation of motion and assignments is accomplished by retinotopically organized units which carry out simple operations in parallel

E dynamics The gating units are updated by a weighted summation of inputs followed by a normalizing nonlinearity:

$$G_k(i,j) := \frac{exp(\tilde{G}_k(i,j))}{\sum_o exp(\tilde{G}_o(i,j))} \tag{1}$$

With:

$$\tilde{G}_k(i,j) = 1/\beta_2 \sum_l E(i,j,l)\theta_k(i,j,l) - \eta \sum_{m,n} \alpha_{ij}^{mn} G_k(m,n) \tag{2}$$

M dynamics Similarly the motion expert units are updated by a weighted summation of inputs followed by a normalizing nonlinearity:

$$\theta_k(i,j,l) := \frac{exp(\tilde{\theta}_k(i,j,l))}{\sum_o exp(\tilde{\theta}_k(i,j,o))} \tag{3}$$

With:

$$\tilde{\theta}_k(i,j,l) = 1/\beta_1 \sum_{m,n} w_{ij}^{mn} G_k(m,n) E(i,j,l) \tag{4}$$

Where w_{ij}^{mn} is a Gaussian window and α_{ij}^{mn} is a Gaussian window modulated by the local intensity network.

4 Energy Function

The dynamics can be derived from the following energy function:

$$\begin{aligned} J(\theta, G; E) \quad = \quad & \sum_{kijlmn} w_{ij}^{mn} G_k(m,n)\theta_k(i,j,l)E(m,n,l) \quad (5) \\ & -\eta \sum_{ijmnk} \alpha_{ij}^{mn} G_k(i,j) G_k(m,n) \\ & +\beta_1 \sum_{ijk} G_k(i,j) \log G_k(i,j) \\ & +\beta_2 \sum_{ijlk} \theta_k(i,j,l) \log \theta_k(i,j,l) \end{aligned}$$

To understand the justification for this energy function, consider the expression:

$$J_{ij}(\theta_k) = \sum_l \theta_k(i,j,l)E(i,j,l) \tag{6}$$

Recall that $E(i,j,l)$ measures the motion error at location i,j and hence the higher the motion error for a velocity the higher the penalty for a unit with that preferred velocity to be active. Due to the ill-posedness of the motion estimation problem, $J_{ij}(\theta_k)$ will have multiple minima. Therefore a smoothness constraint may be imposed via a larger integration window (as in [Lucas and Kanade, 1981]) :

$$J_{ij}(\theta_k) = \sum_l \sum_{m,n} w_{ij}^{mn} \theta_k(i,j,l)E(m,n,l) \tag{7}$$

But a large integration window is likely to contain multiple motions. Hence we gate the errors to the kth expert by G_k:

$$J_{ij}(\theta_k) = \sum_l \sum_{m,n} w_{ij}^{mn} G_k(m,n)\theta_k(i,j,l)E(m,n,l) \tag{8}$$

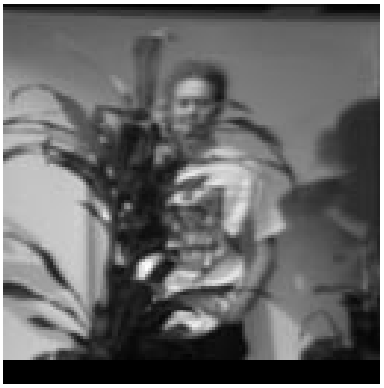

Figure 2: A frame from a sequence presented to the network. The sequence shows a person moving behind a plant.

This gives the first term of the energy function. The second term reflects the fact that nearby points having similar intensities should be assigned to the same expert. Finally the last two terms (the entropies) penalize for distributions where only one unit is active: omitting these terms causes the softmin function in equations 3 and 1 to be replaced by a "hard" winner take all function.

It is easy to show that constrained minimization of the energy function with respect to $\theta_k(i,j,l)$ gives the M dynamics. Minimizing the function with respect to $G_k(ij)$ gives a slightly different version of the E dynamics with the term $E(i,j,l)\theta_k(i,j,l)$ replaced by the weighted average $E(i,j,l)\sum w_{ij}^{mn}\theta_k(m,n,l)$. Note, however that the velocity fields of each expert are by construction smooth. Hence, this sum is well approximated by the term $E(i,j,l)\theta_k(i,j,l)$ and the E dynamics can be viewed as an approximate minimization. In practice we have found that the approximate solution works as well as the exact one.

5 Simulation Results

The performance of the network on a real image pair is illustrated in figures 2 and 3. An important parameter in our network is the number of velocity tuned units assumed to exist at every location, i.e. the sampling used in discretizing velocity space. In the simulations reported here, we assumed that the sampling is sufficiently dense such that the distribution of unit activity approximates a continuous function. As a measure of motion error we used the gradient constraint (cf. [Horn, 1986]):

$$E(i,j,l) = (dx^t V_l + dt)^2 \tag{9}$$

Where dx, dt denote the temporal and spatial derivatives at location i,j respectively. Note that this expression is quadratic in V_l. Thus the term $\tilde{\theta}_k(i,j,l)$ in equation 4 is also quadratic in V_l and equation 3 can be evaluated analytically.

Figure 2 shows one frame from a sequence showing a person moving behind a plant. Figure 3 shows the activity in the network as a function of time. On the left is shown the activity in a sheet of the gating subnetwork. The grey level represents the probability that a pixel be assigned to one of the experts: white regions are confidently assigned, black regions are confidently rejected and grey regions can be equally assigned to both experts (these are regions where there is no motion information). As can be seen, the network converges rapidly to a correct motion estimate and segmentation.

6 Discussion

The energy function in equation 5 is, of course, not the only possible one to use as a cost function for motion estimation and integration. In related work [Weiss and Adelson, 1994] we have experimented with other cost functions. The common feature of the various functions we have explored is that they contain the following three terms:

- a term measuring the local prediction error, i.e. how well does the expert to whom this pixel is assigned predict the local motion measurements.
- a term rewarding coherence of the motion fields of each expert.
- a term rewarding coherence of the assignments. i.e, rewarding assignments in which neighboring pixels of similar intensities are assigned to the same expert.

It is the third term that differentiates our work from many computer vision algorithms for motion segmentation. We believe that the integration of form and motion cues for segmentation is crucial. In our current work we are studying ways to improve this integration by having perceptual organization cues, rather than simple local intensity modulate the local interconnections in the gating network.

A neural net model which also includes gating of motion energy units has been recently suggested by [Nowlan and Sejnowski, 1993]. However, their model, unlike the one presented here, does not compute segmentation or grouping. In their algorithm, the gating units are trained off-line and essentially learn to suppress measurements centered on motion boundaries.

The network we have been using is closely related to the EM algorithm for mixture estimation studied by [Jordan and Jacobs, 1994]. The main difference is that in their mixture of experts network the experts and the gating networks are assumed to be generalized linear models. This serves to keep the number of parameters estimated significantly smaller than the number of measurements. Here we keep the large number of parameters to estimate (which enables us to segment arbitrarily shaped regions) and add additional smoothness constraints on both the gating parameters and the motion parameters. A second difference between our work and that of Jordan and Jacobs is our emphasis on parallel implementation. Unlike the general mixture estimation problem, motion segmentation has the feature that all measurements are typically acquired simultaneously. One is tempted therefore to look for algorithms that can be implemented in hardware by retinotopic units performing simple operations in parallel. As our simulation results show, units of this type can collectively produce rapid and accurate motion estimation and segmentation.

References

[Black and Anandan, 1993] Black, M. J. and Anandan, P. (1993). The robust estimation of multiple motions: affine and piecewise smooth fields. Technical Report spl-93-092, Xerox PARC.

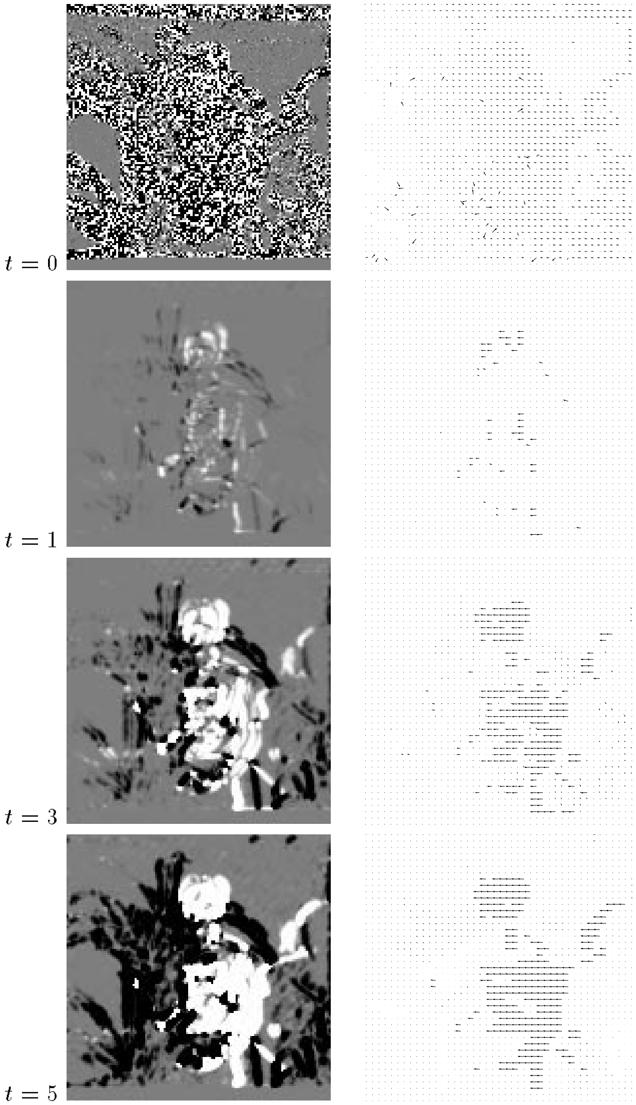

Figure 3: The activity of sheet one of the gating network $G_1(i, j)$ (left) and the estimated flow (right) as a function of time, starting with random initial conditions

[Bulthoff et al., 1989] Bulthoff, H., Little, J., and Poggio, T. (1989). A parallel algorithm for real-time computation of optical flow. *Nature*, 337(6207):549–553.

[Darrell and Pentland, 1991] Darrell, T. and Pentland, A. (1991). Robust estimation of a multi-layered motion representation. In *Proc. IEEE Workshop on Visual Motion*, pages 173–178, Princeton, New Jersey.

[Horn, 1986] Horn, B. K. P. (1986). *Robot Vision*. The MIT Press, Cambridge, MA.

[Hutchinson et al., 1988] Hutchinson, J., Koch, C., Luo, J., and Mead, C. (1988). Computing motion using analog and binary resistive networks. *IEEE Computer magazine*, 21:52–64.

[Jordan and Jacobs, 1994] Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214.

[Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130.

[Nowlan and Sejnowski, 1993] Nowlan, S. and Sejnowski, T. (1993). Filter selection model for generating visual motion signals. In Hanson, S., Cowan, J., and Giles, C., editors, *Advances in Neural Information Processing Systems 5*, pages 369–376.

[Poggio et al., 1985] Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317:314–319.

[Simoncelli et al., 1991] Simoncelli, E., Adelson, E., and Heeger, D. (1991). Probability distributions of optical flow. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 310–315.

[Terzopoulos, 1986] Terzopoulos, D. (1986). Regularization of inverse visual problems involving discontinuities. *IEEE Trans. PAMI*, 8:413–424.

[Wang and Adelson, 1994] Wang, J. Y. A. and Adelson, E. H. (1994). Representing moving images with layers. *IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638.

[Weiss and Adelson, 1994] Weiss, Y. and Adelson, E. H. (1994). Perceptually organized EM: a framework for motion segmentation that combines information about form and motion. Technical Report 315, MIT Media Lab, Perceptual Computing Section.