# Multiple-Class Spatiotemporal Flow Estimation Using a Modified Neural Gas Algorithm

Manish P. Shiralkar and Robert J. Schalkoff*
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634

October 23, 2008

**Abstract**

The problem of estimating optical flow fields corresponding to multiple moving objects in a spatiotemporal image sequence is addressed. A modified version of the Neural Gas (NG) unsupervised learning algorithm is used to implement a non-linear interpolation strategy to overcome the aperture problem encountered during local motion estimation. Local motion constraints are formulated and the best information over 4 point pairs is used to produce a single motion estimate. Wherever the aperture problem is encountered, the minimum norm estimate is produced. These local estimates are then refined using modified NG. NG provides a framework for the fusion of local and incomplete motion information into complete and global estimates. Due to the self-organizing nature of NG, the number of motion classes need not be specified apriori. The technique leads to generation of an optical flow field without the 'smearing' of flow fields encountered in regularization-based techniques. Motion estimation results obtained on synthetic and natural image sequences are shown.

Keywords:
*motion estimation, optical flow, unsupervised learning, neural gas, aperture problem*

*Please address all correspondence to R.J. Schalkoff; **email:** ***rjschal@clemson.edu***

# 1 Introduction

Optical flow computation is an important component of early vision. Optical flow has applications in fields such as surveillance, object-based video compression and recovery of 3D shape of objects.

We describe a new optical flow field estimation technique that is applicable to image sequences containing multiple and possibly occluding objects. It is well-known that the determination of motion parameters for a single motion class using an optical flow formulation poses several well-known challenges, including an inherently locally ill-posed estimation problem with the possibility of the aperture problem. An apriori unknown number of motion classes further compounds the problem complexity.

Numerous computational models for estimating optical flow exist. They can be classified into following main groups [BB95]: (1) intensity-based differential methods; (2) frequency-based filtering methods; and (3) correlation-based methods. Optical flow computation based on differential methods can further be classified into global methods such as the Horn-Schunck [HS81] approach and into local methods such as the Lucas-Kanade [LK81] technique. Black and Anandan [BA96] have developed a framework for robust estimation which can be applied to both local and global methods. Bruhn et. al. [BWS05] introduced combined local-global (CLG) method which holds promise of real-time performance.

In this paper, a modified unsupervised learning approach (Neural Gas) is used to identify motion parameters for each of the motion classes. Our overall approach consists of two parts:

1. Local estimation of image motion parameters (flow field) using the spatiotemporal constraints (Section 2); and

2. A global pass for determination of complete motion estimates using an unsupervised learning technique, specifically, a modified form of Neural Gas (NG) (Section 4.2).

# 2 Optical Flow Constraints and Algebraic Properties

The 3D motion of multiple objects and (possibly) the image sensor induce 2D motion on the image plane. This 2D motion, also called apparent motion

or optical flow, needs to be recovered from the image sequence. Suppose an image point $\underline{x} = (x, y)$ at time $t$ is moved to $(x + d_x, y + d_y)$ at time $(t + d_t)$. Under a constant intensity assumption, the images of the same object point at different times have the same luminance value. Therefore

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t).$$

A Taylor series expansion yields

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t) + \left(\frac{\partial \psi}{\partial x}\right) d_x + \left(\frac{\partial \psi}{\partial y}\right) d_y + \left(\frac{\partial \psi}{\partial t}\right) d_t.$$

or

$$\left(\frac{\partial \psi}{\partial x}\right) d_x + \left(\frac{\partial \psi}{\partial y}\right) d_y + \left(\frac{\partial \psi}{\partial t}\right) d_t = 0.$$

This equation can be written in terms of the flow vectors by dividing both sides by $dt$, yielding:

$$\nabla \psi(\underline{x}, t)^T \underline{v} = -\frac{\partial \psi(\underline{x}, t)}{\partial t} \tag{1}$$

where $\underline{v} = [v_x \ v_y]^T$ represents the velocity vector (also called the flow vector) at $\underline{x}$ and $\nabla \psi(\underline{x}, t)^T = [\frac{\partial \psi}{\partial x} \ \frac{\partial \psi}{\partial y}]$ is the spatial gradient vector of $\psi(x, y, t)$. Equation 1 is the motion constraint equation or the *optical flow constraint equation*.

## 2.1 Adding Local Constraints

Optical flow, using spatiotemporal information at a single pixel location, is under-constrained. Typically, other nearby points are used, with the assumption of locally constant [LK81] or slowly varying velocities [HS81]. This implies *spatial coherence of flow vectors*. With another spatially close point, the optical flow constraint may be augmented as:

$$\begin{bmatrix} \frac{\partial \psi}{\partial x_1} & \frac{\partial \psi}{\partial y_1} \\ \frac{\partial \psi}{\partial x_2} & \frac{\partial \psi}{\partial y_2} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} \frac{\partial \psi}{\partial t_1} \\ \frac{\partial \psi}{\partial t_2} \end{bmatrix} \qquad or \qquad D\underline{v} = -\underline{f}_t. \tag{2}$$

We refer to this augmentation as the spatiotemporal constraint. The spatial coherence assumption is not valid when these neighboring points $\underline{x}_1$ and $\underline{x}_2$ correspond to two different motion classes. Moreover, even with a valid spatial coherence assumption, we may still face the aperture problem.

## 2.2 The Aperture Problem: Two Viewpoints

**Geometric Visualization.** Consider the region on the edge (Aperture 2) shown in Figure 1. As shown in Figure 2, given $\nabla(\psi)$ and $\frac{\partial \psi}{\partial t}$, the projection of the velocity vector along the normal direction is fixed, with $v_n = -(\partial\psi/\partial t)/||\nabla\psi||$, whereas the projection onto the tangent direction, $v_t$, is undetermined [HK87]. Any value of $v_t$ would satisfy the equation. That means any point on the tangent line will satisfy the equation. This ambiguity in estimating the motion vector is known as the *aperture problem.*

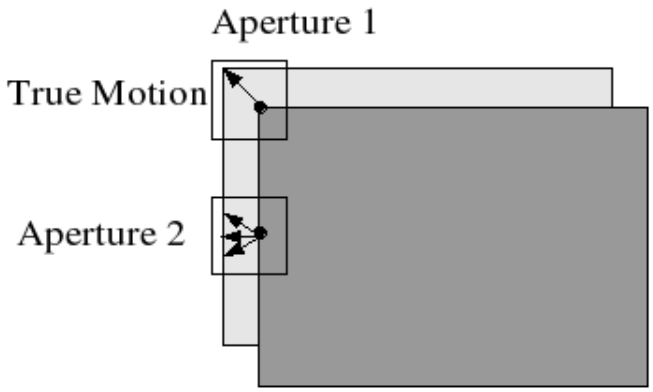



Figure 1: Aperture Problem.

**Algebraic Viewpoint of the Aperture Problem.** An alternative viewpoint of the aperture problem can be obtained considering the properties of matrix $D$ in Equation 2. There are 3 cases:

1. rank$(D) = 0$. This case corresponds to a location with no spatial texture or where motion is not discernible.

2. rank$(D) = 1$. This image location suffers from the aperture problem. Let $\underline{v} = \underline{v}_{partEst} + \underline{v}_a$, where $\underline{v}_{partEst}$ is the observable component of motion and where $\underline{v}_a$ is the component that cannot be determined due to the aperture problem. $\underline{v}_a$ satisfies $D\underline{v}_a = \underline{0}$ or $\underline{v}_a \in nullspace(D)$. The Moore-Penrose inverse of $D$ is used to find minimum norm solution for this rank deficient case.

3. rank$(D) = 2$. This case leads to a complete and acceptable motion estimate if the condition number of $D$ is reasonable.

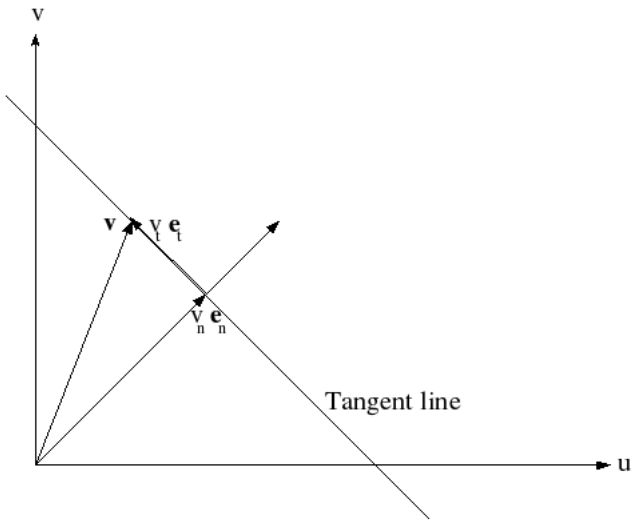



Figure 2: Decomposition of flow vector.

We use the algebraic viewpoint in our work. It provides a formal, systematic technique for the recovery of local motion component $\underline{v}_a$ using a modified version of the Neural Gas (NG) algorithm.

# 3 Motion Recovery using Neural Gas

We propose a method to identify partial or complete local motion information and then to recover missing motion information. The general idea is as follows:

1. Use the spatiotemporal constraint (Equation 2) to produce local motion estimates. Pixel locations that suffer from the aperture problem yield a partial, or incomplete, flow vector estimate.

2. Refine the motion estimates using a modified form of NG. Pixel locations where one of the motion components is missing due to the aperture problem get that component restored as a part of the nonlinear (NL) interpolation implemented by the modified NG procedure.

A significant issue in our application is that the number of motion classes is not known apriori. NG is able to accommodate this constraint.

**Unsupervised Learning Approaches Considered.** Unsupervised learning techniques are used in a number of problem domains [XW05]. Some measure of pattern associativity or similarity is used to guide the learning process, which usually leads to some form of correlation, clustering, or competitive behavior [Sch97]. Using competitive learning, these algorithms identify the clusters in data. Techniques that exhibit such cluster discovery capability in absence of prior cluster information are termed self-organizing. This soft-computing paradigm is especially useful when we are dealing with imprecision, uncertainty and partial truth. This is certainly the case for motion estimation.

Unsupervised learning approaches have been applied to motion estimation. Detailed discussion of merits and demerits of unsupervised algorithms for their use in motion estimation can be found in [Shi04].

Duc, et al. [DSB95] used fuzzy C-Means for motion estimation and segmentation. The technique requires an upper threshold for the expected number of motion classes. C-Means gives good results when the number of motion classes are known before hand. This is usually not the case and the choice of $c$ is a challenging issue.

SOFM employs a soft-max adaptation [Koh90], and can be applied to motion estimation without knowing the number of motion classes apriori. Kothari and Bellando [KB97] used SOFM to estimate optical flow. The technique works on edges, but does not lead to dense optical flow.

Neural Gas (NG) [MBS93, AS98] also uses a soft-max adaptation rule. Neural Gas vector adaptation is determined by the relative distances within the neural weight space and not determined by relative distances between neural units within any topologically pre-structured lattice. In addition, the number of motion classes need not be specified apriori. It provides a relatively simple and highly effective algorithm for clustering [CHHV06, XW05]. NG implements stochastic gradient descent on a cost function and is less subject to getting trapped in local minima (Section 4.3). We use NG to achieve nonlinear interpolation of flow estimates. Unlike regularization-based approaches [HS81], our modified-NG based approach will not put nonzero flow vectors in image regions whose spatiotemporal constraints do not support them and where visually no flow is discernible.

Growing Neural Gas (GNG) [Fri95] also uses a soft-max adaptation rule. It differs from Neural Gas with respect to the evolution of parameters over time. GNG grows as it learns, adding units and connections until some criteria is met. In our experience with GNG, motion cluster formation is

inferior to Neural Gas.

# 4 The Overall Motion Estimation Algorithm

Our technique uses NG-based refinement of preliminary, and perhaps incomplete, motion estimates to get the final (refined) motion flow field. Another unique feature is the opportunistic initialization of NG weights and the associated training of those weights.

## 4.1 The Motion Estimation Sequence (2 Frames)

The steps of the motion estimation technique are exemplified by processing the two images of a synthetic image sequence as shown in Figure 3. For illustration, the images have eight square objects moving in eight different directions, therefore we have eight motion classes to be detected. The true motion $(v_x, v_y)$ for each object is shown in Figure 4. The origin of the coordinate system is at the top left corner of the image. The temporal difference image is given in Figure 4.

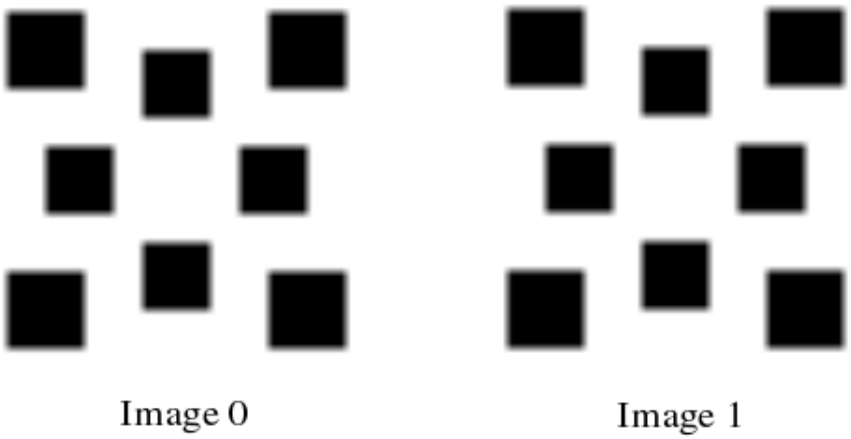



Figure 3: Binary Image Sequence.

**Step 1: Pre-processing of Image Sequence.** For binary input images, spatial smoothing using a Gaussian kernel is done. Smoothing of binary

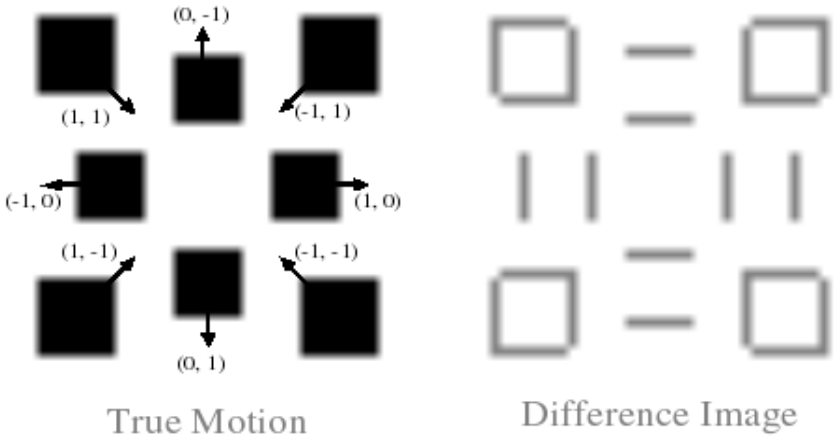



Figure 4: True Motion and Temporal Difference Image.

images is required as the gradient estimation step assumes that the input images are differentiable. Gray level images are assumed to be differentiable, so no smoothing is applied.

**Step 2: Gradient Estimation.** The second step in motion estimation is the computation of gradients. We must estimate these partial derivatives from the discrete set of image intensity measurements available. It is important that the estimates of $\psi_x$, $\psi_y$ and $\psi_t$ be consistent. That is, they should refer to the same point in the image at the same time. We use the approximate differentiation as given in [HS81]. Figure 5 shows the spatial and temporal gradients computed for the image sequence.

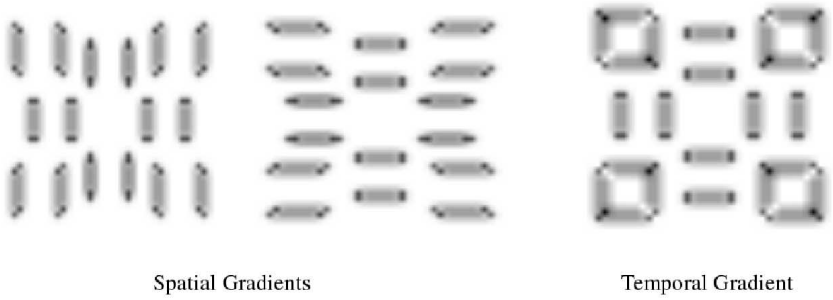



Figure 5: Gradients $\psi_x$, $\psi_y$ and $\psi_t$.

**Step 3: Local Motion Estimation.** At each pixel location we generate four independent motion constraints, one for each of the four orientations, as seen in Figure 6. Only the best spatiotemporal constraint is retained as defined by Equation 4. This scheme provides fine-grained control over choice of training vectors for use in Neural Gas.

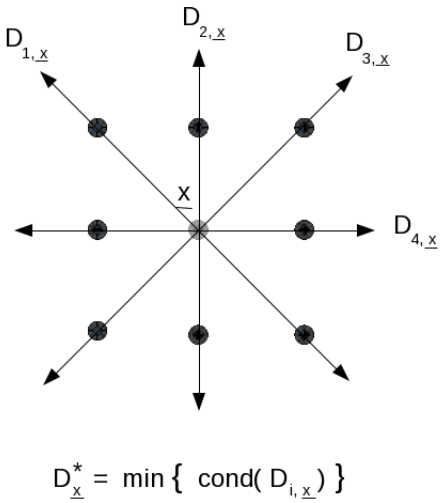



Figure 6: Neighbors used in production of motion estimates

More succinctly, at pixel location $\underline{x}$, using neighbor set (direction) $i$, the constraint set becomes:

$$D_{i,\underline{x}}\underline{v} = -\underline{f}_t \quad i = 1, 2, 3, 4 \tag{3}$$

where $D$ is defined in Equation 2.

The best local motion estimate is then obtained using:

$$D^*_{\underline{x}} = \min_i \{cond(D_{i,\underline{x}})\} \tag{4}$$

where $cond()$ is the matrix condition number, and

$$\underline{v}^*_{\underline{x}} = -(D^*)^{\dagger}_{\underline{x}}\underline{f}_t \tag{5}$$

where † denotes the Moore-Penrose inverse.

The condition number is used to determine which one of the four speculative pixel pairs is the best choice for production of the local motion estimate. As indicated by Equation 4, the motion estimate corresponding to the orientation with the smallest condition number is generated. If $\text{rank}(D^*_{\underline{x}}) = 1$ then $\underline{v}^*_{\underline{x}}$ from Equation 5 is the minimum norm solution. This corresponds to the aperture problem case and $\underline{v}^*_{\underline{x}}$ is the part of motion that could be estimated. This flow estimate is put into set $H_p$ ($p$ denotes 'partial'). If $\text{rank}(D^*_{\underline{x}}) = 2$, a unique flow vector estimate results. If the condition number is very small then the flow estimate $\underline{v}^*_{\underline{x}}$ is put into set $H_c$ ($c$ denotes 'complete'). Thus, set $H_p$ holds all the partial motion estimates and set $H_c$ holds all the complete motion estimates.

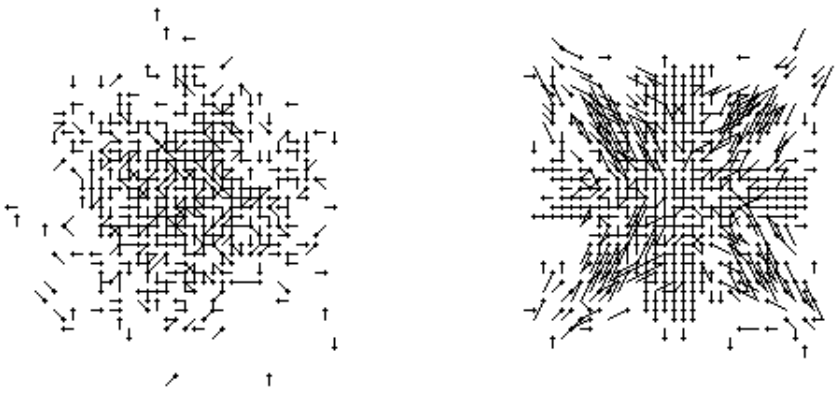
The weight or feature vectors in sets $H_c$ and $H_p$ are of the form:

$$\underline{v} = (x, y, v_x, v_y)^T \tag{6}$$

where $v_x$and $v_y$ are motion estimates at pixel location ($x$, $y$).

**Step 4: Initialization of Neural Gas.** If we initialize the weights of the NG array using a Gaussian distribution, typical flow field results for the example case are shown in Figure 7. However, a more opportunistic scheme for weight initialization exists. We use the set $W$, which is the union of set $H_c$ (vectors with 'complete' motion estimates) and set $H_p$ (vectors with 'partial' motion estimates), to initialize the weights. This yields the initial set of weights for NG processing as shown in Figure 8(a).

**Step 5: Motion Clustering and NL Interpolation of Flow Using Modified NG.** This is the final step in our motion estimation technique. The objective is to put the complete and partial motion estimates that belong to a single motion class into a single cluster. We train the initialized weights $W$ using the set $H_c$. The modified NG correction of Section 4.2, Equation 7, is used. During the training, the training input vector is compared with the weights of NG. Since each training vector exists in the NG weights set (recall $W = H_c \cup H_p$), the training vector finds an exact match in one of the unit's weights. As the local soft-max update takes place, the neighbors selected for update are the ones closer to the winning unit in the weight vector space. For the example case, the training vectors in set $H_c$ correspond



(a) Initial Weights (b) Weights after 10 iterations

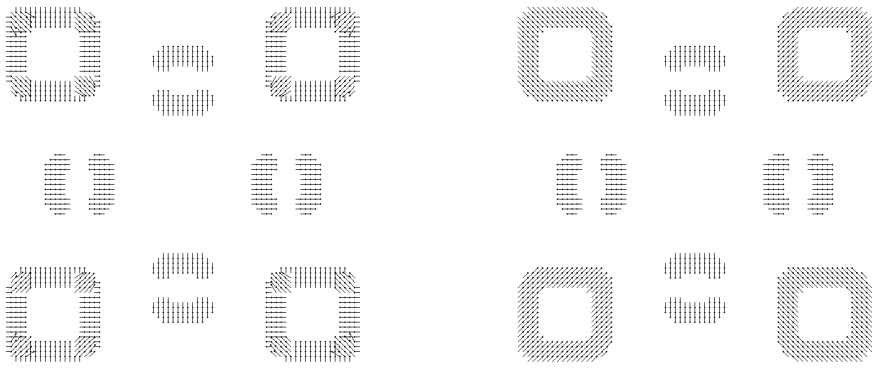
Figure 7: Evolution of Weights During Neural Gas Clustering (Gaussian Initialization).

to the motion vectors for the corners of the squares. $H_p$ consists of motion vectors for the edges of the squares where only one motion component could be estimated. As we have initialized the NG weights using $W = H_c \cup H_p$, vectors associated with both the complete and partial motion estimates are available as NG weights. For example, when a corner (training) vector [1] is used in NG weight adaptation, the weights (neighborhood) that get corrected most are the vectors that are spatially close to the corner. The edges of the squares fall in this neighborhood and their motion gets corrected. The result can be seen in Figure 8(b).

The overall flow of the processing of image sequences to obtain the motion estimates is shown in Figure 10.

**Computational Time, Complexity and Related Concerns.** The machine used for this computation was an Intel(R) Core(TM)2 Duo CPU, T7300 @ 2.00GHz. The overall time required for processing the synthetic image sequence of Figures 8(a) and 8(b) was 0.85s with the cardinality of $H_c$ equal to 256 elements and the cardinality of $W$ equal to 1328 elements. Note that

[1] Assuming $rank(D) = 2$.



(a) Initial Weights (b) Weights after 10 iterations

Figure 8: Evolution of Weights During Modified Neural Gas Clustering: Weights Initialized Using $W$.

processing time is *not* directly a function of image resolution, but rather depends upon the cardinality of sets $H_c$ and $W$. This is due to the fact that the processing time is dominated by the Neural Gas computation. The NG computation time, in turn, depends upon the cardinality of vector sets $H_c$ and $W$. This effect is further shown in the examples of Section 5.

## 4.2 Modifications to NG for Motion Estimation

The unmodified Neural Gas algorithm update [AS98] with training vector $\underline{i}$ is defined by:

$$\Delta \underline{w}_i = \epsilon(t) \,.\, h_{\lambda'}(m_i(\underline{i}, W)) \,.\, (\underline{i} - \underline{w}_i) \tag{7}$$

where $W = \{\underline{w}_1, \underline{w}_2, \ldots, \underline{w}_N\}$ is the whole set of neural weights, $m_i(\underline{i}, W)$ is the implicit ordering of $W$ defined as $m_i = \frac{d_i - d_{min}}{d_{max} - d_{min}}$. $d_{min}, d_{max}$ being the minimum and maximum distance between $\underline{i}$ and all $w_j$s, $h_{\lambda'}(m_i) = exp(-m_i\lambda'(t))$, and $\lambda'(t) = \lambda(t)/(N-1)$.

**Vector Similarity Measure for Implicit Sorting.** The scale of first two vector components (spatial coordinates) is orders of magnitude larger than that of the motion components (Refer to Equation 6). We use a weighted

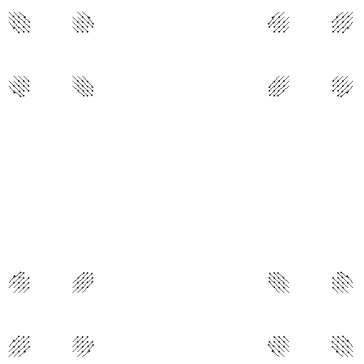


Figure 9: Training Weights.

distance metric given by

$$d(\underline{x}, \underline{y}) = ||\underline{x} - \underline{y}||_R = \sqrt{(\underline{x} - \underline{y})^T \; R \; (\underline{x} - \underline{y})}$$

where $R$ is the diagonal matrix consisting of estimated inverse variance values of the vector components.

**Modified Vector Correction for Update.** We only update the components of weight vectors that carry motion information. Our modified NG corrects the sorted $\underline{w}_j$ as:

$$\underline{w}_j^{t+1} \propto M(\underline{v} - \underline{w}_j^t)$$

where

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

only corrects the last 2 elements of $\underline{v}$ (i.e., the flow). Pixel locations are persistent.

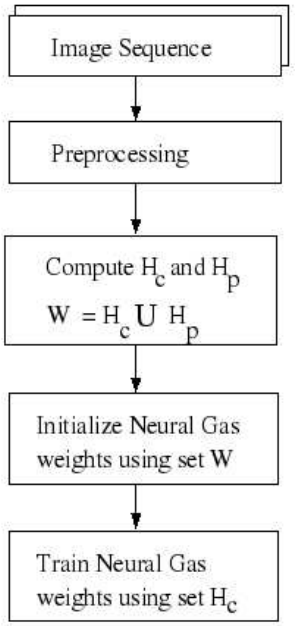



Figure 10: Summary of motion estimation steps.

## 4.3 Behavior of Modified NG Motion Estimate Adjustments

Martinetz [MBS93] has shown that given a training set of $H = [\underline{v}_i]$, NG will self-organize and yield a resulting set of vectors $W = [w_j]$ such that

$$E = \frac{1}{2} \sum_j \int_H p(\underline{v}) h_\lambda(k_j(\underline{v}, W)) ||\underline{v} - \underline{w}_j||^2 \underline{dv} \tag{8}$$

is minimum. In our strategy, NG allows recovery of the aperture component of flow, $\underline{v}_a$. Assume training vectors in $H$ correspond to flow estimates from cases where $D$ has full (2) rank. In these estimates, the aperture problem cannot exist. Thus, the weight correction given by Equation 8 locally restores the component of motion that was missing due to the aperture problem.

**Remarks on Cluster Formation.** One of the factors that heavily influences the cluster formation is the neighborhood size. In the Neural Gas algorithm, the neighborhood size is determined by the parameter $\lambda$ in Equa-

tion 7 [MBS93]. Over iterations, the value of $\lambda$ decreases, thus reducing (or 'shrinking') the neighborhood size. If we start with a large value of $\lambda$ (a large neighborhood), a motion estimate at one location may affect motion estimate at far away locations significantly, corrupting the estimate. In addition, to avoid corrupting motion estimates we use learning rates that do not drastically affect the weights at any given step. Thus, there is a trade off between speed of convergence and the correctness of motion estimates.

**Biological Plausibility of NG-based Motion Self-Organization and Interpolation.** The primary visual cortex (V1) is a well-defined processing region for accumulation of low-level spatial information in vision, and modeling the behavior of cortical networks with self-organizing maps is an active area of research [BKM02],[SM94]. The low-level spatiotemporal motion estimation technique presented herein exhibits structural and functional similarity with low-level biological visual processes.

Initially, individual V1 neurons have strong tuning to a small set of stimuli including changes in visual orientations, spatial frequencies and colors. Later in time (after 100 ms), V1 neurons become sensitive to the more global organization of the input image, and interact (through recurrent processing) with other visual cortex areas.

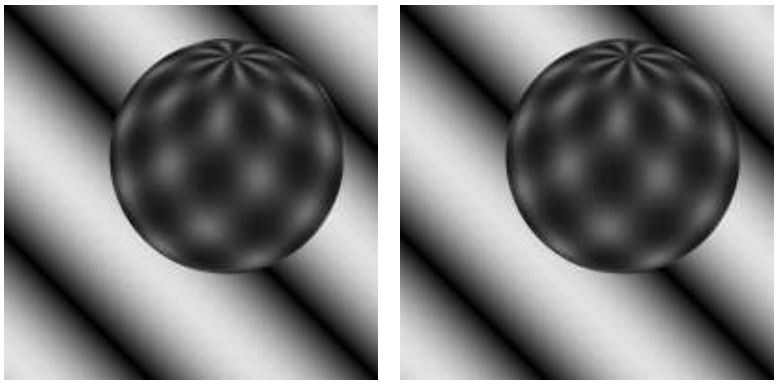
Our computational strategy, in the motion domain, is similar. Preliminary, and perhaps incomplete local motion stimuli are refined using NL interpolation and global motion information. The clustering of motion classes we achieve using NG is analogous to perceptual chunking [GLC$^+$01], where primitive stimuli (in our case estimated motion) is grouped into larger conceptual groups (motion classes).

# 5 Additional Motion Estimation Examples

**Rotating Sphere Sequence.** The rotating sphere sequence was obtained from database used in [BKDB01]. The image frames of the sequence are seen in Figure 11. The flow due to training vectors in set $H_c$ is seen in Figure 13. In the Figure 14 which shows the resultant motion estimate, it can be seen that NG could overcome the aperture problem which is apparent in Figure 12 as the horizontal and vertical flow vectors.

This example also illustrates the NL nature of the NG-generated flow vector interpolation. In untextured image regions, where the spatiotempo-

ral constraint (Equation 2) involves a rank-0 $D$ matrix, motion would not be discernible. Since our algorithm does not include any flow vector estimates from these regions in the NG weights set, erroneous estimates are not introduced into these regions. This is in contrast to regularization-based strategies incorporating a smoothness constraint on the resulting flow field. The computation time was 2min 14s. In this case, the cardinality of $H_c$ was 4658 elements and the cardinality of $W$ was 11580 elements.



(a) First image frame. (b) Second image frame.

Figure 11: Two frames from Rotating Sphere Sequence.

**Taxi Sequence.** The image frames of a taxi sequence are seen in Figure 15. The dominant motion in this sequence is the motion of the foreground car. The set $H$ as seen in Figure 16 has motion vectors corresponding to the background stationary car as well. The training set $H_c$ is seen in Figure 17. The training vectors for the background car are few compared to training vectors for the moving car. If $\lambda$ used for NG is very large, then the moving car motion vectors affect the stationary car vectors as well. So in this case the $\lambda$ value has to be kept small. If the number of training vectors for two spatially adjacent motion classes vary by large amount, then the motion class with larger number of training vectors dominates the motion correction. The motion estimates for the taxi sequence are shown in Figure 18. The computation time was 4.9s with the cardinality of $H_c$ equal to 957 elements and the cardinality of $W$ equal to 2221 elements.

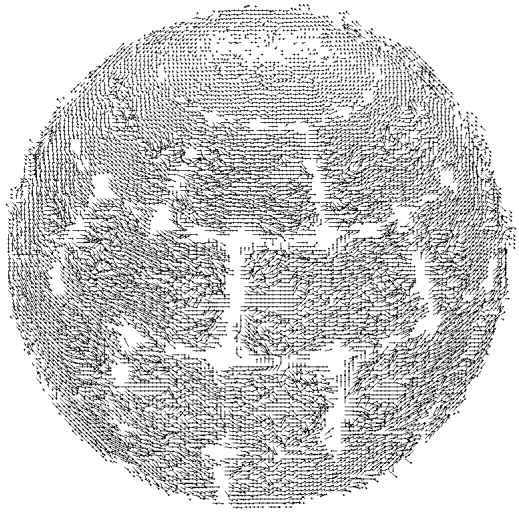


Figure 12: Rotating sphere sequence flow vectors from $W$, initial weights.

**Yosemite Sequence.** The image frames of the Yosemite sequence are seen in Figure 19. Flow vectors due to weight initialization set $H$ are seen in Figure 20. The flow vectors in training set $H_c$ as seen in Figure 21 show very few vectors for the left mountain in the sequence. Consequently, during NG adaptations, motion vectors in $H$ corresponding to mountain region do not get corrected. The resultant motion estimate for the fly-through Yosemite sequence is shown in Figure 22. The computation time was 5min 20s with the cardinality of $H_c$ equal to 6304 elements and the cardinality of $W$ equal to 20033 elements.

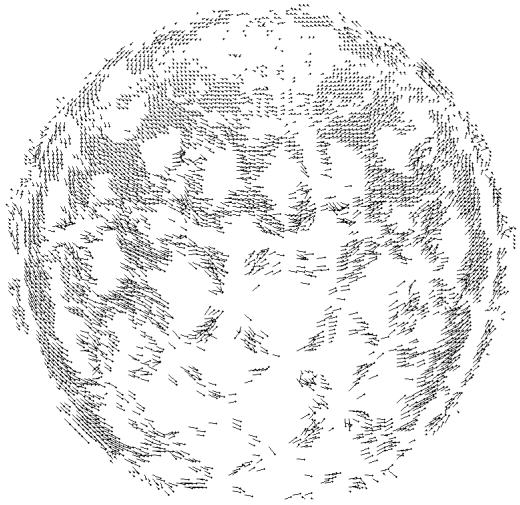


Figure 13: Rotating sphere sequence flow vectors using set $H_c$ as training weights.

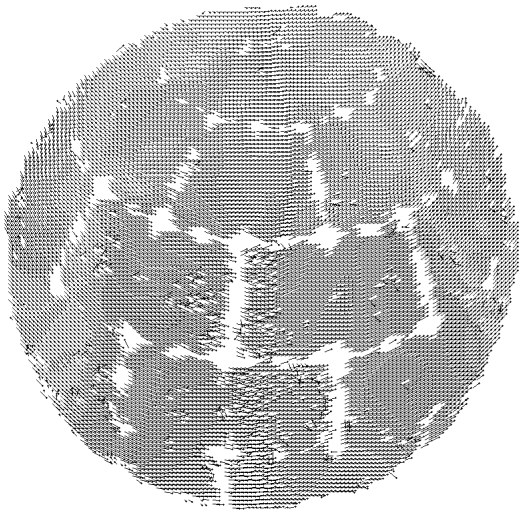


Figure 14: Rotating Sphere sequence flow after clustering.

(a) First image frame. (b) Second image frame.

Figure 15: Two frames from Taxi Sequence.

Figure 16: Taxi sequence flow vectors using set $W$ as initial weights.

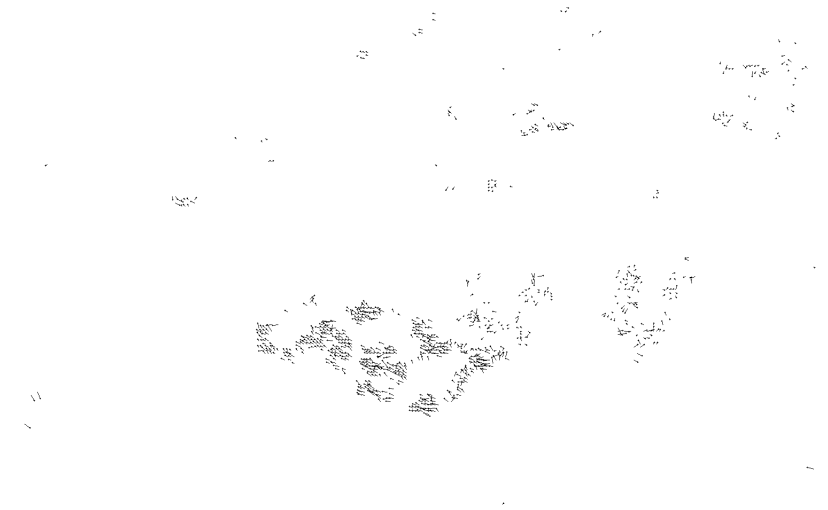
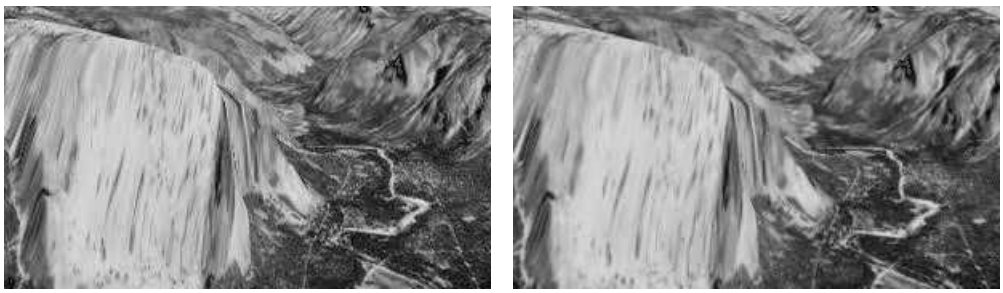


Figure 17: Taxi sequence flow vectors using set $H_c$ as training vectors.

Figure 18: Taxi sequence flow after clustering.



(a) First image frame. (b) Second image frame.

Figure 19: Two frames from Yosemite Sequence.

Figure 20: Yosemite sequence flow vectors from $W$, initial weights.

Figure 21: Yosemite sequence flow vectors using set $H_c$ as training weights.

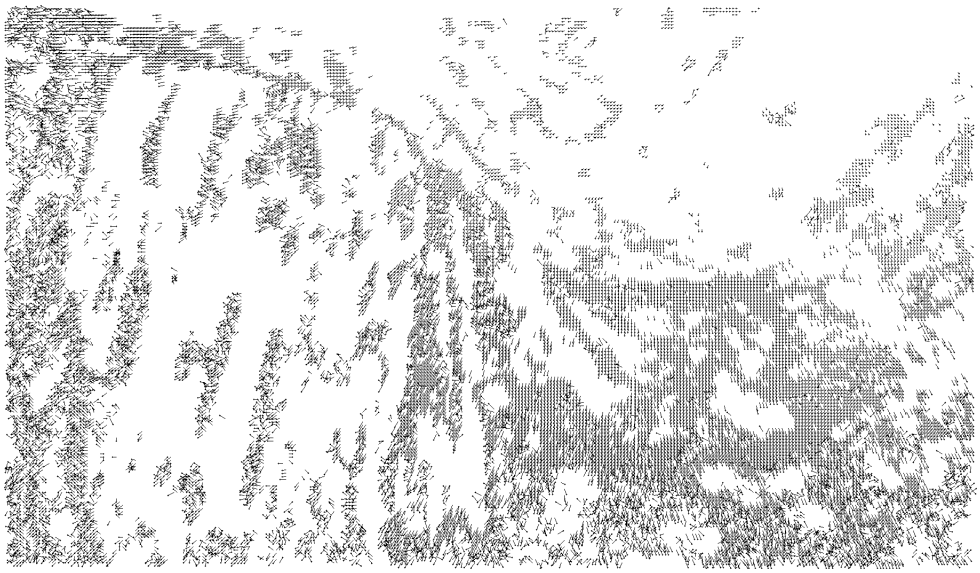


Figure 22: Yosemite sequence flow after clustering.

# 6 Comparison With Other Techniques

We use the same synthetic image sequence (8 motion classes) shown in Figure 3 for comparison. Input binary images are smoothed using a 3x3 Gaussian kernel. Our comparison is restricted to techniques that generate dense optical flow and hence we compare our results with approaches of Horn-Schunck [HS81] and Black-Anandan [BA96]. The OpenCV implementation of [HS81] was used while the implementation of [BA96] was provided by Dr. Black. Figure 23 shows the Horn-Schunck flow results after 30 iterations. Figure 24 shows the Black-Anandan flow after running the code with the default parameters. In both of these alternative approaches, the resulting motion estimates are denser than our modified-NG based approach. Conversely, our neural-gas based approach does not spread the optical flow in areas where there is no motion, as seen in the Horn-Schunck and Black-Anandan examples.

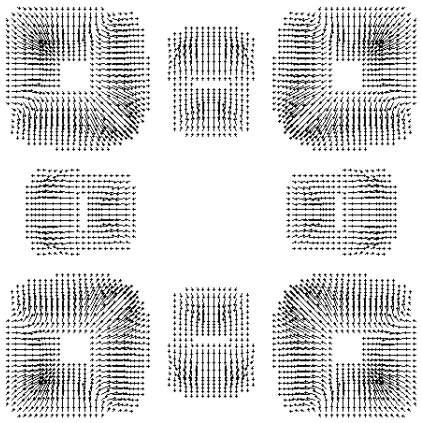


Figure 23: 8 motion case results with Horn-Schunck technique. (Compare with Figure 8)

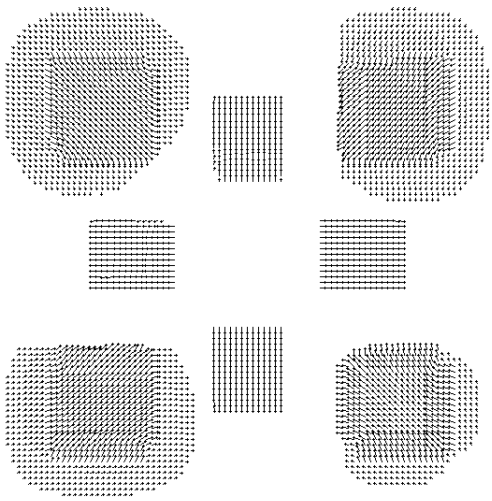


Figure 24: 8 motion case results with Black-Anandan technique. (Compare with Figure 8)

# 7 Conclusions

The combination of a modified Neural Gas unsupervised learning technique with the optical flow constraint provides a framework for the fusion of local and incomplete motion information into complete and global estimates for multiple motion classes. The training of partial motion estimates using complete motion estimates and the modified Neural Gas algorithm causes the propagation of motion estimates from the locations in the image where motion is completely known to the locations where motion is only partially known, due to the aperture problem. It can be seen that the computation time depends upon the cardinality of $H_c$ and $W$. Current efforts include autonomous determination of the NG parameters and GPU-based implementation of the algorithm.

# References

[AS98] A. S. Atukorale and P. N. Suganthan. An efficient neural gas network for classification. In *5th Int. Conf. on Automation, Robotics,*

*Control and Vision, Singapore*, 1998.

[BA96] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow-fields. *CVIU*, 63(1):75–104, January 1996.

[BB95] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *CS*, 27:433–467, 1995.

[BKDB01] McCane B., Novins K., Crannitch D., and Galvin B. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84:126–143, 2001.

[BKM02] J. Bednar, A. Kelkar, and R. Miikkulainen. Modeling large cortical networks with growing self-organizing maps. *Neurocomputing*, 44-46:315–321, 2002.

[BWS05] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *IJCV*, 61(3):211–231, February 2005.

[CHHV06] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Vilmann. Batch and median neural gas. *Neural Networks*, pages 762–771, 2006.

[DSB95] B. Duc, P. Schroeter, and J. Bigun. Motion estimation and segmentation by fuzzy algorithms. In *ICIP95*, pages 472–475, 1995.

[Fri95] Bernd Fritzke. A growing neural gas network learns topologies. In *NIPS*, pages 625–632, 1995.

[GLC+01] F. Gobet, P. C. R. Lane, S. Croker, P. C. H. Cheng, G. Jones, I. Oliver, and J. M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5:236–243, 2001.

[HK87] E.C. Hildreth and C. Koch. The analysis of visual motion: From computational theory to neuronal mechanisms. *Ann. Rev. Neurosci.*, (10):477–533, 1987.

[HS81] Berthold Horn and Brian G. Schunk. Determining optical flow. *AI*, 17:185–203, 1981.

[KB97] R. Kothari and J. Bellando. Optical flow determination using topology preserving mappings. In *ICIP97*, pages 344–347, 1997.

[Koh90] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[LK81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA81*, pages 121–130, 1981.

[MBS93] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(1):558–569, 1993.

[Sch97] Robert J. Schalkoff. *Artificial Neural Networks*. McGraw-Hill, 1997.

[Shi04] Manish Shiralkar. M.S. thesis: Design and parallel implementation of image motion parameter estimation algorithms using unsupervised learning techniques, 2004.

[SM94] J. Sirosh and R. Miikkulainen. Cooperative self-organization of afferent and lateral connections in cortical maps. *Biological Cybernetics*, 71:66–78, 1994.

[XW05] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 2005.