Median Radial Basis Functions Network for Optical Flow Processing

Adrian G. Borş Ioannis Pitas
Department of Informatics
University of Thessaloniki
Thessaloniki 540 06, Greece

ABSTRACT — **In this paper we propose a Radial Basis Functions model for the motion field. This model is represented by means of a two-layer neural network, where the hidden unit activation function is Gaussian. The weights of the network are found based on robust estimators. Marginal median and median of the absolute deviations are applied in order to find the center and the covariance matrix of the Gaussian function. We provide a fast implementation of this algorithm based on data histograms. The proposed method is applied for motion field estimation and segmentation in real images.**

1 Introduction

Radial Basis Functions (RBF) networks have been used as general functional estimators for classification or signal modeling purposes. An overview of different approaches in the RBF field was provided in [2]. In this study the basis functions are chosen to be Gaussian. Their centers, covariance matrices and the weights associated with the output connections are found by means of a robust learning algorithm.

The evaluation of the network weights in our approach is similar to the Learning Vector Quantization (LVQ) algorithm [4]. The Gaussian centers to be estimated correspond to the local estimates for the first order statistics and the covariance matrix for the second order statistics. However, the estimators based on classical statistics produce bias whenever data are not normally distributed [3]. In [6] it was proposed the Median LVQ algorithm and in [5] it was presented a theoretical analysis for the robust estimation of the RBF function parameters when estimating a mixture of Gaussians. We employ median RBF (MRBF) algorithm based on the marginal median estimator for finding the Gaussian centers and median of the absolute deviations for the covariance matrix.

A motion estimation algorithm evaluates the displacements of pixels or groups of pixels between two frames in an image sequence [1]. Motion segmentation algorithms identify the regions having similar motion vectors. Block matching algorithms largely used for motion estimation do not always provide reliable optical flows. They fail to give good estimates in the areas with almost constant intensity. After obtaining a first estimate of the optical flow by means of block matching algorithms, clustering methods can be applied in order to segment the optical flow [7].

Usually the optical flows after clustering contain some outlying vectors. In order to smooth efficiently the vector fields, the clusters are grouped by means of the RBF net and the spurious vectors are filtered out. Each moving region is associated to one motion vector. After the robust learning stage, the network can infer the optical flow and the segmentation for those frames containing block sites that are statistically consistent with the training set. We provide a fast implementation of the proposed algorithm based on the optical flow histogram modeling.

2 Median RBF Learning

RBF neural network is a nonparametric classification method which models a given input-output mapping by a weighted sum of kernels:

$$Y_k(\mathbf{X}) = \sum_{j=1}^{L} \lambda(k,j) * \phi_j(\mathbf{X}) \tag{1}$$

where Y_k is the kth output of the network, L is the number of hidden units and $\phi_j(\mathbf{X})$ is the activation of the jth kernel function when the network is presented with the $\mathbf{X}$ vector. Each kernel function is activated in a region depending of its parameters (network weights):

$$\phi_j(\mathbf{X}) = \exp\left[-(\mu_j - \mathbf{X})' \, \mathbf{\Sigma}_j^{-1} (\mu_j - \mathbf{X})\right] \tag{2}$$

where the kernels are assumed to be Gaussian with the center μ_j and the covariance matrix is $\mathbf{\Sigma}_j$.

RBFs are able to model very general functions [2, 4]. The success in applying a RBF network depends on the estimation of the individual parameters corresponding to each basis function. A two-stage learning algorithm is used for this purpose. In the first stage the input to hidden layer weights are estimated based on Learning Vector Quantization (LVQ). In the second stage the output weights are estimated.

In the LVQ algorithm we update only the center μ_j which is the closest (in Euclidean distance) to the given data sample:

$$||\mu_j - \mathbf{X}||^2 = \min_{k=1}^{L} ||\mu_k - \mathbf{X}||^2 \quad (3)$$

$$\hat{\mu}_j = \hat{\mu}_j + \frac{1}{n_j}(\hat{\mu}_j - \mathbf{X}) \quad (4)$$

where n_j is the number of patterns assigned to the hidden unit j. For the covariance matrix calculation the following formula can be used:

$$\hat{\sigma}^2_{j,hl} = \frac{n_j - 2}{n_j - 1}\hat{\sigma}^2_{j,hl} + \frac{(X(h) - \hat{\mu}_j(l))(X(l) - \hat{\mu}_j(h))}{n_j - 1} \quad (5)$$

for $h, l = 1, \ldots, N$. N is the number of dimensions for the input space. The estimators used in (4,5) are derived from the classical statistics theory. The output weights $\lambda(k, j)$ can be calculated based on the backpropagation algorithm [2].

In the training stage it is desirable to avoid using outlying patterns which may cause bias in the estimation of the RBF network parameters. The data samples which do not correspond to the data statistics (noisy patterns) should be rejected rather than used in the training stage. Marginal median was proposed to be used for LVQ center estimation [6]. The data samples are ordered and the marginal median is assigned as class center:

$$\hat{\mu}_j = \text{med}\,\{\mathbf{X}_0, \mathbf{X}_2, \ldots, \mathbf{X}_{n-1}\} \quad (6)$$

where X_{n-1} is the last data sample assigned to the cluster j according to (3). Marginal median calculation is performed along each data dimension h, independently. Because this is applied in a multidimensional space, the resulted center may not be one of the given data samples.

After the center of a certain hidden unit j is found according to (6), we use the median of the absolute deviation (MAD) for calculating the standard deviation:

$$\hat{\sigma}_j = \frac{\text{med}\,\{|\mathbf{X}_0 - \hat{\mu}_j|, \ldots, |\mathbf{X}_{n-1} - \hat{\mu}_j|\}}{0.6745} \quad (7)$$

where 0.6745 is a scaling parameter in order to make the estimator Fisher consistent for the normal distribution [3].

The cross-correlation factors of the covariance matrix can be calculated based on the MAD estimator [3]. We consider two arrays containing the difference and the sum for each two different vector components of the data samples:

$$Z^+_{hl} = X(h) + X(l) \quad (8)$$

$$Z^-_{hl} = X(h) - X(l). \quad (9)$$

We compute first the median of these new populations as in (6). The squares of the MAD estimates (7) for the arrays $\mathbf{Z}^+_{hl}$ and $\mathbf{Z}^-_{hl}$ consists of their variances and they are denoted as $\hat{V}^+_{j,hl}$ and $\hat{V}^-_{j,hl}$. The cross-correlations are derived as:

$$\hat{\sigma}^2_{j,hl} = \hat{\sigma}^2_{j,lh} = \frac{1}{4}(\hat{V}^+_{j,hl} - \hat{V}^-_{j,hl}). \quad (10)$$

In [5] the theoretical asymptotic performance was evaluated for both classical and robust statistic based algorithms for estimating the parameters of a mixture of Gaussians distribution. In the case when the Gaussian distributions are far away one from each other, the algorithms are expected to provide similar results. The robust estimators employed in the MRBF learning algorithm gave lesser expected bias than classical estimators when a certain overlap occurred among different Gaussian components of the mixture [5].

3 Fast implementation of the learning algorithm

A histogram based median implementation algorithm [8] was adapted in order to be applied for MRBF learning. The first data sample assigned to a unit becomes the starting point in finding the median. In the updating stage we take into consideration pairs of two data samples $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ assigned to the same hidden unit. We build up the marginal histogram associated with each hidden unit, denoted here as $H_{jh}[k]$ where j is the hidden unit, h is the dimension and k represents the level in the histogram. Let us denote by $\hat{\mu}^t_{jh}$ the median at instance t and let us consider $X_i(h) < X_{i+1}(h)$. Median updating can be performed by the following rule according to the rank of the incoming samples:

$$\hat{\mu}^{t+1}_{jh} = \hat{\mu}^t_{jh} + K$$
$$\text{If } \hat{\mu}^t_{jh} \in (X_i(h), X_{i+1}(h)) \text{ then } K = 0$$
$$\text{If } \hat{\mu}^t_{jh} < X_i(h) \text{ then } K > 0 \quad (11)$$
$$\text{If } \hat{\mu}^t_{jh} > X_{i+1}(h) \text{ then } K < 0$$

where K is the number of histogram levels necessary to add or to substract in order to obtain the median location. $\hat{\mu}_{jh}$ is located where the data marginal histogram splits in two parts containing equal number of samples:

$$\sum_{k=0}^{\hat{\mu}_{jh}+K} H_{jh}^{t+1}[k] = \sum_{k=\hat{\mu}_{jh}-K}^{M} H_{jh}^{t+1}[k] \qquad (12)$$

where M represents the total number of levels in the histogram. From this condition we obtain the necessary number of levels K to get the new location of the median.

We implement similarly a fast calculation for the MAD estimator (7) by using the histogram of data samples. In order to estimate the variance we can use the histograms obtained during median calculation. From these histograms we construct new histograms denoted as $\mathcal{H}_{jh}[k]$ and representing the distributions for $|X(h) - \hat{\mu}_j(h)|$:

$$\mathcal{H}_{jh}[0] = H_{jh}[\hat{\mu}_{jh}] \qquad (13)$$

$$\mathcal{H}_{jh}[k] = H_{jh}[\hat{\mu}_{jh} + k] + H_{jh}[\hat{\mu}_{jh} - k]. \qquad (14)$$

The MAD represents the median for the data contained in the $\mathcal{H}_{jh}[k]$ histogram and should fulfill a similar relationship with (12). Let us assume ψ_{jh} be the value where the histogram $\mathcal{H}_{jh}[k]$ splits in two parts containing equal number of samples. The MAD of the distribution can be derived according to this value, taking also into account the quantization error:

$$\hat{\sigma}_{jh} = \frac{1}{0.6745}\left(\psi_{jh} + \frac{\frac{n_j}{2} - \sum_{i=0}^{\psi_{jh}} \mathcal{H}_{jh}[i]}{\mathcal{H}_{jh}[\psi_{jh}+1]} - 0.5\right) \qquad (15)$$

where the second term represents the compensation for the quantization error. By means of (12,15) the RBF network approximates the histogram associated to the optical flow. The learning starts with a given number of hidden units. The hidden units which do not have associated a certain number of data samples according to (3) are discarded.

4 Simulation Results

Motion estimation and segmentation is an important task for image sequence analysis and coding. The block matching algorithms are frequently used in the existing video coding systems for estimating the optical flow. After applying the block matching algorithm we get a first estimate of the optical flow, and these motion vectors are used as inputs in the neural network.

We have used for simulations the "Hamburg taxi" sequence (Figure 1). This sequence represents three important moving objects: a taxi turning around the corner, a car in the lower left moving from left to right and a van in the lower right moving from right to left. We have estimated the optical flow, using the full search block matching algorithm, assuming blocks of size 4×4. The resulted optical flow is quite noisy as can be seen from the Figure 3.

We consider as inputs the two-dimensional optical flow provided by the block matching algorithm. The estimated speed (in pixels/frames) for the moving objects is provided in Table 1. The reference velocity was calculated from the displacements of the "clear" features of each moving object, obtained in a semiautomatic way. Both networks have 8 hidden units. The learning times corresponds to the learning of the Gaussian parameters and they were measured on a Silicon-Graphics Indigo workstation. From the Table 1 it can be seen that the MRBF learning algorithm provides better estimates for the optical flow and requires lesser computation time than the algorithm based on classical statistics (RBF).

Algorithm		Taxi	Van	Car	Learning Time (s)
RBF	x	-5.3	-5.3	5.6	0.22
	y	-1.5	-1.5	1.1	
MRBF	x	-2	-6	6	0.15
	y	-1	0	1	
Reference	x	-2	-5	6	-
	y	-1	0	1	

Table 1: The speed estimation for the moving objects from the "Hamburg taxi" sequence.

The optical flow smoothness was improved when we have enlarged the input space to five dimensions. Besides the optical flow components provided by the block matching algorithm we consider as inputs the gray level and the geometrical position for each block site. All these input entries should be scaled properly, inside of the same range. The values for the optical flow were consistent with those provided in Table 1 and the segmentation of the moving objects was accurate, as can be seen in Figure 2. The data samples identified as outliers are eliminated from the estimation procedure by means of the median RBF. The segmented optical flow is represented in Figure 4. The optical flow is well smoothed inside of the segmented moving objects.

Figure 1: Frame from "Hamburg taxi" sequence.

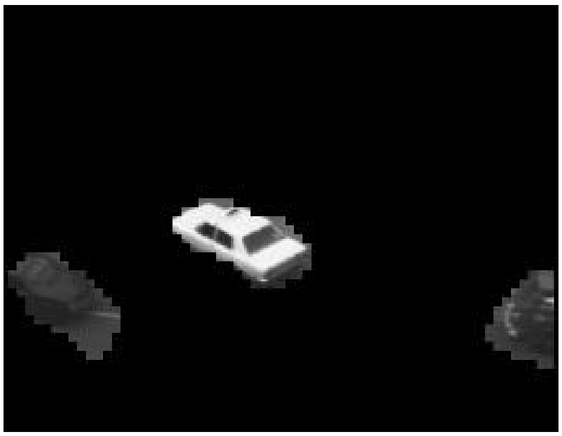

Figure 2: The moving objects segmentation.

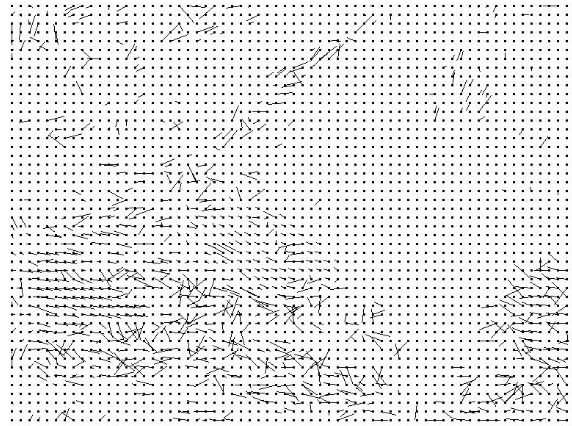

Figure 3: Block matching optical flow.

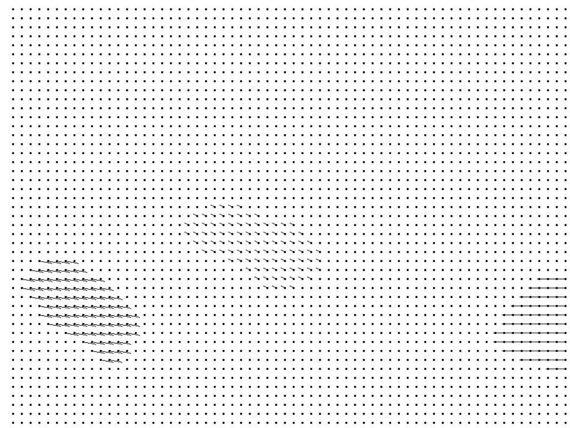

Figure 4: MRBF based optical flow.

5 Conclusions

In this study we have applied a Radial Basis Functions Network to model the optical flow. The weights of the network are found by means of a robust learning algorithm named Median RBF. This algorithm is efficient in rejecting the outliers and is proven to estimate accurately the optical flow associated with different moving objects. The moving objects are segmented according to their optical flow. The information about the image sequence contained in the network weights can be further used for analysis and coding.

References

[1] A. N. Netravali, B. G. Haskell, *Digital Pictures, Representation and compression,* (Applications of Communications Theory, series editor: R. W. Lucky), Plenum Press, 1988.

[2] A. G. Borş, M. Gabbouj, "Minimal topology for a radial basis functions neural network for pattern classification," *Digital Signal Processing,* vol. 4, no. 3, pp. 173-188, July 1994.

[3] G. Seber, *Multivariate Observations,* John Wiley, 1986.

[4] L. Xu, A. Krzyzak, E. Oja, "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," *IEEE Trans. on Neural Networks,* vol. 4, no. 4, pp. 636-649, July 1993.

[5] A. G. Borş, I. Pitas, "Robust estimation of radial basis functions," *Proc. of IEEE Workshop on Neural Networks for Signal Processing IV,* Ermioni, Greece, pp. 105-114, 6-8 Sep. 1994.

[6] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, M. Gabbouj, "A class of order statistics Learning Vector Quantizers", *Proc. of IEEE ISCAS'94* , vol. 6, pp. 387 - 390, London, 1994.

[7] J. Y. A. Wang, E. H. Adelson, "Layered representation for motion analysis," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* New York, USA, pp. 361-366, June 1993.

[8] T. S. Huang, G. J. Yang, G. Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. on Acoust., Speech and Sig. Proc.,* vol. ASSP-27, no. 1, pp. 13-18, 1979.