Modified Self-Organizing Map for Optical Flow Clustering System

B. DOUNGCHATOM, P. KUMSAWAT, K. ATTAKITMONGKOL AND A. SRIKAEW
Robotics & Automation for Real-World Applications Research Unit, Intelligent System Group
School of Electrical Engineering
Suranaree University of Technology
111 University Avenue, Muang District, Nakhon Ratchasima 30000
THAILAND
http://eng.sut.ac.th/ee

Abstract: - Optical-flow field analysis is one of the most efficient tools for segmenting moving objects in an image sequence, especially when a camera itself is also moving. Object segmentation from the optical flow can be considered as a clustering problem. The performance of clustering method can significantly improve results of moving object segmentation. This work presents the unsupervised clustering system using a modified self-organizing feature map (MSOFM) neural network. The network can automatically perform clustering without having any priori knowledge of any initial number of clusters or any initial spatial position. It also can be adjustable to achieve multi-resolution clustering. This allows the proposed network to segment flows of multiple moving objects having nearly same speeds. The system shows desirable results of segmentation of moving objects in the camera-moving image sequence. Results and discussions of adjustable capability of the network are also presented.

Keyword: - Artificial Neural Network, Self-Organizing Feature Map, Clustering, Segmentation, Optical Flow

1 Introduction

In video processing, optical-flow field analysis is one of the most efficient tools for moving objects segmentation, especially in the case when a camera itself is also moving. Object segmentation from the optical flow can be considered as a clustering problem. Besides the performance of the optical flow calculation, the performance of clustering method can significantly improve the segmentation. There has been many data clustering researches. The k-mean method in [9] determines distances between data points and centers of clusters. These distances are used to decide class of cluster by comparing with the minimal average distance and then readjusting the center of cluster. This method, however, requires initial number and positions of clusters. The fuzzy c-Mean [7] applies fuzzy mechanism for readjusting centers of clusters. In [8], the hierarchical clustering method is introduced. It is a sequential approach which provides clustering at many different numbers of clusters along the way. The method uses an agglomerative (clumping) or divisive (splitting) approach and requires distance function for calculating distance between clusters. There are also many works that focus on using artificial neural networks for data clustering such as adaptive resonance theory network in [1-2]

In optical flow field of study, data clustering has been deployed to cluster optical flow fields. In [5], the new EXIN segmentation neural network has been used to automatically clustering optical flows from a still camera. The probabilistic clustering has been utilized in [4]. The inputs of such system are optical flows and image brightness. The Gauss-Newton optimization method is applied to search for the optimal parameters of clustering procedures. In [9], multi-resolution optical flows are computed by a complex discrete wavelet transform. The affine parametric motion model of these flows are determined and used for clustering by the competitive agglomeration algorithm which yields the optimum number of clusters and the center of each cluster. The results are segmented using classical mixture model and the expectation-maximization algorithm. In [6], the region-based nonparametric video object segmentation method has been proposed. In order to be more accurately segmented the video objects, the pre-clustering is processed to remove small and isolated regions based on texture characteristic of those regions. The optical flows are then computed and clustered using fuzzy c-mean. The smoothed motion fields are clustered again and the undesirable flows are eliminated by judging from theirs area size.

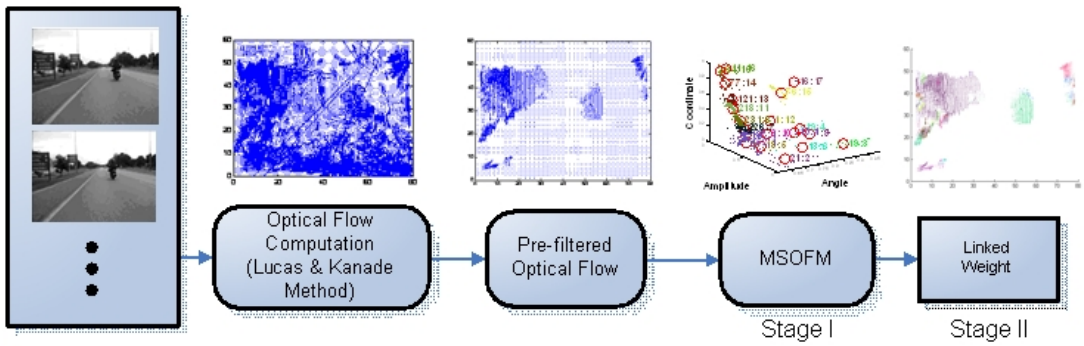


Fig. 1 Optical flow clustering system using modified self-organizing feature map

This paper presents the unsupervised clustering system using a modified self-organizing feature map (MSOFM) neural network. The network can automatically perform clustering without having any priori knowledge of any initial cluster numbers or initial spatial position. The most favorable feature of the proposed network is that it is adjustable to achieve multi-resolution clustering. This allows the network to segment flows of multiple moving objects having nearly same speeds. Details of the system are described in the following sections.

2 Optical Flow Clustering System Using Modified Self-Organizing Feature Map

In order to achieve an automatic clustering system, the unsupervised self-organizing feature map (SOFM) network is applied for the optical flow clustering. Fig. 1 displays an overall system. The Lucas-Kanade optical flow is filtered to eliminate some miscalculated flow. The modified SOFM then performs the clustering process in stage I. After that, all clusters from stage I is then linked to group similar clusters together in stage II. Details of both stages are discussed next.

3 Modified Self-Organizing Feature Map

The main engine of clustering optical flow in this work is the modified self-organizing feature map (MSOFM). A goal of using this modified network is to be able to obtain flow clustering without any supervised learning or any initial number of flow clusters. The network is composed of two main layers: comparison layer and weight layer. The comparison layer is responsible for comparing incoming flow feature vector with stored weights from the weight layer. If there is no weight that is efficiently similar to the input flow feature vector, the network then stores it as a new weight in the weight layer. Otherwise, the closest stored weight will get updated and the network continues with the remaining input flow feature vectors. This coarse clustering of flow feature vectors will be analyzed and grouped to achieve finer clustering in the later process called a weight linking. The structure of this modified network is displayed in Fig. 2.

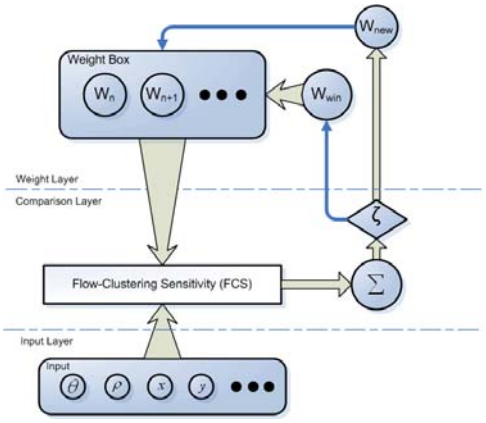


Fig. 2 Network structure of MSOFM.

The network input layer receives optical flow vector as the input to the network. This optical flow vector, called flow feature vector λ, consists of four elements which are a flow angle, a flow magnitude, and flow spatial position x and y as given by

$$\lambda_n = [\theta_n \quad \rho_n \quad x_n \quad y_n]^T$$

where

θ_n = optical flow angle
ρ_n = optical flow magnitude
(x_n, y_n) = optical flow spatial position
n = $1,2,3,\ldots,N$
N = total optical flows.

The MSOFM starts clustering process with no initial weight. The first neuron or weight then comes from the first input flow feature vector. The network stores this first weight as W_1 in the weight layer. The new input flow feature vector is then compared with all weights in the weight layer. This process occurs in the comparison layer. The comparison of these flow feature vectors is performed based on a flow-clustering sensitivity $\bar{s}_{FC}$.

$$\bar{s}_{FC} = [\alpha \quad \beta \quad \sigma \quad \gamma]^T$$

where

α = optical flow angle sensitivity
β = optical flow magnitude sensitivity
σ = optical flow horizontal sensitivity
γ = optical flow vertical sensitivity

This flow-clustering sensitivity provides different levels of comparison for each element in the flow feature vector. The comparison can be carried out by determining the Euclidean distance between the flow feature vector and each of weight vectors. The following relationship is deployed to perform such comparison.

$$\mu_m = \left| \bar{s}_{FC} \cdot (W_m - \lambda_n) \right|$$

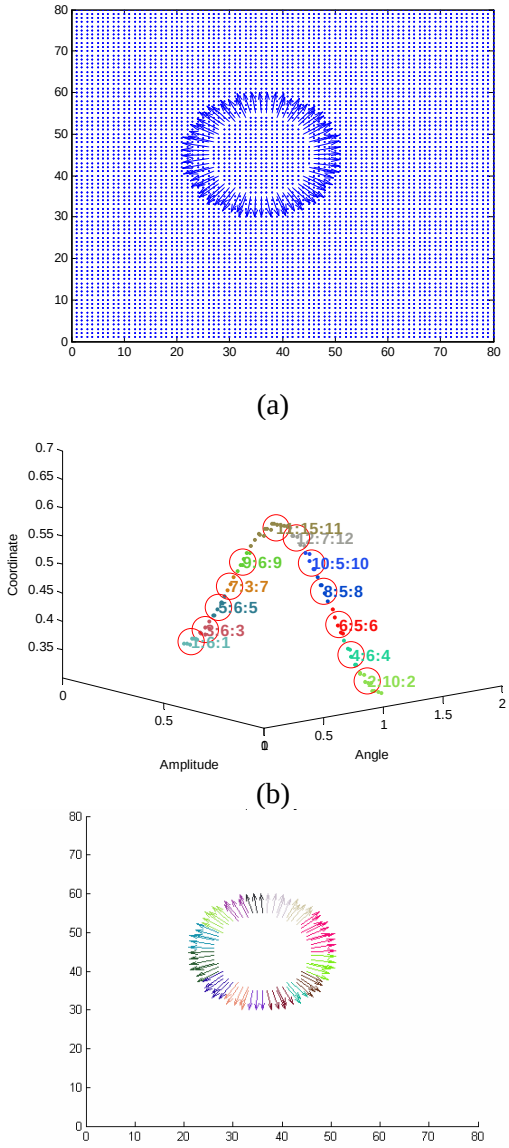
where $m = 1,2,3,\ldots,M$ and M is the number of weight vectors in the network memory. The value of each μ_m is used for making decision whether or not the flow feature vector n is sufficiently similar to the weight vector W_m. Applying flow-clustering sensitivity $\bar{s}_{FC}$ in the comparison process introduces the multi-resolution clustering scheme.

Let μ' is the minimum value among all μ_m from comparing the flow λ_n with the weight vector W'. This μ' is then compared with the vigilance value ζ and leads the network to two actions. Firstly, if $\mu' < \zeta$, this indicates that the flow feature vector is not sufficiently similar to any weight vector in the network memory. The network then creates new weight corresponding to this flow feature vector and classifies it as a new cluster, i.e. $W_{M+1} = \lambda_n$. Secondly, if $\mu' \geq \zeta$, this suggests that the flow feature vector is the most similar to the corresponding weight vector W'. By following the winner-take-all learning scheme, the weight vector W' is then the only weight to get updated by using the following equation.

$$W' = W' + \eta[W' - \lambda_n]$$

where η is the learning constant.

This first stage of optical flow clustering proceeds until the last flow λ_N. An example of flow clustering by the presented network is demonstrated in Fig. 3. In the second stage, all clusters will be determined for connectivity. Details are described in the next section.



(a)

(b)

(c)

Fig. 3 An example of flow clustering using MSOFM (a) a synthesis image of optical flow (b) a result of flow clustering in a flow feature domain (the number represents *time order of neuron construction*: *number of flows in the neuron*: *number of neuron cluster* (c) a result of flow clustering in a spatial domain.

4 Weight Linking

After all optical flows are assigned to clusters, i.e. weights, some of clusters can be considered to have some similarity. These clusters are said to be connected and should be classified to the same moving object that causes the same flows. This work presents a weight linking by using similarity matching approach. The similarity is measured by the Euclidean distance between each cluster. This distance is weighted by flow-clustering linking sensitivity $\bar{s}_l$ as given by

$$\bar{s}_l = [\alpha_l \quad \beta_l \quad \sigma_l \quad \gamma_l]^T$$

where the parameter α_l, β_l, σ_l, and γ_l are again the flow angle, flow magnitude, flow horizontal and flow vertical sensitivity, respectively. The matching of similarity can then be computed by

$$d_s(i) = \left| \bar{s}_l \cdot (W_i - W_j) \right|$$

where $i \neq j$. Consider all flow's cluster index in $G = \{i\}$ where $i = 1,2,3,\ldots,M_f$ and M_f is the total number of clusters in the network. The similarity threshold ε can be chosen for linking each cluster as the following steps. At first, compute $\delta = \{i : d_s(i) < \varepsilon\}$. Next, calculate $k = \min(\delta)$ and then sort all flow's cluster index by using the following relationship.

$$G = \begin{cases} k, & i \in \delta \\ i, & i \notin \delta \end{cases}$$

The above process iteratively continues until no change in G. Each cluster remained in G represents flows that come from the same moving object with resolution specified by $\bar{s}_l$ and ε.

5 Experimental Results

In order to investigate the proposed network, the experiments are performed in various configurations. Numbers that display in the flow clustered image are the order of the neuron construction. Circles exhibit the center of flow clusters. Fig. 4 shows normalized synthesis flows that are clustered to investigate on the vigilance parameter ζ, the flow horizontal sensitivity σ, and the flow vertical sensitivity γ. The values of $\sigma = 1.0$ and $\gamma = 1.0$ indicate that these flows are fully sensitive to spatial position x and y, respectively. Fig. 5 and 6 demonstrate assigning both σ and γ to be 0.4. Results show that flows are less sensitive to x direction (Fig. 5) and y direction (Fig. 6). This allows the flows to be further analyzed based on theirs region in the image which can be very useful in some applications, e.g. the application of lane detection in which the areas of an image are examined in different meaning.

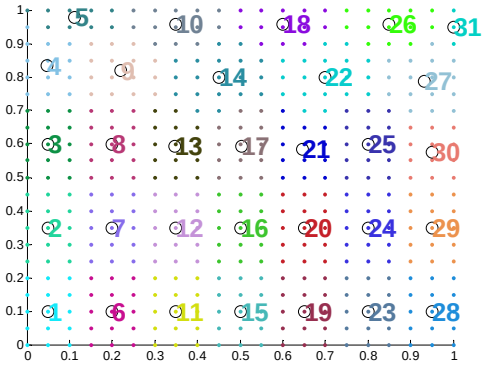


Fig. 4 Normalized flows with $\zeta = 0.13$, $\sigma = 1.0$, and $\gamma = 1.0$.

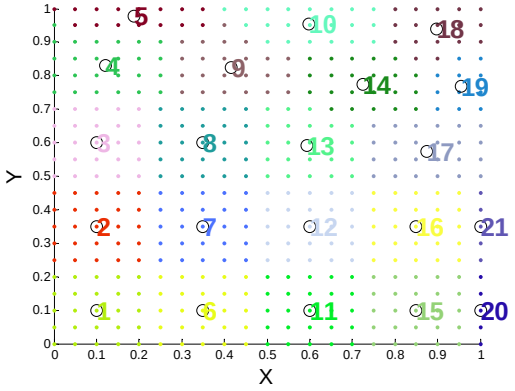


Fig. 5 Normalized flows with $\zeta = 0.13$, $\sigma = 0.4$, and $\gamma = 1.0$.

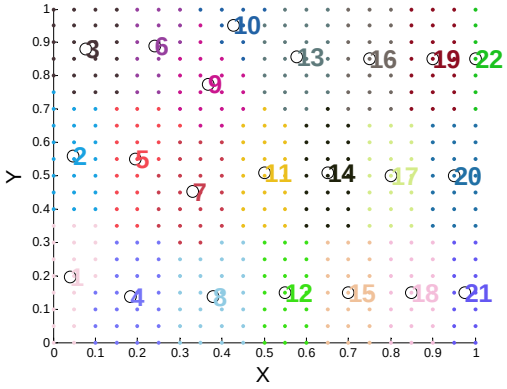


Fig. 6 Normalized flows with $\zeta = 0.13$, $\sigma = 1.0$, and $\gamma = 0.4$.

Fig. 7 and 8 display results of $\zeta = 0.5$ and $\zeta = 0.07$. Obviously, this vigilance parameter allows the different number of neuron being constructed. The higher value indicates the coarse flows in the same cluster while the smaller value indicates the similar flows in the same cluster.

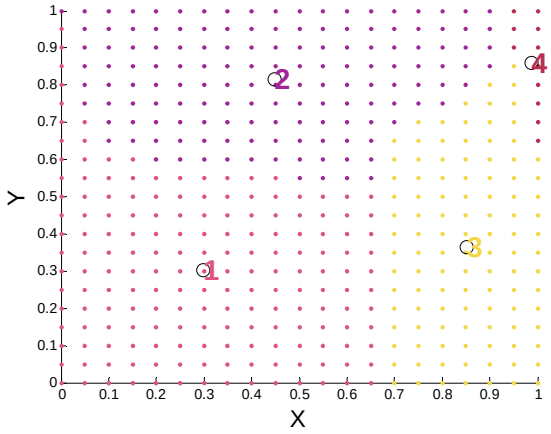


Fig. 7 Normalized flows with $\zeta = 0.5$, $\sigma = 1.0$, and $\gamma = 1.0$.

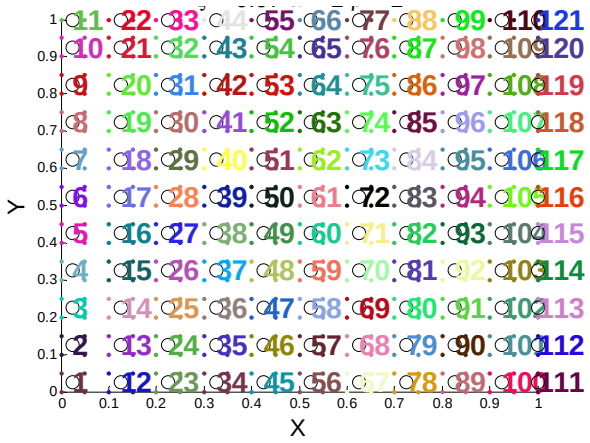


Fig. 8 Normalized flows with $\zeta = 0.07$, $\sigma = 1.0$, and $\gamma = 1.0$.

In order to investigate the flow angle sensitivity α and the flow magnitude sensitivity β, the synthesis flows in Fig. 9 are converted to polar coordinate in Fig. 10. Fig. 11 and 12 show the flow clustering with $\zeta = 0.3$, $\alpha = 1$, and $\beta = 0$ (fully sensitive to the flow angle). By adjusting α and β, results of flow clustering can be seen in Fig. 11 to Fig. 16. Note that the result of clustering with $\alpha = 0$ in Fig. 15. This demonstrates that all flows are clustered with no angle sensitivity. Hence, all flows are located on a horizontal line. The results of different ζ are depicted in Fig. 17 and 18 (coarse clustering with $\zeta = 0.6$) and Fig. 19 and 20 (fine clustering with $\zeta = 0.1$).

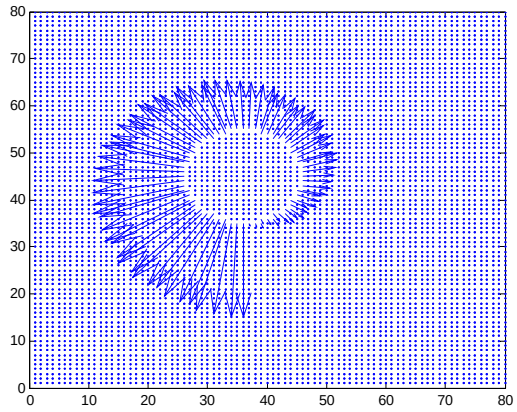


Fig. 9 Synthesis flows

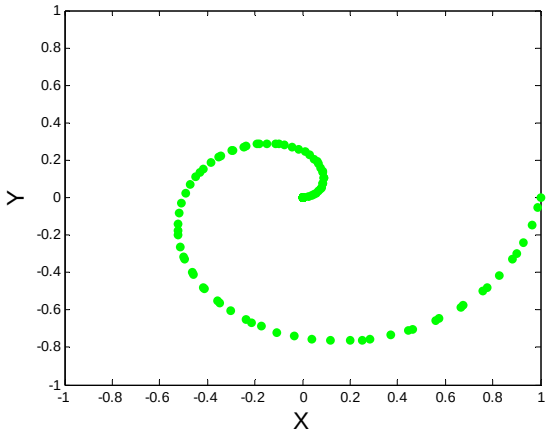


Fig. 10 Flows from Fig. 9 in polar coordinate

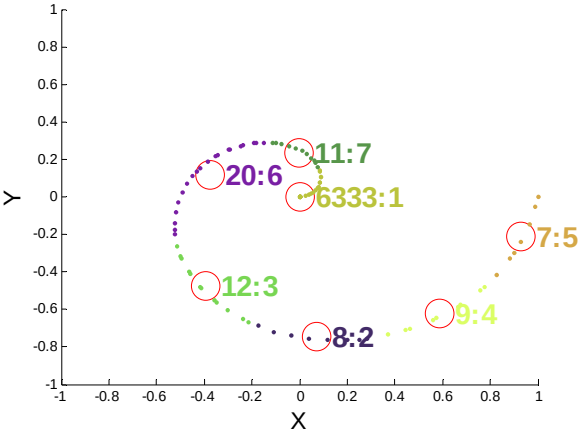


Fig. 11 Flow clustering (*number of flows* : *neuron*) with $\zeta = 0.3$, $\alpha = 1$, and $\beta = 0$.

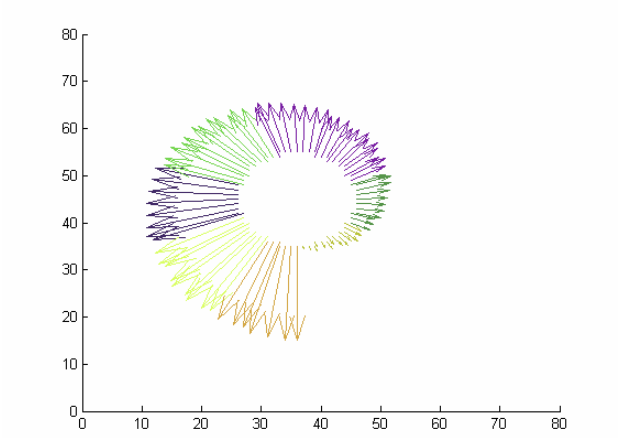


Fig. 12 Flow clustering of Fig. 11 in Cartesian coordinate

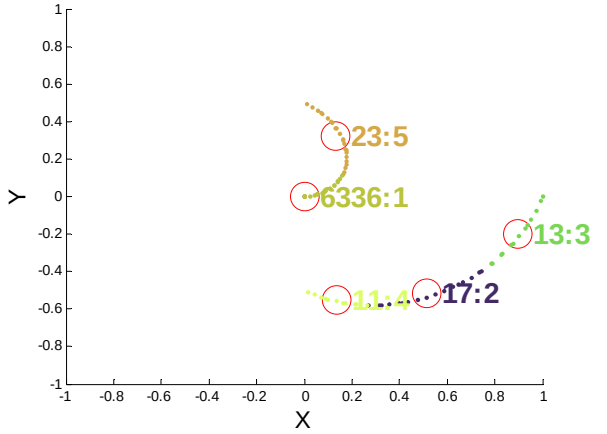


Fig. 13 Flow clustering with $\zeta = 0.3$, $\alpha = 0.5$, and $\beta = 0$

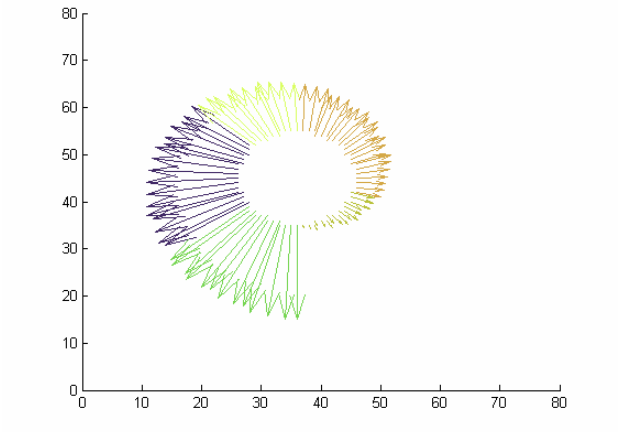


Fig. 14 Flow clustering of Fig. 13 in Cartesian coordinate

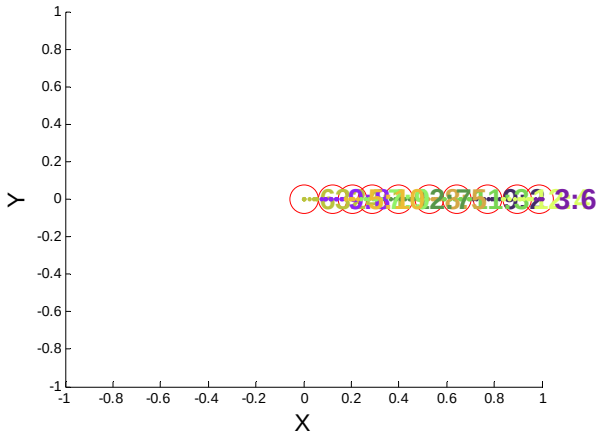


Fig. 15 Flow clustering with $\zeta = 0.3$, $\alpha = 0$, and $\beta = 0.5$

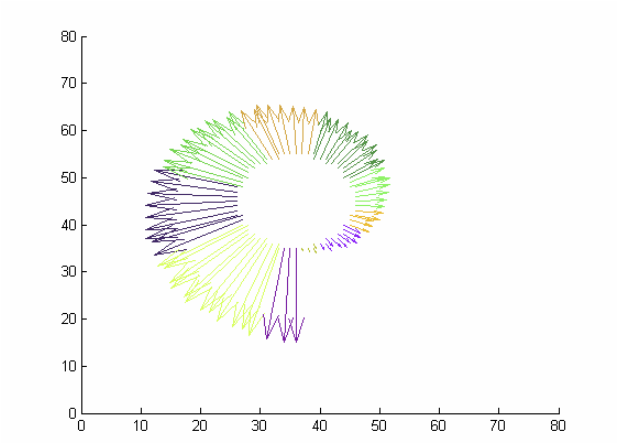


Fig. 16 Flow clustering of Fig. 15 in Cartesian coordinate

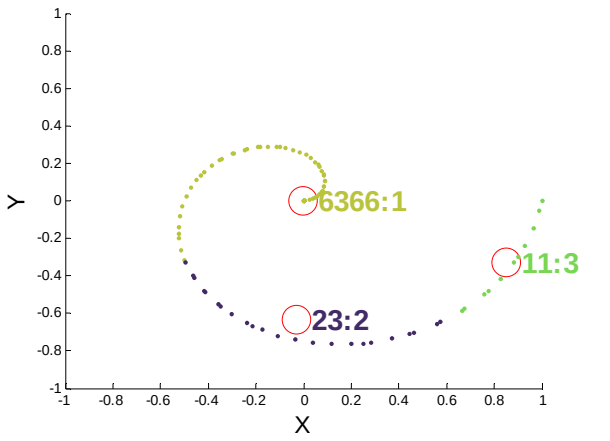


Fig. 17 Flow clustering with $\zeta = 0.6$, $\alpha = 1.0$, and $\beta = 1.0$

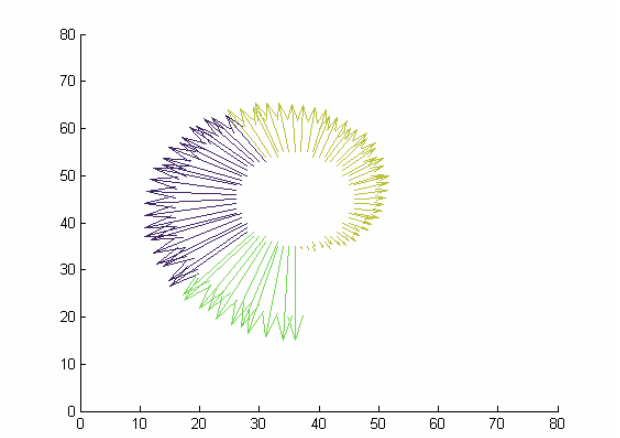


Fig. 18 Flow clustering of Fig. 17 in Cartesian coordinate

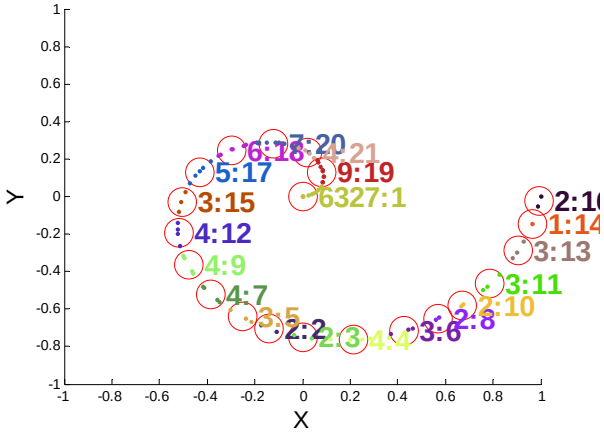


Fig. 19 Flow clustering with $\zeta = 0.1$, $\alpha = 1.0$, and $\beta = 1.0$

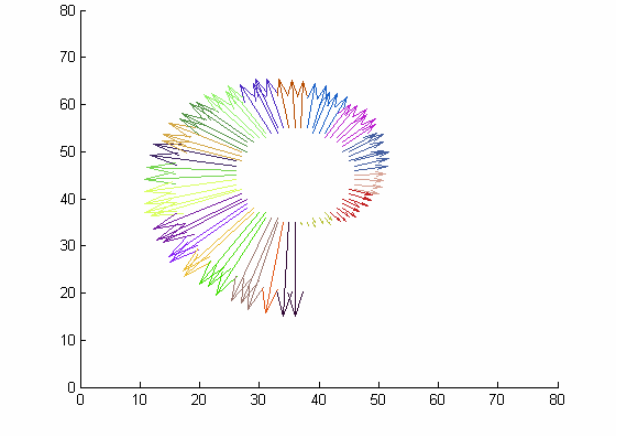
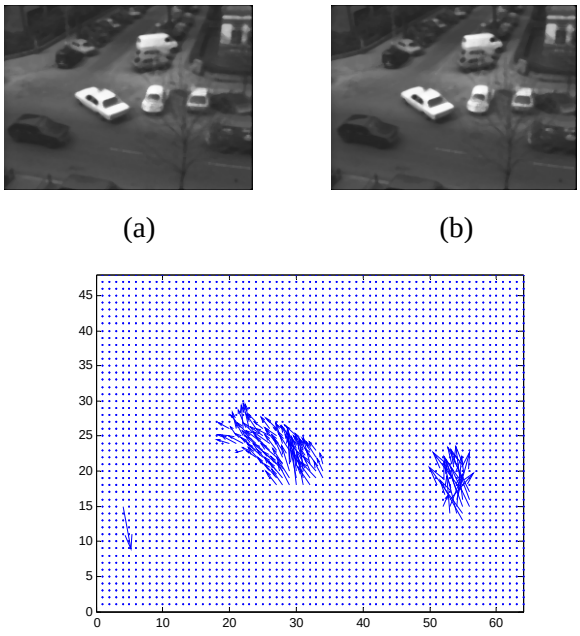


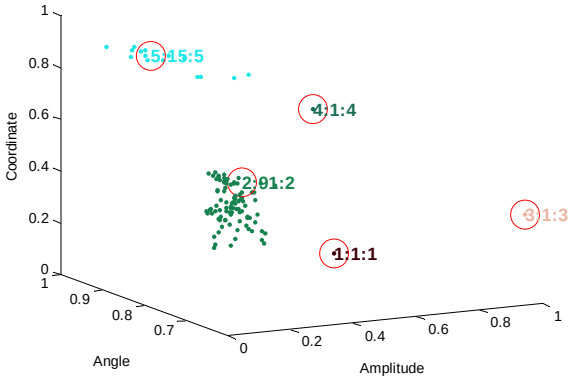
Fig. 20 Flow clustering of Fig. 19 in Cartesian coordinate



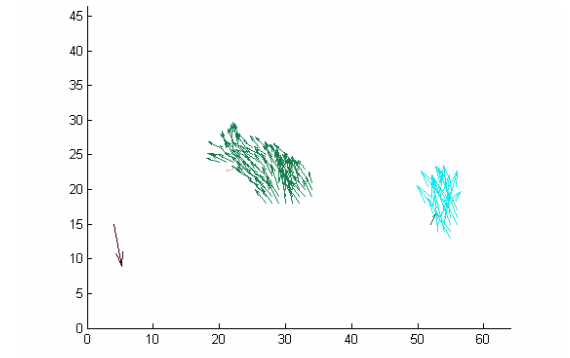
(a) (b)

(c)

Fig. 21 Lucas-Kanade optical flow of images from (a) and (b)



(a)



(b)

Fig. 22 Result of optical flow clustering

Fig. 21 shows Lucas-Kanade optical flow of classic Hamburg Taxi images. The resulting flow clustering is shown in Fig. 22 where $\lambda = [\theta \quad \rho \quad x]^T$ is used

for a purpose of displaying. The parameters of the network are shown as follows:

$\bar{s}_{FC} = [0.5 \quad 1 \quad 2]^T \; \zeta = 0.25 \; \bar{s}_l = [0.4 \quad 0.8 \quad 2]^T$
$\varepsilon = 0.26$

Fig. 24 shows results of optical flow clustering from images with camera installed on a moving vehicle (see Fig. 23). Flows from such sequence of images come from both background and moving objects. The flow feature vector $\lambda = [\theta \quad \rho \quad x \quad y]^T$ is utilized with the following parameters.

$\bar{s}_{FC} = [1.5 \quad 1.8 \quad 1.5 \quad 1.5]^T \; \zeta = 0.3$
$\bar{s}_l = [0.8 \quad 0.5 \quad 1 \quad 1]^T \; \varepsilon = 0.3$

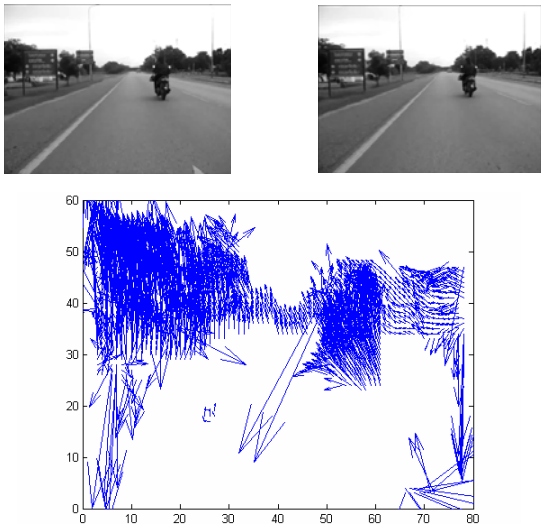


Fig. 23 Lucas-Kanade optical flows from moving camera

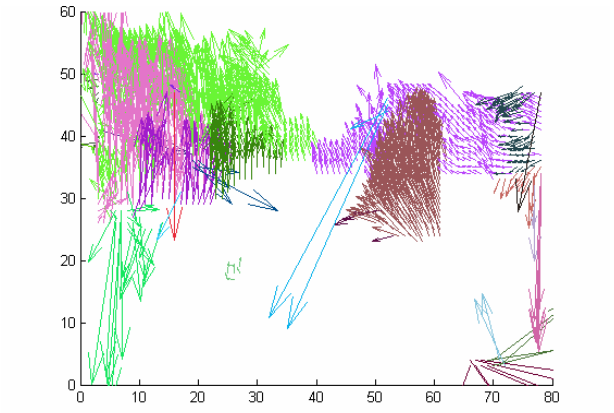


Fig. 24 Optical flow clustering before weight linking

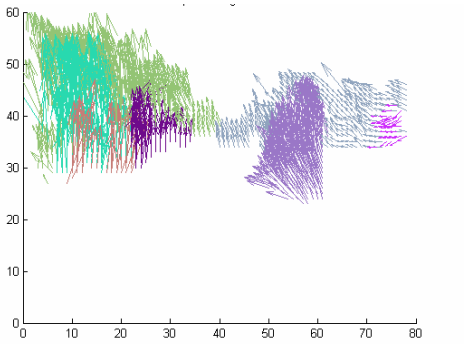


Fig. 25 Optical flow clustering after weight linking

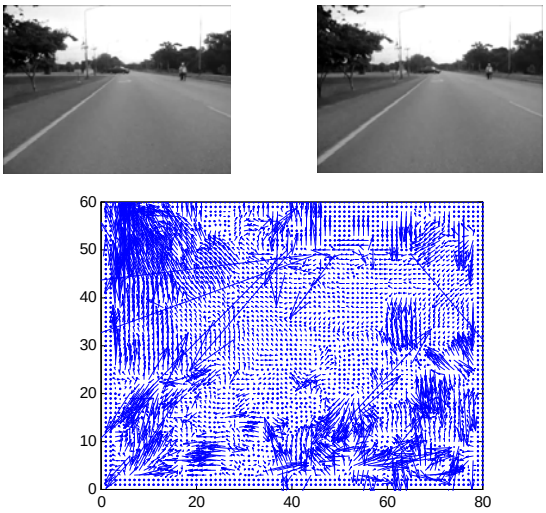


Fig. 26 Optical flows from moving camera

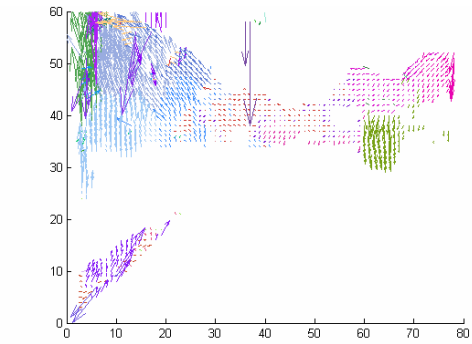


Fig. 27 Flow clustering of Fig. 26

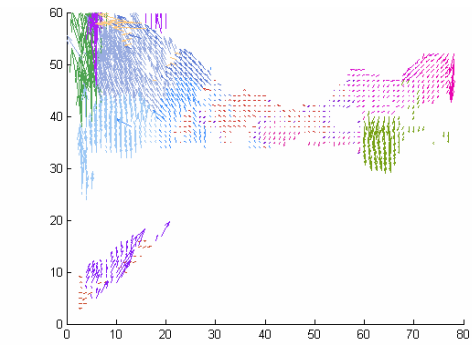


Fig. 28 Flow clustering with elimination of clusters with number of flows less than 10

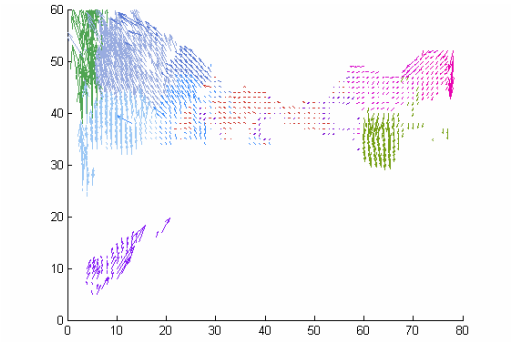


Fig. 29 Flow clustering with elimination of clusters with number of flows less than 50

Results of flow clustering can be adjusted by the value of $\bar{s}_{FC}$. When the value of $\bar{s}_{FC}$ is small, it indicates the less sensitivity between flows. The new weight or neuron has less probability to be created. Each element in $\bar{s}_{FC}$ allows the network to be emphasis independently. This effect is the same for choosing the value of $\bar{s}_l$ in the stage II of flow clustering. The values of ζ and ε also provide the coarse and fine clustering results. The higher values results in the more differences between the clusters. Note that the misestimated flows are also filtered out by the adjustment of these network parameters. Additionally, flows from Fig. 26 can be smoothed by eliminating clusters that have number of flows less than some threshold. This shows that the presented system is highly adjustable which allows deploying this network to various kinds of applications.

6 Conclusion

This work presents the automatic optical flow clustering system. The modified self-organizing feature map is used as a clustering system. This modified network is able to efficiently cluster flow vectors without any priori knowledge of initial number of flow clusters. The proposed system is also highly adjustable for flow angles, magnitudes, and spatial positions allowing the system to be flexible for using in different purposes. This also yields the ability to smooth out misestimated flow at some degrees. The results of flow segmentation are desirable and showing a multi-resolution clustering capability of the system. This allows the network to be able to segment flows of multiple moving objects having nearly same speeds. Further work can be focused on investigation relationship of each parameter of the network to obtain the more suitable network for different kinds of applications.

Reference:

[1] A. Baraldi, E. Alpaydin, Constructive Feedforward ART clustering networks I, *Neural Networks, IEEE Transactions*, Vol.13, issue 3, 2002, pp.645-661.

[2] A. Baraldi, E. Alpaydin, Constructive Feedforward ART clustering networks II, *Neural Networks, IEEE Transactions*, Vol.13, issue 3, 2002, pp.662–677.

[3] B.D. Lucas and T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision, *Proceedings of Imaging understanding workshop*, 1981, pp.121-130.

[4] C.K. Cheong, K. Aizawa, Structural motion segmentation based on probabilistic clustering, *Image Processing, Proceedings, International Conference*, Vol.1, 1996, pp.505 – 508.

[5] G. Cirrincione and M. Cirrincione, A Novel Self-Organizing Neural Network for Motion Segmentation, *Applied Intelligence*, Vol.18, No.1, 2003, pp.27-35.

[6] Kai-Kuang Ma, Hai-Yun Wang, Region-based nonparametric optical flow segmentation with pre-clustering and post-clustering, *Multimedia and Expo, ICME '02.Proceedings, IEEE International Conference*, Vol.2, 2002, pp.201 – 204.

[7] M.V Modenesi, M.C.A. Costa, A. Evsukoff, N.F.F. Ebecken, Parallel Fuzzy c-Means Cluster Analysis, *VECPAR'06*, 2006.

[8] P. Willet, Recent Trends in Hierarchical Document Clustering: a Critical Review, *Process, Management*, 1988, pp.577-597.

[9] S. Nitsuwat, J.S. Jin, H.M Hudson, Motion-based video segmentation using fuzzy clustering and classical mixture model, *Image Processing, Proceedings, International Conference*, Vol.1, 2000, pp. 300 – 303.

[10] S. Kantabutra and Alva L. Couch, Parallel K-means Clustering Algorithm on NOWs, *NECTEC Technical Journal*, Vol.1, No.6, 2000.