Detecting Dominant Motion Patterns in Crowds of Pedestrians

Muhammad Saqib[a], Sultan Daud Khan[b], Michael Blumenstein[a]
[a] Griffith University, School of ICT, Institute of Integrated and Intelligent Systems, Australia
[b] Makkah Techno Valley, Saudi Arabia

ABSTRACT

As the population of the world increases, urbanization generates crowding situations which poses challenges to public safety and security. Manual analysis of crowded situations is a tedious job and usually prone to errors. In this paper, we propose a novel technique of crowd analysis, the aim of which is to detect different dominant motion patterns in real-time videos. A motion field is generated by computing the dense optical flow. The motion field is then divided into blocks. For each block, we adopt an *Intra-clustering* algorithm for detecting different flows within the block. Later on, we employ *Inter-clustering* for clustering the flow vectors among different blocks. We evaluate the performance of our approach on different real-time videos. The experimental results show that our proposed method is capable of detecting distinct motion patterns in crowded videos. Moreover, our algorithm outperforms state-of-the-art methods.

Keywords: Crowd flow, dense optical flow, dominant motion patterns, clustering.

1. INTRODUCTION

Public safety and security is becoming a crucial problem in most urban areas. Large events such as marathons, religious festivals, sports and political gatherings attract thousands of people in a constrained area. Despite safety measures, crowd disasters still occur frequently. A crowd can be characterized by the complex behavior, exhibited by the motion of individuals as well as the motion caused by their interactions. As a result of interactions, conflicting motion patterns may lead to dangerous situations such as stampede, congestion, bottleneck that pose potential threat to the safety of pedestrians. Therefore, segmenting motion patterns provide us useful information to infer about high-level scene semantics. The segmentation is the important step in the general implementation pipeline of visual analysis of crowd. The segmented motion patterns can also be used to initialize and validate simulation models. Conventional video analysis methods use detection of objects and subsequently tracking the objects for understanding the behavior of the crowd. However, these methods are limited to the scenes with the sparse crowds of completely visible pedestrians. While, in high-density crowded scenes, frequent partial and complete occlusions make detection and tracking difficult and inapplicable. Therefore, as a solution, researchers adopt holistic approaches to gather global motion information from the scene. These holistic approaches are based on the computation of dense optical flow or sparse key-point descriptors, e.g. SIFT to extract motion information. These approaches tend to be robust even when crowd is large with substantial occlusions. A crowd flow segmentation framework based on Lagrangian particle dynamic to segment crowd flows is proposed in [1]. The Lagrangian particle dynamic system finds Lagrangian Coherent Structures (LCS) which divide the flows into different coherent regions. The notion of Lagrangian Coherent Structures is synonymous to edges in the images. But their approach is complex and computationally expensive and cannot detect small flows related to small group of pedestrians. [2] used sparse SIFT features to extract motion information from the scene. Flow map is generated after computing flow vectors based on SIFT features. Flow map is then divided into small non-overlapping blocks and within each block, dominant flows are estimated by employing spatial and orientation based clustering of flow vectors. [3] proposed approach for crowd flow segmentation based on computation of sparse optical flow followed by spectral clustering. A min cut/max segmentation algorithm is employed in [4] to segment the crowd flows. First, the foreground is extracted from video using a Gaussian Mixture Model (GMM). The frame is divided into fixed size blocks, and the correlation among the blocks is computed in a spatiotemporal domain. To remove noise and to have more consistent representation of crowd flow, a grid of the particle is initialized on the foreground mask and tracked using pyramidal Lucas Kanade approach. Crowd motion is segmented by employing multiple local features in [5], where the foreground pixels are accumulated by estimating the number of pedestrians passing through a virtual strip wire. In all of the above mentioned approaches, we cannot find clear boundaries among different flows after segmentation. A crowd flow segmentation approach based on histogram analysis of optical flow orientations is proposed in [6]. The derivative of histogram curve is used to segment crowd flows. Since this method considers only peaks of the histogram curve,

Eighth International Conference on Graphic and Image Processing (ICGIP 2016), edited by Tuan D. Pham, Vit Vozenilek, Zhu Zeng, Proc. of SPIE Vol. 10225, 102251L · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2266825

therefore it loses information about the crowd flows. [7] Segments the crowd flows by employing k-means clustering followed by blob absorption approach. This approach detects clear boundaries among different flows and detects large as well as small flows, but the algorithm suffers from estimating the hyper parameter k (number of clusters), which is determined experimentally. In this paper, we proposed an approach for crowd flow segmentation, where motion flow field is generated by computing dense optical flow using Brox and Sand method [8]. Motion flow field is then divided into non overlapping blocks. Local and global flows are estimated by adopting a novel 2-stage clustering algorithm. After clustering, small clusters (blobs) appear at the boundaries of different flows which are smoothed by employing blob absorption approach. Comparing to other approaches, our approach can detect small as well as large motion segments and by employing blob absorption method, we generated clear boundaries between different flows. Moreover, unlike [7], our algorithm does not require hyper parameter settings. Our main contributions are: (1) Two stage clustering algorithm for detecting local and global flows (2) A simple, yet effective method to segment crowd flows without detection and tracking. (3) Does not require training. The paper is organized as follows: the following section discusses proposed methodology. In Sect. 3 we discuss the experimental results. Finally, conclusions end up the paper.

2. PROPOSED METHODOLOGY

In this section, we discuss the proposed methodology for extracting motion patterns from the given sequence of frames. Given a sequence of frames, the main steps involved in extracting the motion patterns are: (1) optical flow field computation followed by construction of the motion field; (2) dividing the motion field into non-overlapping blocks and (3) finding clusters by employing our proposed 2-stage clustering algorithm; (4) the final clusters are smoothed by employing blob absorption approach. The details of the proposed method is described below.

2.1 Motion Field Computation

At this step, we compute the optical flow between two adjacent frames of the given sequence followed by the construction of motion field. Let $f(x, y, t)$ and $f(x, y, t+1)$ be the two consecutive frames of the given sequence, we compute the dense optical flow by employing the Brox and Sand (BS from now on) method followed by a Median filter and Gaussian filter in order to remove noise. After computing the optical flow, the next step is to construct a motion field. The motion flow field is the set of independent flow vectors, and each flow vector is associated with its spatial location. The motion flow field contains the instantaneous and temporal information of the scene and can be used for learning motion patterns of the given video. In the next step, we employ our proposed 2-stage clustering algorithm described in the following section.

2.2 Two-Stage Clustering

In order to characterize dominant motion patterns in the scene, we need a clustering algorithm that groups the similar flow vectors belonging to a motion pattern in the scene. Before applying our clustering algorithm on the motion flow field, we first divide the motion flow field into $N X M$ blocks. Let $\{b_1, b_2, ..., b_n\}$ in B is the list of blocks, where each block b_l is of size $i x j$. Our proposed algorithm involves two steps, (i) *Intra-clustering* and (ii) *Inter-clustering*. At the *Intra-clustering* level, we cluster the flow vectors within the block. Whilst in *Inter-clustering*, we cluster flow vectors among the blocks. In the following, we summarize our *Intra-clustering* algorithm.

1. For every block b_l in B, select a set of points whose magnitude Ω_k is greater than a threshold η. Let P be the set of points (or flow vectors) whose magnitude is greater than Ω_k.
2. Initially we assume each point of the block as a separate cluster and we assign a distinct identifier to each point of the block.
3. Compute the centre of the cluster by computing the circular mean μ of each cluster as in [9]. Let $C = \{c_1, c_2, ..., c_n\}$ be a set containing a list of clusters for all n number of blocks, where $c_i = \{\mu_1, \mu_2, ..., \mu_k\}$, is a vector of k number of clusters detected in block b_i.
4. Compute a distance matrix D, where each element of the matrix is the distance between circular means μ_i and μ_j, given by $D_{i,j} = \mu_i - \mu_j$.
5. Compute $d_{min} = argmin\ D$ for all $i \neq j$.
6. If d_{min} is less than σ, then cluster the flow vectors whose distance between the means is d_{min} and continue to step 3 through step 6. Otherwise go to step 7.

7. Repeat the same steps (1-6) until all the blocks in set *B* are processed.

A set of clusters local clusters C are achieved after the first stage of cluttering. Now these local clusters C are then combined into global clusters G (corresponds to large flows) by employing the second stage of the clustering algorithm. Pseudocode of inter clustering algorithm is described in Fig. 1.

```
Input: Set of Local Cluster C
Output: Global Cluster G
Function INTERCLUST(C)
    Initialize array G as empty
    G ← insert the first list of cluster c1 from C
    For all List of cluster c in C do
        G=MergeBlocks(G,c)
    End for
End function
Input: Cluster c1,c2
Output: Updated Cluster G
Function MergeBlocks(G,C)
    Initialize cluster identifier "id" to 1.
    For i=1 to sizeof(G) do
        For j=1 to sizeof(c) do
            if ||G[i]|-c[j]||≤τ then
            μm ← compute the mean of c[j] from G[i]
            G[id]= μm
            Delete c[j]
    End if
    End for
    Increment id by 1.
End for
Return G
End function
```

Figure 1. Inter clustering algorithm for achieving global clusters

2.3 Blob Absorption

After employing two-stage clustering, we noticed that small blobs appear as shown in Fig. 2 (2nd column) which represents small clusters in the scene and resulted due to the following reasons. (1) If the object moves slowly, the inside and outside flow vectors of the object are not the same and hence are classified as two different flows; (2) the optical flow near the boundaries of two conflicting flows is ambiguous. We get rid of these small clusters by adopting a blob absorption approach which follows the following steps:

1. Cluster weights is computed for all the detected clusters, i.e. Let $G_{w,j}$ be the weight for j_{th} cluster computed by $G_{w,j} = \sum_{j=1}^{K} {n_j}/{T_f}$ where n_j is number of feature points in the cluster G_j and T_f is the number of total foreground points.
2. The blob analysis is performed to find the area of the blobs for selected cluster i.e. G_j
3. The small blobs that need to be absorbed are found by comparing the area of all blobs with the threshold value *L*. The threshold area value *L* is selected as *80%* less than the value of big blob in the cluster. Let $S=\{s_1,s_2,s_3...s_n\}$ is the set of blobs whose size is smaller than *L* that need to be absorbed. As the video resolution for all the video is almost same, therefore the value L is kept constant for all the videos.
4. Select blob s_i from set S, edges are found by edge detector for the blob.
5. Search the local neighborhood for the detected edge point and find the nearest neighbor clusters and store their *ids* in *M*. As the blobs cannot be absorbed by itself the cluster *id* of the analysed blob is removed from *M*.

6. Blob weight is computed for all the remaining blobs in M, i.e. Let $BW_{w,i}$ be the weight for the remaining blobs computed by $BW_{w,i} = \sum_{j=1}^{P} {n_j}/{T_M}$, where P is the total number of cluster ids in *M* while n_j is number of feature points with cluster id *j* in and T_M is number of points in *M*.
7. Total weight $W_{t=} G_{w,i} + BW_{w,i}$ is used to select the cluster with *id j*, which will absorb the small cluster and hence *id j* is transferred to all the points of the blob absorbed.
8. Repeat steps 4 to 7 until S is empty
9. Repeat step 2. Here one thing to be noticed that Background is also considered as cluster with *id* and $G_{w,j}$ =0.

3. EXPERIMENTAL RESULTS

In this section we discuss the qualitative results obtained from the experiments conducted. The experiments are carried out on a PC with a 3 GHz core i7 processor and with 8.0 GB memory. We used a publicly available data set from University of Central Florida [3] to evaluate and also be able to compare our results with other state of the art approaches. The data set covers variety of crowd scenes from real-world events such as religious festivals, road cross and marathon. Fig. 3 shows the qualitative results of our experiments. We select 60 frames from each video to evaluate our proposed algorithms. First, optical flow is computed and then our proposed clustering algorithm is applied followed by blob absorption technique [7]. The first column shows the samples from original sequences used in experimentation. The second column shows the results after applying our clustering algorithm. As obvious from the Fig. 3 (2nd column), small blobs appear after segmentation due to the reasons discussed above and are removed through blob absorption approach. The final output of our approach is shown in Fig. 3 (3rd column), where the analyzed scene is segmented in different motion patterns. The extracted motion patterns are color coded based on orientations.

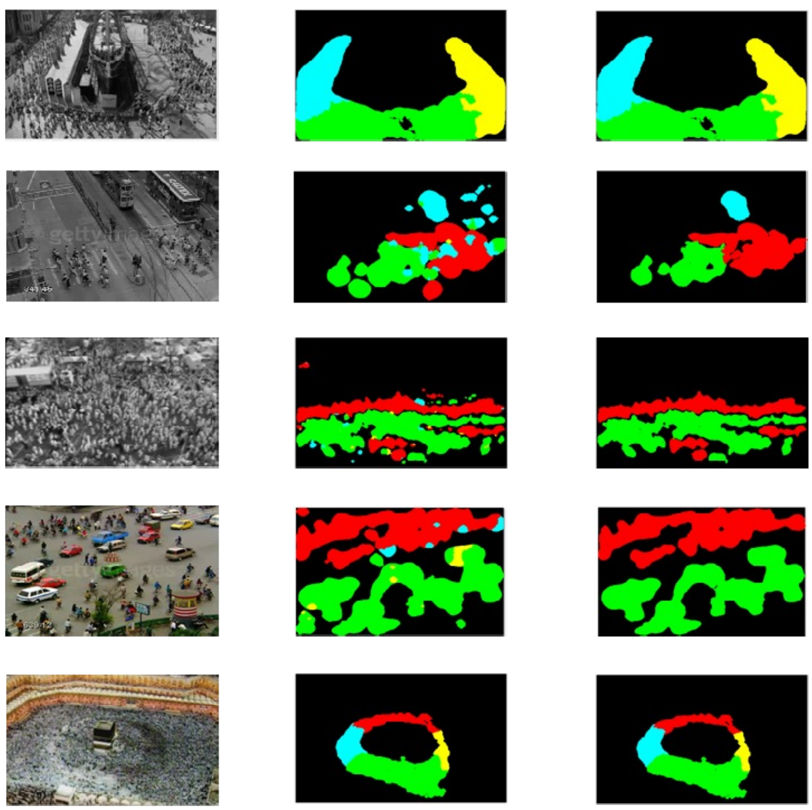

Figure 2. The first column shows the input frames from original sequences. The second column shows the segmented motion patterns of the corresponding input frames, and the third shows the segmented motion after blob absorption.

We qualitatively compare our approach with other state of the art approaches as shown in Fig. 2. As obvious from the Fig. 2 (first row), the multi-label optimization technique [4] is not able to segment crowd flow into dominant flows and boundaries are not clear due to small blobs appear after segmentation. In the second row, we compare our method with histogram curve based segmentation [6]. Histogram curve loss much information about the crowd flow, since it only considers the peaks of histogram curve. In the third row, we compare our results with dynamic segmentation [1] and spectral clustering method [3]. As obvious from the figure, spectral clustering performs segmentation on a sparse optical flow and gives approximate segments and we cannot see clear boundaries among different flow segments. Moreover, dynamic segmentation fails to detect small groups in the crowd. Our proposed algorithms resolve the above mentioned issues by segmenting dominant motion patterns with clear and distinct boundaries without losing useful crowd flow information.

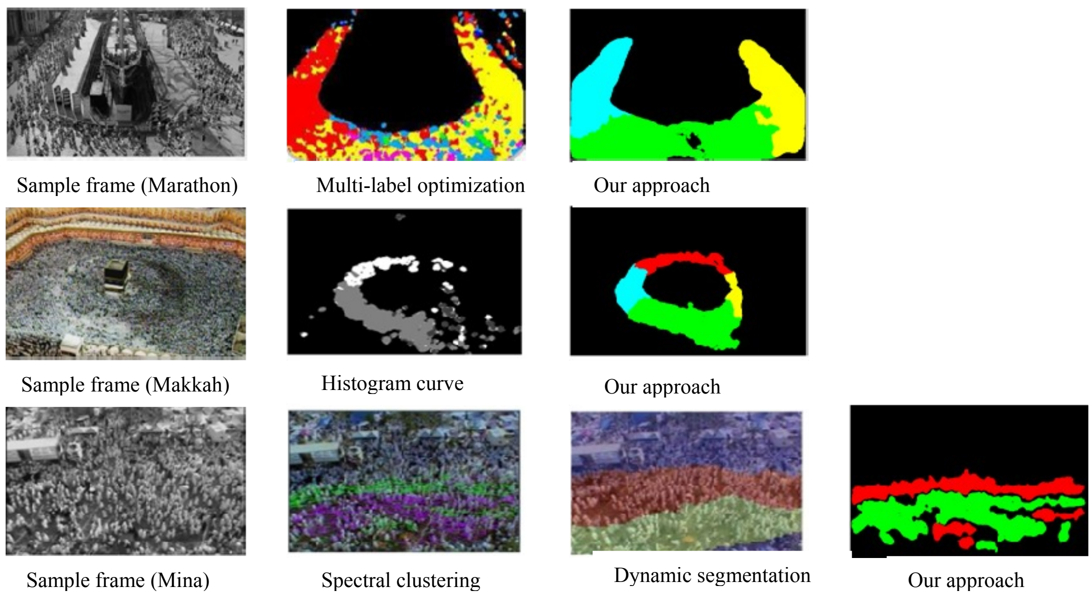

Figure 3. Comparison Results

4. CONCLUSION

In this paper, we used a dataset that covers both high and low density crowded situations and we propose a framework that segments the crowd flow into different meaningful semantic regions. These kind of analysis are very useful for crowd managers in understanding the crowd behavior. Moreover, this analysis also provides a useful input to crowd simulation models for initialization and validation of their models. Our approach does not rely on the traditional methods of detection and tracking and rather adopts a holistic approach by employing optical flow, followed by clustering algorithm. This approach can be applied to different situations and it is independent of the camera view.

REFERENCES

[1] S. Ali and M. Shah, "A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 1-6, (2007)

[2] O. Ozturk, T. Yamasaki and K. Aizawa, "Detecting dominant motion flows in unstructured/structured crowd scenes," Pattern Recognition (ICPR), 2010 20th International Conference on, Istanbul, 3533-3536, (2010)

[3] G. Eibl and N. Brandle, "Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, Tampa, FL, 1-4, (2008)

[4] H. Ullah and N. Conci, "Crowd motion segmentation and anomaly detection via multi-label optimization," in ICPR workshop on Pattern Recognition and Crowd Analysis, (2012)

[5] S. Srivastava, K.K. Ng and E.J. Delp, "Crowd flow estimation using multiple visual features for scenes with changing crowd densities," Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on, Klagenfurt, 60-65, (2011)

[6] W. Li, J.H. Ruan and H.A. Zhao, "Crowd movement segmentation using velocity field histogram curve," 2012 International Conference on Wavelet Analysis and Pattern Recognition, Xian, 191-195, (2012)

[7] S.D. Khan, G. Vizzari, S. Bandini and S. Basalamah, “Detecting dominant motion flows and people counting in high density crowds,” Journal of WSCG, **22**(1):21–30, (2014)

[8] T. Brox, A. Bruhn, N. Papenberg and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” 8th European Conference on Computer Vision, Prague, Czech Republic, 25–36, (2004)

[9] P. Berens and M.J. Velasco, “The circular statistics toolbox for matlab,” MPI Tech. Rep., 184:1–21, (2009)