Volkan OKBAY
1741479

EE 636 Project Preliminary Report

Tracking in High Density Crowds

Tracking in a high density field is challenging due to extreme number objects, interaction, small pixel sizes. This paper [1] coined a new method using three force field models, namely: Static, boundary and dynamic floor fields. The aim is to calculate a probabilistic model depending on correlation, global and local forces that is shaped by motion and crowd behaviors.

The main reason to implement such more complex method is that simpler object detection (edge detection, only correlation etc.) will not be enough to track objects with small sizes. The whole official name for the algorithm is given as such: *the locomotive behavior of an individual in a crowded scene is a function of collective patterns evolving from the space-time interactions of individuals among themselves and with the layout of the scene.*

First point of the view is that objects are combination of particles, which are set of pixels. Those pixel sizes would be a global parameter and adjustable for each scene, video resolution or so. Each individual (object) would have a center pixel indication position on the plane. Time on the other hand, is represented by discrete frame steps, as usual. Using all the information in the given particle set, the objective is to acquire a probability map in the predefined neighborhood window. This window stands for the next possible locations of the object depending on the force fields and some other parameters explained below.

One is the similarity pattern between current and next position pixel maps. Second to fourth parameters are floor fields themselves, and also their constant weights. The main probability equation is :

$$p_{ij} = Ce^{k_D D_{ij}} e^{k_S S_{ij}} e^{k_B B_{ij}} R_{ij}$$

(1) The main equation for probabilistic tracking

Here, i and j represents the cell numbers in the neightborhood. To clarify, those cells may consists of one or more pixels. C constant is normalization constant. R represents similarity pattern, that is nothing but normalized correlation in 2-D. Finally, the rest is floor field factors. The subscripted "k" parameters are experimentally decided: $k_S = 0.02$, $k_B = 0.02$ and $k_D = 0.02$. And D_{ij}, S_{ij}, B_{ij} models can be summarized as follows:

a) Static Floor Field (SFF): Static floor field is calculated by the very first M frames of the footage, because it has heavier computational load than other Ffs have. It stands for the static behavior of the crowd movement, in other words, a global force of the motion. SFF will have a higher value if the crowd has tendency to move to j cell from i. And it will positivelty affect the likelihood to get there in the next time instance. The algorithm models enter and exit points in the whole scene, then gives every cell a value to show relation with exit points (in terms of distance to exit) .So the method for finding SFF has two steps: point flow estimation and sink-seeking.

Point flow field is formed by simple optical flow estimation methods. The aim is to form a 4-element array for each pixel (or cell) that has position and velocity vectors in both dimensions. In the end, every cell array would be in the form of $Z_i = [x_i\ y_i\ Vx_i\ Vy_i]$.

In mathematical view, the methods examined in the lecture (Horn-Schunk, Lucas-Kanade etc.) or dense optical flow computation [3] can be used. The resulting visuals will be given under the Work Done title.

Second step is to determine sink points, which are attractive regions of the flow in the given scene. In the same way, they may be perceived as goals desired, tendencies or exits. The representation will be obtained with calculating distance to those sinks, from any cell. To be more precise, they initialize a point on the cell grid image with non-zero velocity. Then, at every step it moves to next location with the influence of its neighhbors and try to reach an exit point. The velocity at each point is re-estimated. Once the weight of the neighbours and the velocity of the object are not enough to move to a new position, the current location becomes sink for that point. The distance between start point and its sink is assigned to that point. This procedure is repeated for every cell. Consequently, a 3-D model of static global motion of the crowd is obtained. The velocity in the next step and the next position are estimated as given in the following equations.

$$\widetilde{Z}_{i,1} = Z_i \ , \quad \widetilde{X}_{i,t+1} = \widetilde{X}_{i,t} + \widetilde{V}_{i,t}, \qquad (2)$$

$$\widetilde{V}_{i,t} = \frac{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} V_n W_{t,n}}{\sum_{n \in Neighbor(\widetilde{X}_{i,t})} W_{t,n}} \ , \ W_{t,n} = \exp\left(- \left\| \frac{\widetilde{V}_{t-1} - V_n}{h_{t-1}} \right\|^2 \right), \qquad (3)$$

(2) and (3) estimation of position and velocity in the next sink step

Next position is simply sum of the current position and velocity vector. But, the velocity estimation is more sophisticated; first, neighbor weights are calculated and the the influence of each velocity in the window is taken into account by averaging.

In the for every sink path, the distance is generated and assigned. The resulting shape is explained very well in the original paper [1]: *SFF translates into a force in the direction that requires the least number of steps to reach the nearest exit or sink. The shape of the SFF emphasizes the notion that if you place a particle at any location, it will roll down towards the exit. This is precisely what the goal oriented dynamics of the crowd in this scene represent.*

b) Boundary Floor Field (BFF): The next two FFs needs less computation. First is boundary related floor field, which represents natural compulsive barriers and walls. They may be physical or virtual, too. A virtual barrier would be created in the presence of crowds having different directions. Boundary field is also calculated with the use of N_B frames starting from time instance T_B.

They, in fact, implemented a segmentation algorithm proposed earlier by themselves [2]. In that algorithm, barriers are represented by ridgelines of FTLE fields. Then, they acquired a segmentation map and edge map. That would be shape of compulsive effects of walls.

On the other hand, in order to get a simple result first and due to timing issues, I decided use a easier method that gives the almost same result for the impelemented (marathon video) footage in terms of BFF. I simply used a threshold to segment movement from the pre-calculated optical field of ours. Of course, this will not have virtual barrier detection ability but it will let us get a working result sooner. Here below, their resulting BFF model is given taken from the paper [1]. This can be compared with our primitive result located in the Work Done part. BFF values are indicated by the distance to nearest barrier.

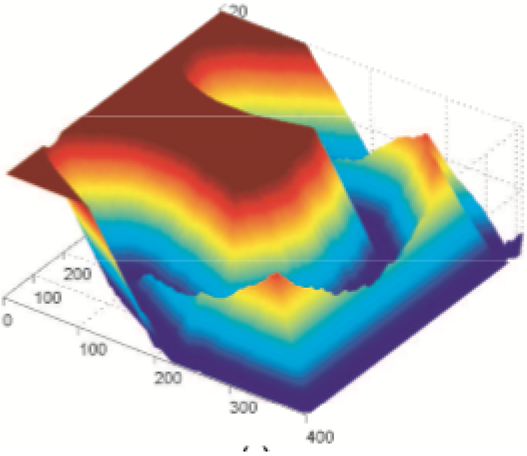

Figure 1: Boundary floor field calculated by their algorithm

c) Dynamic Floor Field (DFF): This final floor is representation of local crowd motion. The instantaneous information of the crowd around the tracked individual is an important cue for the next probable location. DFF, different than other FFs, is calculated in real time for each time step in a time window of N_D frames.

Again, optical flow came into action. Optical flow between consecutive frames are calculated and stacked. A grid of particles is initited and numerical advection [2] is implemented. In more detail, interaction value between i and j cells are incremented by one, whenever a particle moves from i to j. As a result, DFF has non-negative, dynamic values of neighbors representing local interactions for each cell.

1. LITERATURE

In literature, there given lots of tracking algorithms. But our scope is not only tracking, also tracking in dense object population. For all tracking approaches the survey of Yılmaz et al.[4] might be eye-opening. Please note that, there also exists contribution of one of our writers, M. Shah. That is rather long work examining object representations, feature selection methods and distinct object detection algortihm bases (e.g. Background subtraction, point detectors, segmentation).

It is noted that Zhao et al. [5] is one the pioneers proposing an algorithm regarding our cause. Their algorithm used articulated ellipsoids to model the human shape, color histograms to model appearance, and a Gaussian distribution to model the background for segmentation [1]. But it is pointed out that, this method is not perfectly suits our needs in high density crowds. Because, it is not always possible to fit a ellipsoid human body in the scene. Moving to probabilistic approaches, Brostow et al. [6], presented a framework for clustering points moving together, so they are likely to be the part of the same individual.

There also given other interesting algorithms given with short references and links. But none is directly resembles to floor field modeling.

Please note that, optical flow algorithm may be alternated to other types in order to satisfy our further needs. In SFF step, equation (3) is employment of a kernel based estimator. This is similar to mean shift approach, but appearance similarty is not used. There stated that other methods are exists that are not suitable for "distance to sink" objectives.

2. WORK DONE

In this section, step-wise implementation so far and results are given in order. Some chosen parameters are indicated, which are determined experimentally depending on the timing constraints and precision. To be analogous with the paper, I chose "maraton-3" video for development of the code. Its features are 720x404 resolution, 464 frames, 25 FPS and RGB(24-bit depth) format.

- First, start with reading video file into MATLAB.
- Find optical flow using first 20 frames. The optical flow estimation is done between consecutive grayscale frames. Now, we have 19 point flow maps.
- Take the average of the velocity vector of the optical flow map to get a resultant map (fig.2).

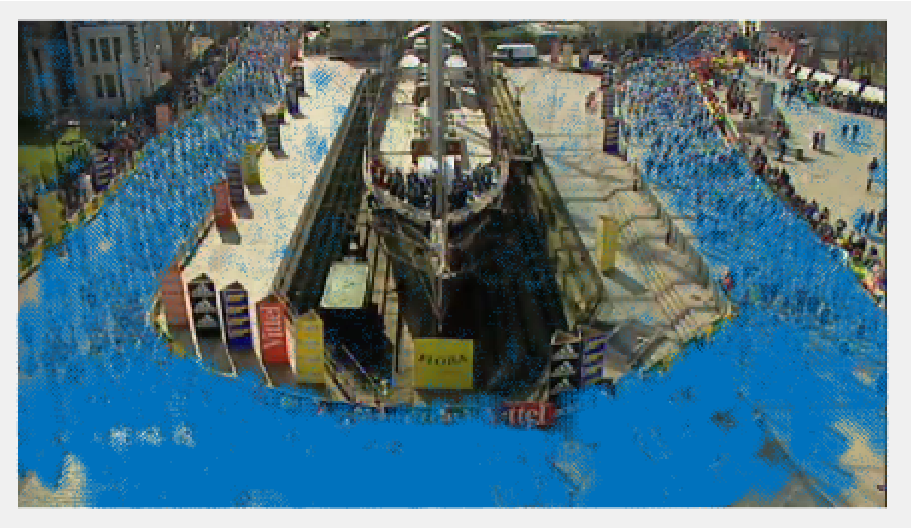

Figure 2: Point flow field (with blue velocity arrows)

- Move to sink-seeking process. Using Otsu's method, threshold norm of the velocity vectors (fig. 3). Also, clean up small regions.

Figure 3: Thresholded optical flow map

- Grid the optical flow field map with 10x10 blocks (figure 4). Take only center points.

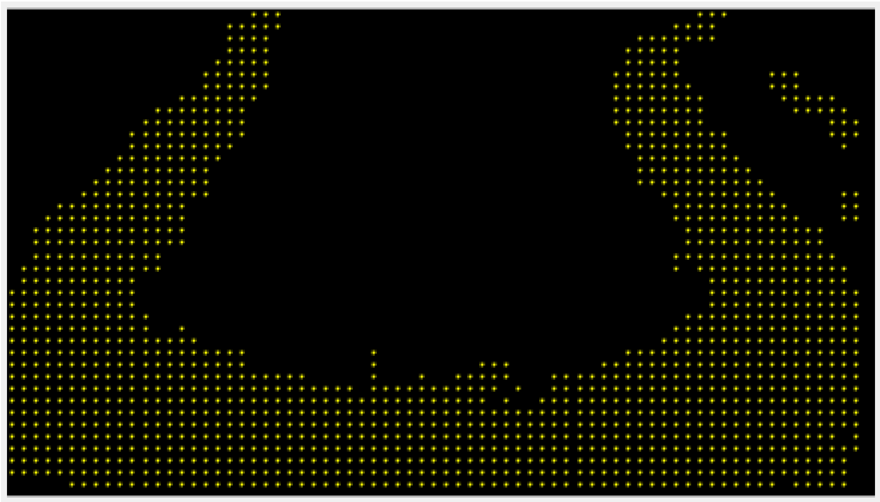

Figure 4: Grid of the optical flow

- Now, apply sink-seeking process only to the center points. Considering one point takes approximately 0.03 seconds to reach its sink, this grid and center point approach shortens the whole process into a reasonable time interval (~32 seconds). See Figure 5.

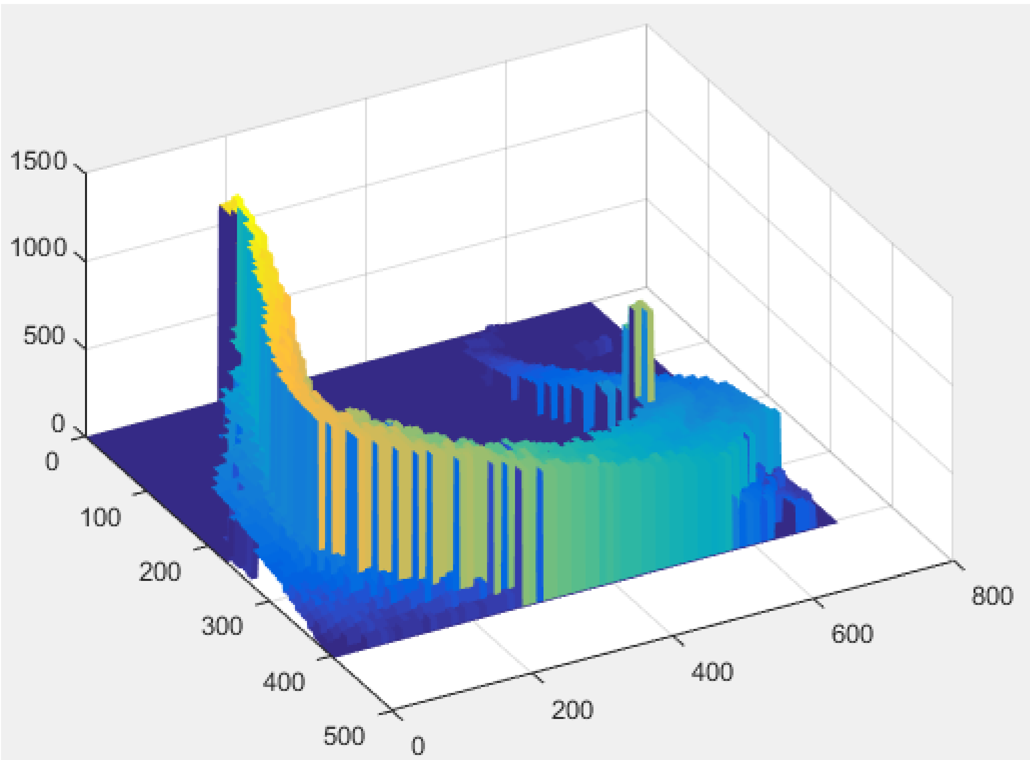

Figure 5: Static floor field model

- A moving window of 7x7 is chosen for velocity estimation in sink-seeking. Every sink path is started from those center points given in the figure 4. When exit point has reached, distance variable is assigned to corresponding grid in the SFF map. Reaching an exit point means fulfilment of one the following condition:
  - Last two positions are identical
  - There is a NaN valued velocity

- Sink steps exceeded limit of 500.

- For BFF extraction, again determine a threshold with Otsu's method on the flow map. Pre-process the binary image with morphological operation and clear small area.
- Then apply distance transform on the resulting image, given in the figure 6.

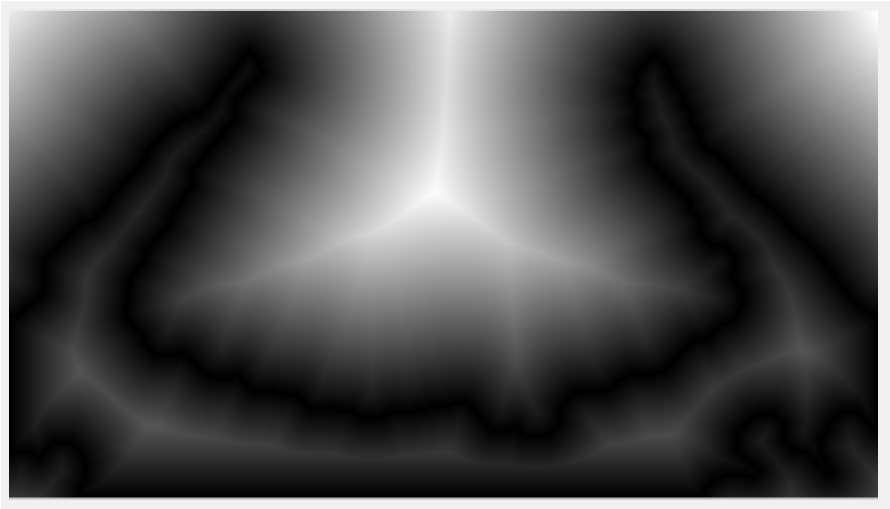

Figure 6: Distance transform on the thresholded flow map

- Sature BFF values which are greater than level of 50. The result is close enough to that of the method proposed by authors given in Figure 1. But, it lacks of virtual boundary detection ability.(fig. 7)

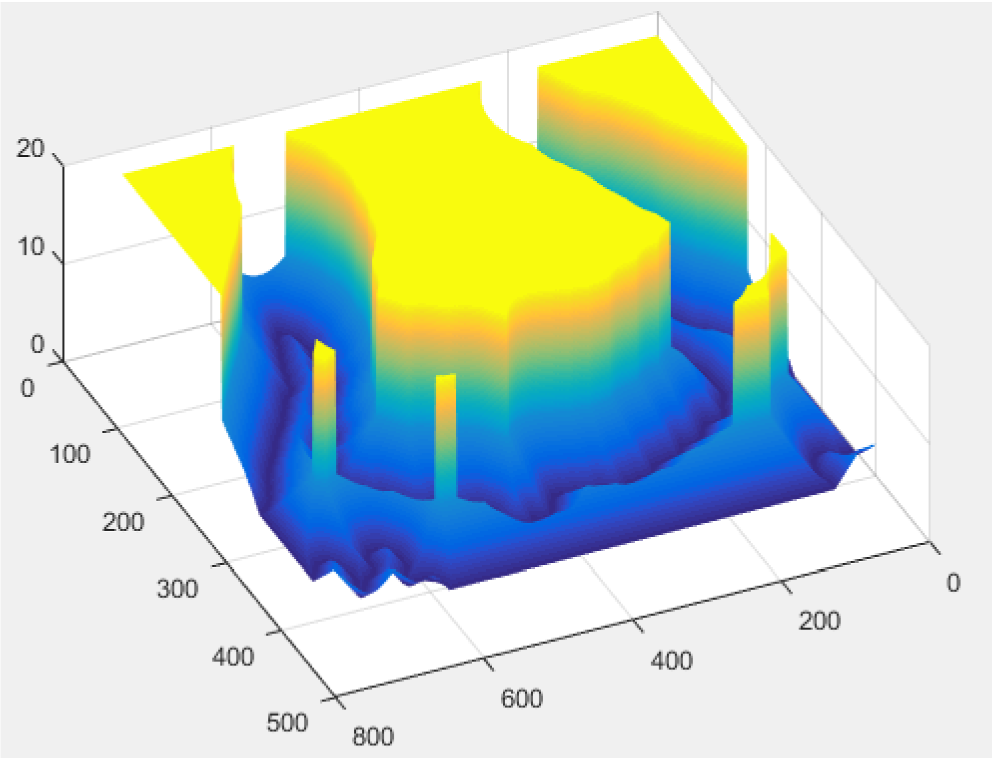

Figure 7: Boundary floor field model

3. FUTURE WORK

The future work includes finding DFF likewise SFF and BFF. Then a simple GUI is intended to be created that allows user to select a rectangular region of object. That will be static sized and tracked with the help of equation (1) probability matrix. In the end, different inputs and videos will be tested. Parameters, constants, window sizes or required frame counts of differente operation may change and explained with reasoning. After, reported evaluation and oral presentation will be prepared.

REFERENCES

[1] S. Ali and M. Shah, Floor Fields for Tracking in High Density Crowd Scenes, ECC, 2008.
[2] S. Ali and M. Shah, A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis, IEEE CVPR, 2007.
[3] R. Gurka et al., Computation of Pressure Distribution Using PIV Velocity Data, In Proc. of the 3rd International Workshop on Particle Image Velocimetry, 1999.
[4] A. Yilmaz et al., Object Tracking: A Survey, ACM Journal of Computing Surveys, Vol. 38, No. 4, 2006.
[5] T. Zhao and R. Nevatia, Tracking Multiple Humans in Crowded Environment, IEEE CVPR, 2004.
[6] G. Brostow and R. Cipolla, Unsupervised Bayesian Detection of Independent Motion in Crowds, IEEE CVPR, 2006.