Real-time Crowd Safety and Comfort Management From CCTV Images

Muhammad Baqui[a] and Rainald Löhner[a]

[a]Center for Computational Fluid Dynamics, College of Science, George Mason University, M.S. 6A2, Fairfax, VA 22030-4444, USA

ABSTRACT

High density pedestrian flows are a common ocurrence. Pedestrian safety and comfort in high density flows can present serious challenges to organizers, businesses and safety personnel. Obtaining pedestrian density and velocity directly from Closed Circuit Television (CCTV) would significantly improve real-time crowd management. A study of high density crowd monitoring from video and its real-time application viability is presented. The video data is captured from CCTV. Both cross correlation based and optical flow based approaches are studied. Results are presented in the form of fundamental diagrams, velocity vectors and speed contours of the flow field.

Keywords: High density crowd, pedestrian flow, pedestrian traffic management, particle image velocimetry, pedestrian traffic planning, crowd surveillance

1. INTRODUCTION

The rise of urbanization, increase in population, and comfortable means of transportation have resulted in events where a large number of people congregate. These can be seen in political demonstrations, national celebrations (e.g. independence day, presidential inaugurations), music concerts, sport events, large airports and train stations, as well as religious gatherings. For some of these gatherings, the number of participants can reach several hundred thousand, and even millions. Maintaining safety and comfort of large crowds in these circumstances can be extremely challenging. The current way of managing these events is based on on-site visual monitoring and not based on objective or computational methods incorporating pedestrian density or velocity. Visual monitoring is prone to error as it depends highly on personal judgement.

The accurate estimation of high density pedestrian flows has received considerable attention over last few years. The PED-series of Conferences[1 2] gives a good summary of the state of the art in this emerging field. Image processing and computer vision techniques can contribute a great deal in extracting instantaneous pedestrian flow state through examining Closed Circuit Television(CCTV) footage. Velocity and density are two fundamental properties of high density pedestrian flows. Real-time evaluation of these properties would contribute a great deal to the safety and comfort management of people in these events. Image based velocity extraction from moving objects has been an active area of research over the last decade. The optical flow technique[3 4] has been one of the most popular algorithms for velocity extraction from images with moving objects. A number of studies have adopted the optical flow method for velocity extraction from high density crowd flow[5 6 7 8] .

In experimental fluid dynamics, the Particle Image Velocimetry (PIV) technique [9 10 11 12] is used to extract velocities of fluid particles from videos that are captured by a high speed camera. At the heart of the process is the statistical pattern matching through cross correlation. The technique is fast and possesses a good potential in velocity extraction from crowd images. In this work, a comparative study is performed between the optical flow and PIV for crowds image processing.

Two issues need to be addressed in image based crowd management. First, the processing time and second, the correction of perspective. For real-time monitoring of crowds using CCTV, the image processing needs to be capable of producing results in real-time. Otherwise, it will not be able to offer any benefit to the operations

Further author information: (Send correspondence to Muhammad Baqui)
Muhammad Baqui: E-mail: mbaqui@gmu.edu,
Rainald Löhner: E-mail: rlohner@gmu.edu, Telephone:703-993-4075

Real-Time Image and Video Processing 2017, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE Vol. 10223, 1022304 · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2262319

/ security personnel. On the other hand, velocities obtained using image processing will come as 2D pixel displacements. These need to be projected into 3D world coordinates for the elimination of perspective and conversion into real world units. Estimating accurate pedestrian position and velocity would also contribute greatly to the benchmarking and improvement of pedestrian motion models, e.g. by being able to incorporate more precise initial conditions. Recently, the microscopic pedestrian models have been able to simulate in excess of a million pedestrians in real-time[13].[14] Accurate initial conditions using CCTV image processing would greatly benefit these models in managing large public events and in designing proper evacuation plans.

2. MATERIALS AND METHODS

In this section the velocity extraction techniques and perspective correction scheme is discussed. In section 2.1.1 the cross correlation and in section 2.1.2 the optical flow is presented. The perspective correction scheme is described in section 2.2.

2.1 Velocity Extraction

2.1.1 Cross Correlation

Statistical cross correlation is used in Particle Image Velocimetry (PIV) for velocity extraction. Cross correlation computes component-wise the inner product between two input signals (images) and produces maximum similarity by a single peak at the center. In equation 1, A and B are two input signals (images) upon which cross correlation is performed. A and B are identical except they are separated by a small time gap. From the expanded equation 1, the third term 2AB (inner product) is responsible for the correlation. The correlation obtained in this fashion is the so-called Direct Cross Correlation(DCC). The dimension of the inner product (2AB) is $m_x \times m_y$, where the input image dimension is nxn. The time complexity of the process is O(n^2 m^2). One can realize that it is not efficient for processing crowd images where image dimension is generally in the order of thousands.

To improve this processing time convolution based cross correlation is generally applied. The convolution based correlation is shown in equation 3. It can be seen that it is identical to the DCC equation 1, except now the direction of B is opposite. So if B is initially flipped and then the convolution is done, the results will be identical to the DCC equation. The time complexity of one Fourier transform is $O(m \log m)$. Hence, the entire correlation can now be obtained in $O(n^2 \log m)$ operations. The entire process is explained in the pseudo code shown in algorim 2.1.1.

$$\forall m_x, m_y \quad |A-B|^2 = A^2 + B^2 - 2AB \tag{1}$$

$$C(i,j) = \sum_{i}^{m_x} \sum_{j}^{m_y} A(i,j)B(i-m_x, j-m_y) \tag{2}$$

$$C(i,j) = IFFT(FFT(A(i,j)) * FFT(B(m_x, m_y))) \tag{3}$$

$$= \sum_{i}^{m_x} \sum_{j}^{m_y} A(i,j)B(m_x - i, m_y - j) \tag{4}$$

In Figure 1, two sample images and their cross correlation result are shown. To obtain any cross correlation, the input image needs to be divided into sub images called interrogation spots. In Figure 1, the spot size considered was 60x60 pixels. The highest peak of the correlation surface can also be seen in the image. A Gaussian function is fitted between this peak location with its neighbors in x and y in order to obtain the displacement within subpixel accuracy. From this displacement and time difference of the frames, the velocity can be obtained for each interrogation spot.

Algorithm 1: Cross correlation velocity

Input: Image A and B, spot size m_x,m_y, overlap
Algorithm:
1. Load images A and B.
2. **for** *window loop imsizex* **do**
   for *window to imsizey* **do**
      Extract window from image A;
      Extract window from image B;
      Flip image window B upside down;
      a1=FFT(image window A);
      b1=FFT(image window B);
      c1=a1*b1;
      c=IFFT(c1);
      for *indices in c* **do**
         peakx=fit gaussian between i,i+1,i-1;
         peaky=fit gaussian between j,j+1,j-1;
         u=(window index-peakx)*scale;
         v=(window index-peaky)*scale;
      end
   end
end

(a) (b) (c)

Figure 1: PIV correlation at spot size 60X60 (a) First image (b) Second image (c) correlation surface. The single peak at the center of the correlation surface indicates good correlation.

2.1.2 Optical Flow

In optical flow the image captured by a camera is thought as a spatio-temporal function of gray level values represented as $I(x, y, t)$. Optical flow theory makes the assumption that the change in this gray level values in a short interval of time is negligible i.e. it does not change. This is called the brightness constancy assumption that leads to the following equation 5

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{5}$$

here $\mathbf{u} = (\mathbf{u}, \mathbf{v})$ is the optical flow vector of pixel location x, y between t and $t + 1$. The equation can be linearized by Taylor Series expansion

$$I(x,y,t) = I(x,y,t+1) + \nabla I(x,y,t+1)^T \begin{bmatrix} u \\ v \end{bmatrix} \tag{6}$$

$$0 = I(x,y,t+1) - I(x,y,t) + \nabla I(x,y,t+1) \begin{bmatrix} u \\ v \end{bmatrix} \tag{7}$$

$$\tag{8}$$

The partial derivatives in equation 6 can be represented as I_t,I_x,I_y which leads to the optical flow constraint equation as follows

$$OFC(u,v) : 0 = I_t + I_x u + I_y v \tag{9}$$

The optical flow constraint (OFC) equation has two unknowns and one equation. Among many techniques developed to solve this problem, the solutions provided by Horn et al. and Lucas et al. are the most prominent. Horn et al.[3] constructs a variational formulation to solve the system. Lucas et al. citeLucas81 on the other hand takes an alternative approach where an overdetermined minimization problem is solved following the least squared approach. In particular, Lucas et al.[4] solved 10 in the least squared approach with the assumption that the flow is constant along the neighborhood of individual pixels. These pixel neighborhoods typically contain nxn pixels where n is smaller than 15.

$$\underset{u,v}{min} \left\{ \sum_{x,y \in N} \left(I_t(\mathbf{x}') + I_x(\mathbf{x}')u + I_y(\mathbf{x}')v \right)^2 \right\} \tag{10}$$

To get the velocities within sub pixel accuracy, it is common to use image pyramids. The pyramids solve the low resolution images first then gradually go into higher pyramid levels. The steps of getting velocities through optical flow has been summarized in the following algorithm 2.1.2.

Algorithm 2: Optical flow velocity

```
Input: Image A and B, window,level,nloop
Algorithm: for each image do
    Create image pyramid based on level;
    [I0,I1,..]=pyramid(I0,I1,window,level)
end
for each level of image do
    Ia=load image A for current scale;
    Ib=load image B for current scale;
    compute x derivative of Ia;
    compute x derivative of Ib;
    compute y derivative of Ia;
    compute y derivative of Ib;
    compute time derivative between Ia,Ib;
    while i¡nloop do
        obtain u,v iteratively solving optical flow equation;
    end
end
[u,v]=merge(u,v,level)
```

2.2 Perspective Correction

The displacements obtained from cross correlation come in 2D pixel displacements. With the help of projective geometry and a pinhole camera model, they can be converted to real world units (meters). The methodology can be explained with the help of the ray diagram shown in Figure 2. There are three different coordinate systems involved in the process: the 2D image coordinate, the 3D camera coordinate, and the 3D world coordinate.

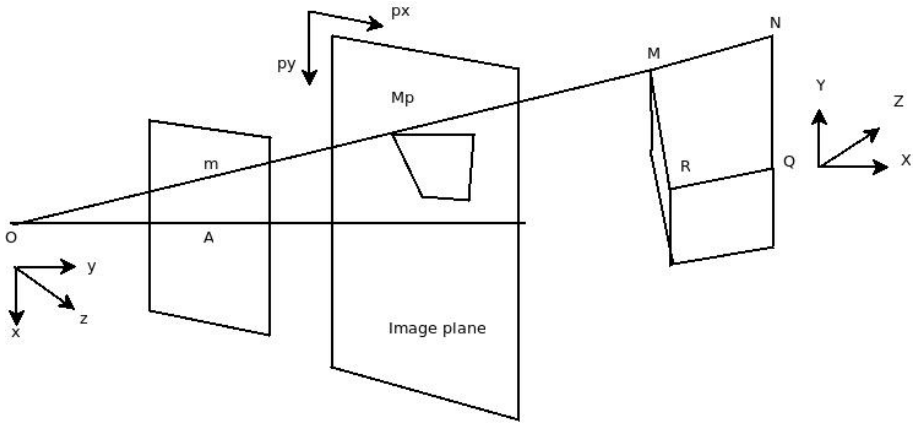


Figure 2: Camera projection from image coordinate to world coordinate.

An object MNQR creates a plane in the world coordinate system (X,Y,Z). A light ray from M in the world coordinate is mapped to Mp in the image coordinate. When extended further, the light ray eventually ends at the optical center, denoted by O in Figure 2. Now, in the camera coordinate system, m is the intersection point of plane MRQ and the line om. In order to project the point m in the world coordinate system the following transformation is needed [Equation 11]:

$$m_{world} = C * m \tag{11}$$

In Equation 11, C is the camera matrix that represents the location of the camera in the world coordinate system. C is expressed with a translation vector and a rotation matrix. In this work, the camera location in the world coordinate determined by the P3P algorithm.[15]

After the matrix C is determined, the points MNQ can be projected to the camera coordinate by the following transformation [Equation 12]:

$$Pc = inv(C) * P, \quad where \quad P = [M, N, Q] \tag{12}$$

An equation of a plane can be constructed from Pc in the camera coordinate system. Let ap be the final form of this plane. This plane will consist of coefficients n and intercept d.

$$ap = [n(1), n(2), n(3), d], \quad where \quad n = [n(1), n(2), n(3)] \tag{13}$$

The next step is to project the pixel points from the image to the camera coordinate system. Let px and py be the x and y pixel coordinates of an image. It can be projected to the camera coordinate by the principal point of the image and focal point of the camera [Equation 14]. The principal point and the focal point are known from the camera manufacturer information. Let, b,c and r correspond to the values after projecting the px, py to the camera coordinate system [Equation 14].

$$\begin{aligned} Pimg &= [px, py] \\ Pimg_c &= [px - principal_x/fx, py - principal_y/fy, 1] \\ Pimg_c &= [b, c, r] \end{aligned} \tag{14}$$

A parametric equation of a line in the camera coordinate system with points $Pimg_c$ [Equation 14] and optical center O (0,0,0) can be formed in the following way [Equation 15]:

$$\begin{aligned} xi = t * b, yi = t * c, zi = t \\ t = d/(n(1) * b + n(2) * c + n(3) * r) \end{aligned} \tag{15}$$

where t is the parameter and xi, yi, zi is the intersection point of the line $OPimg_c$ that goes through the optical center. From equation (12) and (13) the value of t may be obtained. Given t, the values of xi, yi, zi can be determined, i.e. the world coordinate locations of image pixels px,py.

The methodology explained above is used correct the perspective of images. In Figure 3, the points used for this conversion are shown. The results and accuracy of the projection are presented in section 3.2.

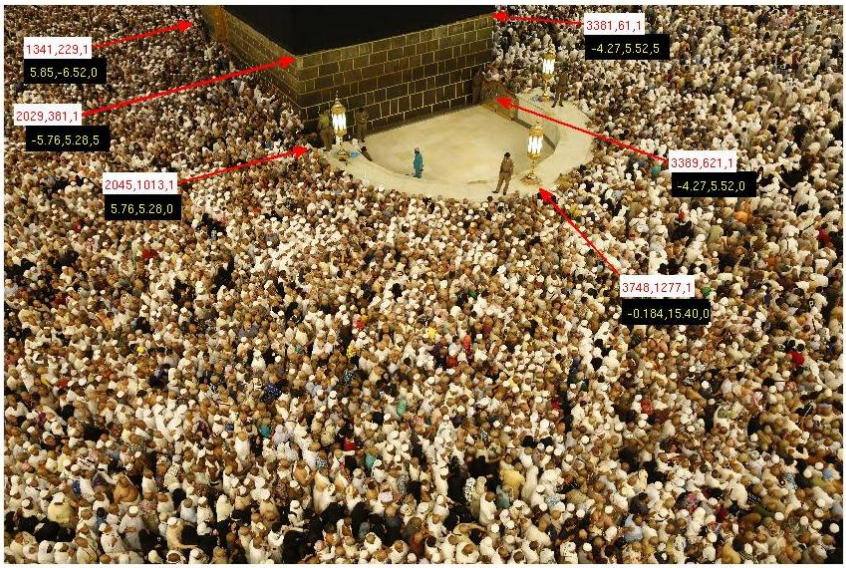


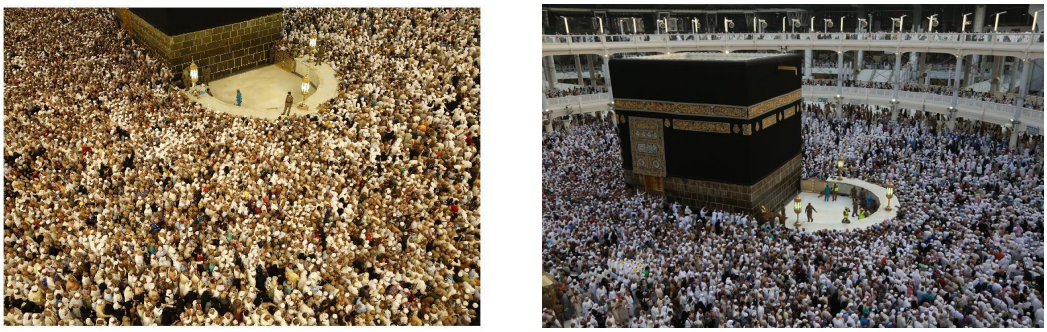
Figure 3: The landmark points with known image and world coordinate locations. These points are used for perspective correction.

2.3 Dataset Description

The overall aim of the present work is to study real-time crowd management issues for a high density religious event. Every year about 2 million Muslim pilgrims gather at Mecca, Saudi Arabia, to perform the so-called Hajj. It has been reported that people density gets very high in the event and sometimes create crowd quake situations[16] .[17] To study the event carefully, several sets of time lapse photos were made available to the authors. These photos are collected from the CCTV cameras located at the Grand Mosque of Mecca. In Figure 4 a single frame from the two sets of photos analyzed are shown. The first set is an example of a high density situation whereas the second set corresponds to a moderate density situation. The resolutions of both the images was 5760x3840 pixels.

2.3.1 Head Counting

To construct the fundamental diagram, the number of people residing in a per unit area is needed. For this purpose, the total number of persons in the image were manually annotated. Each individual image were divided into 100 smaller images to facilitate this annotation. A total of 600 image patches were manually counted for the purpose. In Figure 5 a sample block with annotated pedestrians can be seen.



(a) (b)

Figure 4: A sample frame of the dataset (a) first set (b) second set

Figure 5: Sample image with ground truth annotations

3. EXPERIMENTS AND RESULTS

3.1 Velocity Estimation Results

The velocity vectors from PIV analysis are shown in Figure 6 for both sets of test images. The interrogation spot is taken as 150x150 pixels. The sizes are chosen on the basis of real-time performance and quantitative accuracy. Making the spot size bigger would result in faster processing time. However, results will not have good accuracy.

From Figure 6 the movement pattern of the crowd can be seen. The rotational motion around the black stone is well captured by the PIV algorithm. For comparison, the velocity vectors of PIV and optical flow can be seen side by side in Figure 7. It can be seen that vectors obtained from PIV and optical flow are qualitatively identical. The quantitative accuracy of PIV in resolving velocity can be found in Table 2.

Although the bulk motion is identical between optical flow and PIV, the optical flow has shown better performance in resolving local motions from the image. In Figure 8, one can see that the optical flow has a better decomposition of local vectors from the image patches.

An important feature of velocity extraction from images is their performance in detecting nonmoving objects such as the wall, standing pedestrians etc. Both PIV and optical flow have similar performance in these cases. In Figure 9 the vectors can be seen only for moving pedestrians. No vectors indicating movement are found along the walls.

Results from previous experiments reveal a slightly better performance of optical flow to PIV in velocity extraction. The reason for this is that the optical flow resolves vectors for each individual pixels whereas PIV resolves vectors based on spot window size. For each spot only one vector is obtained in PIV. As optical flow

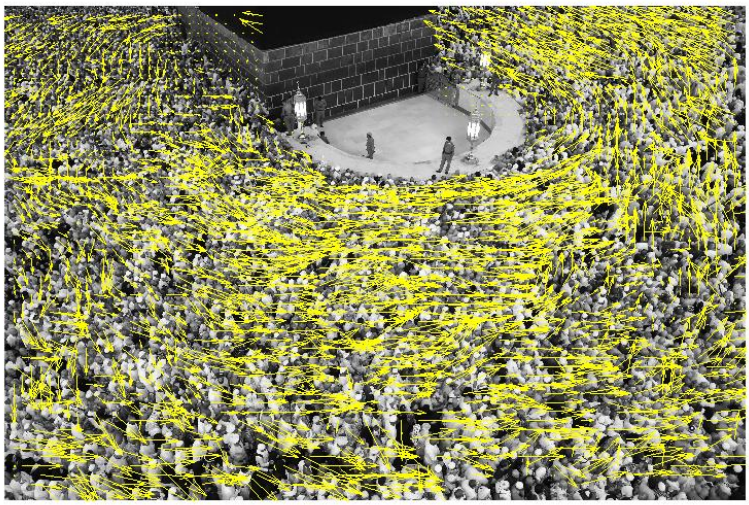


Figure 6: Input image and vectors from PIV

(a) (b)

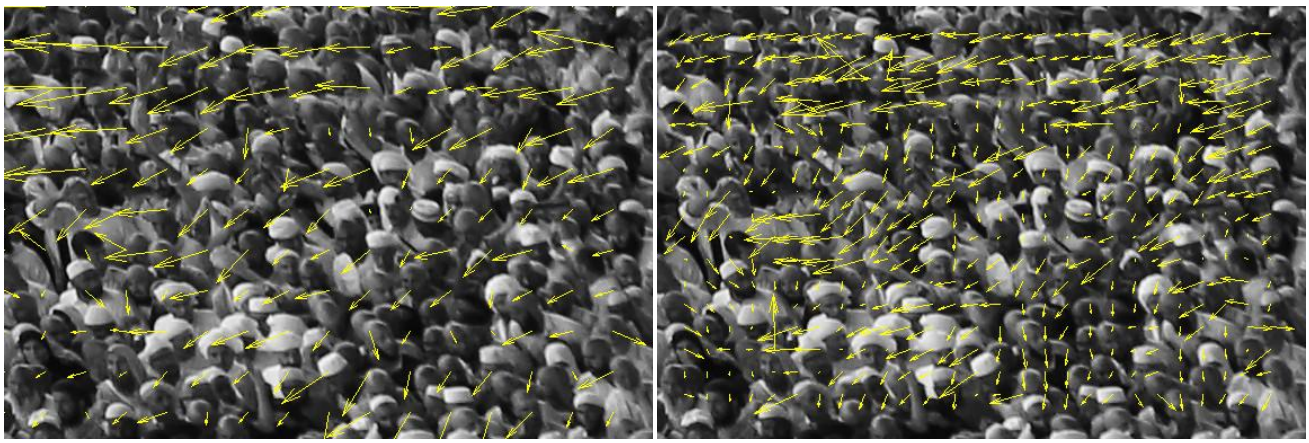
Figure 7: Velocity vectors from (a). Optical Flow (b) PIV

determines vectors for each pixel, it takes a significantly higher amount of time in producing these vectors. This makes optical flow less suitable for real-time processing. PIV, on the other hand, incorporates one vector per spot size. It resolves velocities per spot on an average sense. Depending on the need of the facility management, the spot size can be made bigger or smaller with the compromise in processing time. Table 1 shows the comparative processing times of optical flow and PIV.

Image Size (pixel)	Optical Flow (sec)	PIV (sec)
576x384	51.46	0.0469
288x192	20.26	0.0286
5760x3840	-	2.92

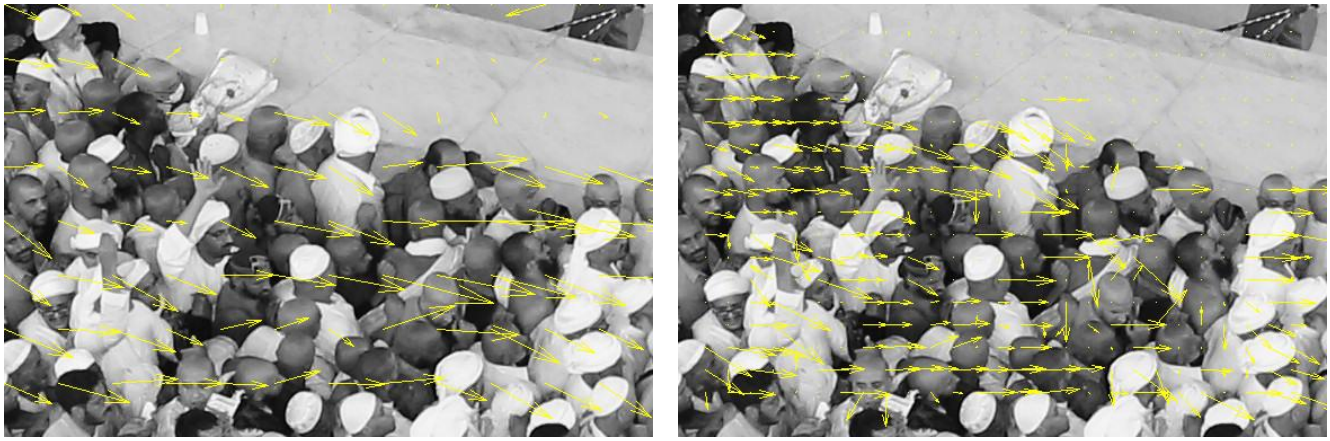
Table 1: Comparison of PIV timing with optical flow

The experiments on timings reported in Table 1 were performed in an Intel Core i5 processor with 8GB main



(a) (b)

Figure 8: Local velocity vectors from (a). Optical Flow (b) PIV



(a) (b)

Figure 9: Velocity vectors near wall region (a). Optical Flow (b) PIV

memory. The computer program for PIV written by the author in C++ and for optical flow, the implementation of the open source library OpenCV[18,19] is used. For the PIV implementation no parallelization or performance optimization was performed. From Table 1 the superior performance of PIV is clearly evident. It can be seen that optical flow has not been able to resolve velocities from the full CCTV image. This is because the optical flow is currently beyond real-time.[20] To make the comparison complete, lower dimensional image segments are chosen for processing. The authors would like to note that with better compute power and GPU parallelization, the running time of optical flow can be improved markedly. However, one has to realize that with more compute power, the timing of PIV can also be improved further. Also, with more compute power, a higher number of images from many cameras can be processed with a GPU based PIV implementation. Such an architecture would result in a greater coverage of safety for events with large crowds.

The pixel displacements obtained from PIV have been studied for its quantitative accuracy. For this purpose, six locations from an image are randomly selected. For these locations, the ground truth pedestrian movements are measured by manual tracking of pedestrians between the frames. The results from PIV and ground truth evaluation are shown in Table 2.

The accuracy of PIV for these six locations can be seen in the fifth column of the table. It can be seen that the maximum difference in pixel displacement is close to 15%. However, these and all other vectors in the image are corrected for perspective and converted from pixel displacements to meters following the approach outlined in section 2.2. The average velocity of the flow calculated after the converstion is 0.44 m/s. Other studies of

Actual dx (pixel)	PIV dx (pixel)	Actual dy (pixel)	PIV dy (pixel)	%error
-3	-2.52	3	2.73	12.43
-3	-3.89	5	4.08	3.32
-8	-6.87	3	2.39	14.86
-5	-4.28	4	4.14	7.08
4	3.08	-2	-2.14	16.13
8	7.55	2	1.72	6.09

Table 2: Comparison of PIV displacement with ground truth displacement values

velocities in the same location conform out findings,[16].[21]

3.2 Perspective Correction Results

The pedestrians are placed in the world coordinates from image coordinate following the approach outlined in section 2.2. The images were captured from CannonEOS 5D Mark III of focal length of 44 mm and sensor size of 35x24 mm. The points considered for forming equation of plane in camera coordinate, can be seen in Figure 3. Once the plane is formed, the individual image pixels and displacements are projected to the camera coordinate system using focal lenght and principal point. The intersection of these pixels with this plane will give the location of these pixels in camera coordinates. Each pixel will have its corresponding intersection point. Once this intersection point is known, then camera extrinsic parameters can be used to project this point back into the world coordinate system. In the following Table 3 the accuracy of this transformation process is outlined for four random locations. It can be seen that the maximum amount of error incorporated during the projection corresponds the approximately the width of 3.54 persons. In determining the error, it is seen how many people can be accommodated in the difference of projected and actual points. The maximum error of 3.54 indicates, at a maximum there around 4 persons that are wrongly projected in the 3D world coordinate. The width of a person is considered as 0.5m, a commonly accepted value (see, e.g. Predechenskii et al.[22]).

World x (m)	World y (m)	World z (m)	Projected x (m)	Projected y (m)	Projected z (m)	Error (person width)
1.77	15.35	0.0	2.54	14.65	0.0008	2.08
-1.84	15.40	0.0	-0.183	15.39	0.0008	2.02
5.85	-6.52	0.0	4.09	-6.31	0.0008	3.54
-4.27	5.52	0.0	-4.87	5.18	0.0008	1.37

Table 3: Comparison of projected location vs actual location.

3.3 The Fundamental Diagram

The fundamental diagram provides a snapshot of the flow at a particular time. It is a plot of density of a region or area vs. speed of pedestrians. By constructing the fundamental diagram, the condition of the pedestrian flow can be evaluated at a given time. In Figure 10, fundamental diagram is presented for both sets of test images. In the figure the triangle dots represent the first set and circle dots represent the second test set. The velocities are transformed to 3D world coordinate after being obtained from PIV. The fundamental diagrams of Predechenskii et al.,[22] Helbing et al.[23] and Wiedmann[24] are also shown in the figure as PM, Helb and WD. For other empirical work of fundamental diagrams for various crowd densities see Seyfried et al.[25] and Chattaraj et al.[26] It should be noted that the first set of images are taken in high density while the second set is taken in low to moderate density. The sporadic nature of crowd flow is clearly evident in the density-velocity relation. One can see in low density the spread of velocities is high. As the density increases, the fluctuation in velocities gets reduced. The same nature of flow is reported by Predechenskii et al.[22] The shape of the fundamental diagram provides another proof that the velocities obtained by PIV are realistic.

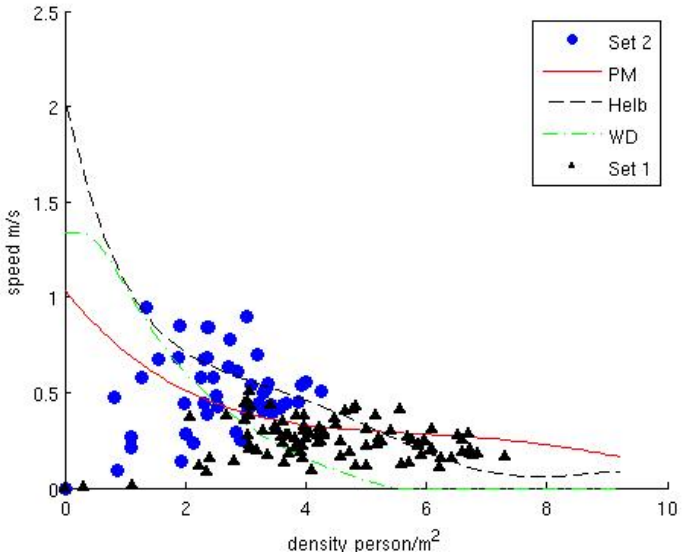


Figure 10: Fundamental diagram of first (blue circles) and second set (black triangles). The red line is the Predechenskii curve (PM), the green dotted line is the Wiedmann curve (WD) and black dotted line is the Helbing curve (Helb).

The density of pedestrian reported in the previous fundamental diagram is obtained through manual annotation of pedestrians from test images. When the density and velocity relation of crowds for a given event is determined once, that can be used as a reference for future calculations. The Predechenskii et al.[22] curve can be used as well. The velocities from PIV can be used along with Predechenskii et al.[22] curve to estimate corresponding densities.

An attractive way to construct the fundamental diagram would be from a predictive estimation of pedestrian density. Image processing and computer vision techniques have been applied in order to obtain the density from images[27].[28] In this work, a machine learning model in the form of boosted ferns regressor,[29],[3031] is constructed for obtaining pedestrian density. The model is trained with crowd images with pedestrian counts obtained from annotations. The model showed promising results in obtaining the predictive counts as presented in the following Table 4:

Test set number	Image number	Predictive count	Ground Truth	%error
1	5	2949	2910	1.34
1	6	3024	3263	7.32
1	8	2981	2816	5.58
2	4	4449	4451	0.0004
2	10	4564	4686	2.60
2	20	4423	4729	0.6

Table 4: Predictive pedestrian counts vs Ground truth counts for a number of images

3.4 Local Flow Properties

Along with the fundamental diagram, the density and the speed contours are also capable of providing spatial properties of the flow. In Figure 11 the density distribution of the crowd can be seen. For the convenience of annotations, the entire picture is divided into 100 smaller cells. It can be seen that in certain cells the density exceeds 8 persons/sq.m area indicating the presence of a very high density pockets. Other studies of the event have also found similar pockets of high density,[16].[21]

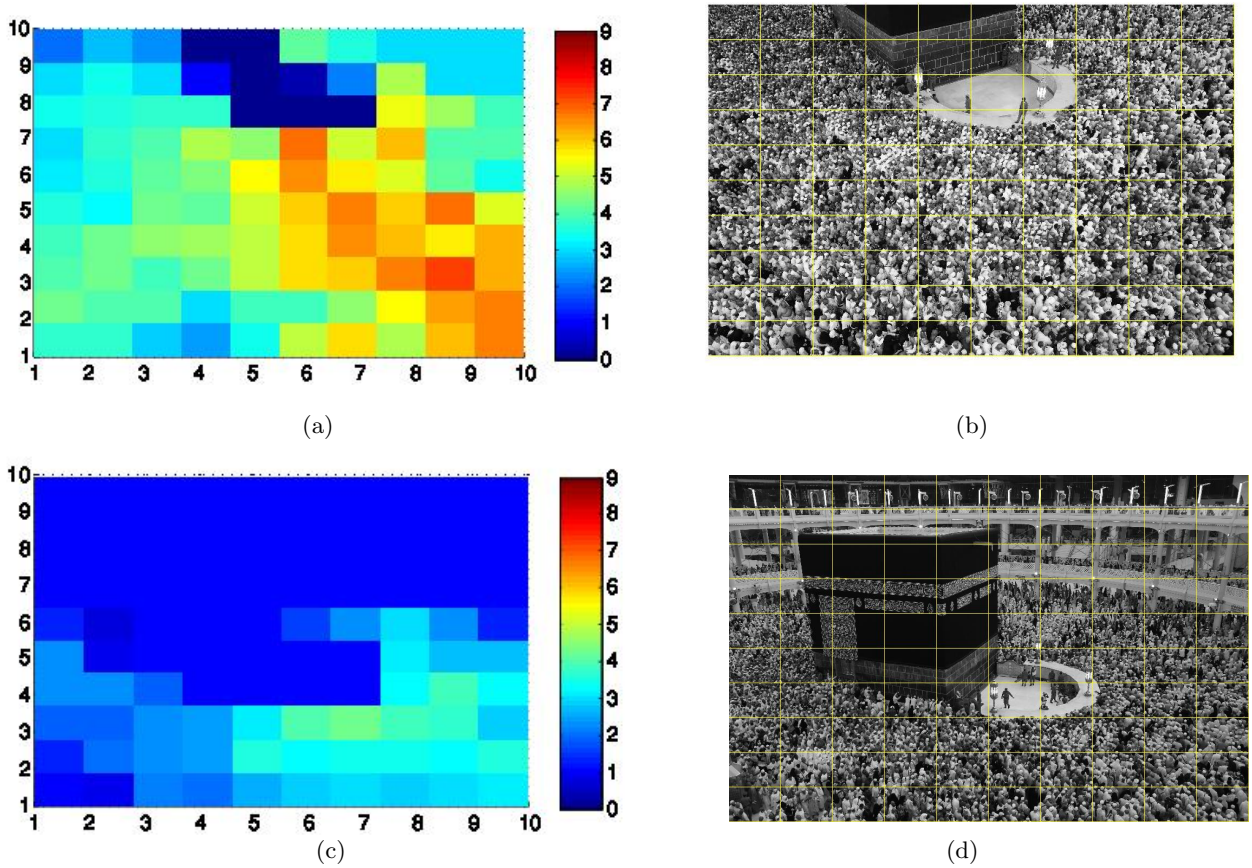


Figure 11: Block wise density(person/sq.m) (a) arrangements of cells (b) for first set. Density (c) arrangement of cells (d) of second set.

The speed contour consisting local distribution of speed of the image can be seen in Figure 12. Some areas can be seen to have high velocities. Most of the flow is seen uniform. In the case of a potential congestion, some areas along the domain would indicate a standing flow i.e. high density and low velocity. It can be seen from both sets that areas where density is higher [Figure 11], the velocities in those cells [Figure 12] are lower.

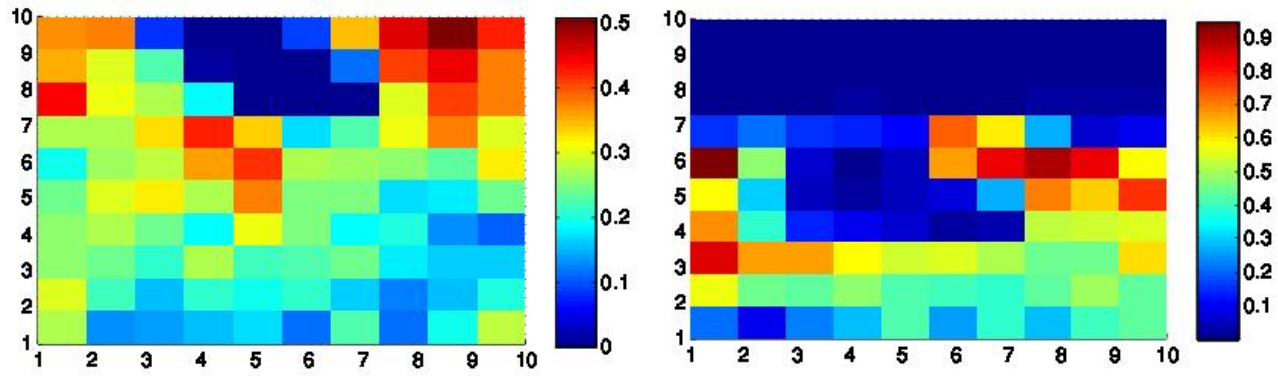


Figure 12: Block wise velocity distribution first set (a) second set(b).

4. CONCLUSIONS AND OUTLOOK

A study has been performed on real-time velocity extraction from crowd image processing. Two prime methods: optical flow and cross correlation technique were compared based on their accuracy and processing time. The optical flow showed slightly higher accuracy than cross correlation. The cross correlation on the other hand is significantly faster than optical flow, allowing for real time monitoring. The pixel displacements obtained from image processing are transformed to the world coordinate system through a pinhole camera based perspective correction scheme. The velocities of the crowd can be used to obtain local flow properties through speed contours. The real-time velocity extraction would benefit greatly in the construction of a fully automated onsite crowds monitoring system.

REFERENCES

[1] Daamen, W., Duives, D., and Hoogendoorn, S., "The conference in pedestrian and evacuation dynamics (ped 2014)," *Transportation Research Procedia* **2**, 1–818 (2014).

[2] Song, W., Ma, J., and Fu, L., "The conference in pedestrian and evacuation dynamics (ped 2016)," *University of Science and Technology of China Press* **1**, 1–618 (2016).

[3] Horn, B. K. and Schunck, B. G., "Determining optical flow," *Artificial intelligence* **17**(1-3), 185–203 (1981).

[4] Lucas, B. D., Kanade, T., et al., "An iterative image registration technique with an application to stereo vision," (1981).

[5] Dehghan, A., Idrees, H., Zamir, A. R., and Shah, M., "Automatic detection and tracking of pedestrians in videos with various crowd densities," in [*Pedestrian and Evacuation Dynamics 2012*], 3–19, Springer (2014).

[6] Rodriguez, M., Sivic, J., Laptev, I., and Audibert, J.-Y., "Data-driven crowd analysis in videos," in [*Computer vision (ICCV), 2011 IEEE international conference on*], 1235–1242, IEEE (2011).

[7] Ali, S. and Shah, M., "Floor fields for tracking in high density crowd scenes," in [*European conference on computer vision*], 1–14, Springer (2008).

[8] Mehran, R., Oyama, A., and Shah, M., "Abnormal crowd behavior detection using social force model," in [*Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*], 935–942, IEEE (2009).

[9] Willert, C. and Gharib, M., "Digital particle image velocimetry," *Experiments in Fluids* **10**(4), 181–193 (1991).

[10] Adrian, R. and Yao, C., "Development of pulsed laser velocimetry (plv) for measurement of turbulent flow," in [*In Symposium on Turbulence. X.B. Reed Jr., G.K. Patterson ed*], 170–184 (1984).

[11] Keane, R. and Adrian, R., "Optimization of particle image velocimeters .1. double pulsed systems," *Measurement Science and Technology* **11**(1), 1202–1215 (1990).

[12] Adrian, R., "Particle-imaging techniques for experimental fluid-mechanics," *Annual Review of Fluid Mechanics* **23**, 260–304 (1991).

[13] Lohner, R., Baqui, M., Haug, E., and Muhamad, B., "Real-time micro-modelling of a million pedestrians," *Engineering Computations* **33**(1), 217–237 (2016).

[14] Löhner, R., Haug, E., Zinggerling, C., and Onate, E., "Real-time micro-modeling of city evacuations," in [*Proc. Pedestrian and Evacuation Dynamics*], 500–504 (2016).

[15] Kneip, L., Scaramuzza, D., and Siegwart, R., "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in [*Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*], 2969–2976, IEEE (2011).

[16] Dambalmath, P., Muhammad, B., Haug, E., and Löhner, R., "Fundamental diagrams for specific very high density crowds," in [*Proc. Pedestrian and Evacuation Dynamics*], 6–11 (2016).

[17] Jiayue, W., Wenguo, W., and Xiaole, Z., "Comparison of turbulent pedestrian behaviors between mina and love parade," *Procedia Engineering* **84**, 708–714 (2014).

[18] Itseez, "Open source computer vision library." https://github.com/itseez/opencv (2015).

[19] Itseez, *The OpenCV Reference Manual*, 2.4.9.0 ed. (April 2014).

[20] Wedel, A. and Cremers, D., [*Stereo scene flow for 3D motion analysis*], Springer Science & Business Media (2011).
[21] Dridi, M., "Tracking individual targets in high density crowd scenes analysis of a video recording in hajj 2009," *Current Urban Studis* **3**, 35–53 (2015).
[22] Predtechenskii, V. and Milinskii, A., [*Planning for foot traffic flow in buildings*], National Bureau of Standards, US Department of Commerce, and the National Science Foundation, Washington, DC (1978).
[23] Helbing, D., Johansson, A., and Al-Abideen, H. Z., "Dynamics of crowd disasters: An empirical study," *Physical review E* **75**(4), 046109 (2007).
[24] Weidmann, U., "Transporttechnik der fussgänger," (1992).
[25] Seyfried, A., Steffen, B., Klingsch, W., and Boltes, M., "The fundamental diagram of pedestrian movement revisited," *Journal of Statistical Mechanics: Theory and Experiment* **2005**(10), P10002 (2005).
[26] Chattaraj, U., Seyfried, A., and Chakroborty, P., "Comparison of pedestrian fundamental diagram across cultures," *Advances in complex systems* **12**(03), 393–405 (2009).
[27] Johansson, A., Batty, M., Hayashi, K., Al Bar, O., Marcozzi, D., and Memish, Z. A., "Crowd and environmental management during mass gatherings," *The Lancet infectious diseases* **12**(2), 150–156 (2012).
[28] Johansson, A., Helbing, D., Al-Abideen, H. Z., and Al-Bosta, S., "From crowd dynamics to crowd safety: a video-based analysis," *Advances in Complex Systems* **11**(04), 497–527 (2008).
[29] Friedman, J. H., "Greedy function approximation: a gradient boosting machine," *Annals of statistics* , 1189–1232 (2001).
[30] Dollár, P., "Piotrs image and video matlab toolbox (pmt),https://pdollar.github.io/toolbox/," (2013).
[31] Elith, J., Leathwick, J. R., and Hastie, T., "A working guide to boosted regression trees," *Journal of Animal Ecology* **77**(4), 802–813 (2008).