Estimating the Real Positions of Objects in Images by Using Evolutionary Algorithm

Zhixing Huang
School of Computer Science and Engineering
South China University of Technology
Guangzhou, China

Weili Liu
Sun Yat-Sen University
Guangzhou, China

Jinghui Zhong (Corresponding author)
School of Computer Science and Engineering
South China University of Technology
Guangzhou, China
jinghuizhong@scut.edu.cn

***Abstract*—Estimating the real positions of objects in images is a fundamental operation in many data-driven modeling approaches. However, due to the perspective principle, the positions extracted directly from images usually are not accurate enough. Traditional data-driven modeling approaches using these data may not be able to build satisfying models. To solve this problem, this paper proposes an evolutionary algorithm based method to estimate the real positions of objects in images. Specifically, we formulate the problem of estimating the real positions of objects in distorted images as a parameter optimization problem by using some reference information of the images. Then, the differential evolution (DE) is utilized to search for the optimal solution. Our method is tested on two real-world datasets and the experimental results demonstrate that the proposed method is effective to estimate the real positions of objects in images.**

***Keywords*-Position Estimation; Evolutionary Algorithm;Differential Evolution**

I. Introduction

Data-driven modeling approach has now become a popular and powerful approach to model complex systems such as crowd system [1]–[4], traffic system [5], and etc. The key idea of the data-driven approach is to utilize data to train components of the model so that the trained model can fit the real-world data. One of the most commonly used data formats is image (video can also be treated as a series of images). For data-driven modeling approaches based on images, knowing the positions of objects in images is a fundamental operation.

With the rapid development of pattern recognition and image processing, extracting the positions of objects in images becomes feasible [6]. However, the images obtained are usually distorted because the images are usually not recorded in the perfect top-down view. A typical example is shown in Fig. 1, from which we can see that the rectangle ABDC is distorted into a trapezoid because of perspective principles. The positions of objects in distorted images are in-accurate. Training model based on in-accurate data will reduce the accuracy of the trained model.

Theoretically, it is possible to calculate the real positions of objects in distorted images when the parameters of the camera are given in advance. However, this is often impossible. For many public image libraries such as those in [7], the images are distorted but the parameters of the camera are not available.

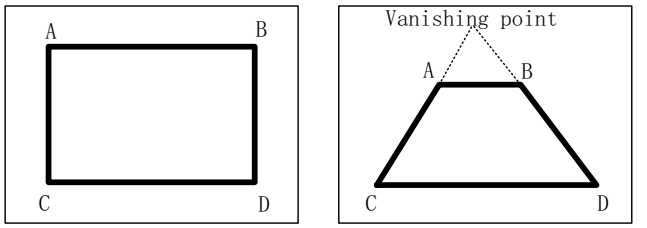


Figure 1: distortion in retinal plane.

So far, there is little work in the literature to deal with this problem. Tsai [8] has proposed a method to estimate position of objects in images, but it requires many intrinsic cameras'parameters. Teknomo [9] proposed a preliminary linear fitting method to solve this problem, but the estimated results are not accurate enough. To address the above issues, this paper proposes a novel method based on evolutionary algorithm.

In the proposed method, we utilize some reference information of the image to get a transformation matrix which is further used to estimate the real positions of objects in images. We formulate the problem of finding the correct transformation matrix as a continuous optimization problem. Then we use the Differential Evolution (DE) to solve the problem. The proposed method was tested on two real datasets [7], [9]. The experimental results demonstrate that the proposed method is effective to estimate the real positions of objects in images.

II. Preliminaries

In this section, we introduce some basic knowledge on coordinate transformation to facilitate readers to better comprehend our method. First, the pixel coordinate system, camera coordinate system, and the world coordinate system are described. Then, the transformation between coordinate systems is presented.

A. Coordinate systems

As shown in Fig.2, we define a rectangular coordinate u-v based on the stored format of pixel in computers. Such

978-1-5090-4993-6/17 $31.00 © 2017 IEEE
DOI 10.1109/CMVIT.2017.20

Authorized licensed use limited to: Victoria University of Wellington. Downloaded on February 17,2022 at 08:04:19 UTC from IEEE Xplore. Restrictions apply.

coordinate system is called the pixel coordinate system, where a pixel is labeled as (u, v). Because the pixel coordinate system can only represent positions of pixels in digital figures. To get the real positions of pixels in physical units, we need to build a retinal coordinate system based on physical unit (such as center-meter). The coordinate in the retinal coordinate system is labeled as (x, y). In x-y coordinate system, the original point, O_1, termed as principal point, is defined as the cross of the optical axis and the retinal coordinate system. The principal point is in the center of the image.

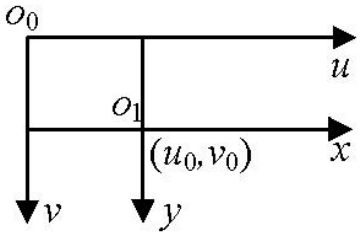


Figure 2: Retinal coordinate system and pixel coordinate system.

Denote the coordinate of O_1 as (u_0, v_0), and the deviation of one pixel in x-axis and y-axis of the retinal coordinate system as dx, dy respectively. The relationship of these two coordinate systems can be demonstrated by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & s' & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

where s' denotes the skew factor caused by nonorthogonality of the axes in retinal coordinate system.

The geometrical relationship of the camera and object is shown in Fig.3, where O denotes the optical center of the camera and Z_c axis denotes the optical axis of camera which is perpendicular with retinal plane. X_c axis and Y_c axis are parallel with x axis and y axis in retinal coordinate system. The cross of optical axis and retinal plane is principle point O'. The camera coordinate system is consisted with O point and $X_cY_cZ_c$ axes. OO' is the focal length of the camera.

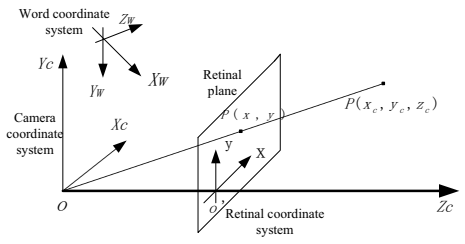


Figure 3: Camera coordinate system and world coordinate system.

To describe the position of camera and object, a world coordinate system in the environment is needed. The relationship between camera coordinate system and world coordinate system can be described by a 3×3 orthogonal identity rotation matrix $\mathbf{R}$ and a 3-dimension translation vector $\mathbf{t}$. Therefore, any point P in the space will have a coordinate of $(X_w, Y_w, Z_w, 1)^T$ in the world coordinate system and a coordinate of $(X_c, Y_c, Z_c, 1)^T$ in camera coordinate system. These two coordinate systems have following relationship

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_1 \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2}$$

Where $\mathbf{0} = (0, 0, 0)^T$, M_1 is the relation matrix between two coordinate systems.

As shown in Fig.4, based on the principle of pinhole imaging, the light who gets through the projection center and is perpendicular to retinal plane is the projection axis or optical axis. $x_1y_1z_1$ is the Cartesian coordinate systems based on the camera following the right-hand rule. The original point is in the projection center. z_1 axis coincides with projection axis. X_c axis and Y_c axis are parallel with x_1 axis and y_1 axis in the retinal coordinate system respectively. The distance between X_cY_c plane and retinal plane, OO_1 , is the focal length of camera F. To avoid the retinal plane being upside down in real camera, we assume there is a virtual retinal plane $X'Y'$ lying in front of the projection center.

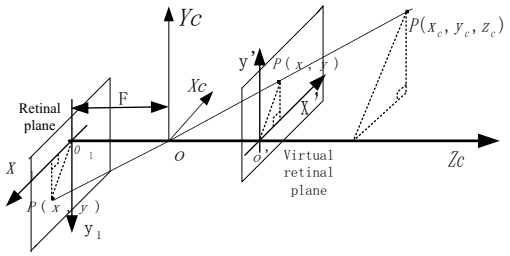


Figure 4: Camera coordinate system and world coordinate system.

B. Transformation between Coordinate systems

The relationship of camera coordinate system and retinal coordinate system is:

$$x = \frac{fX_c}{Z_c}, y = \frac{fY_c}{Z_c} \tag{3}$$

where (x, y) is the coordinate of **P** in the Cartesian coordinates of retinal plane, $P(x_c, y_c, z_c)$ is the coordinate of **P** in the camera coordinate system. We use the homogeneous coordinate matrix to show their relationship:

Authorized licensed use limited to: Victoria University of Wellington. Downloaded on February 17,2022 at 08:04:19 UTC from IEEE Xplore. Restrictions apply.

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{4}$$

Substitute the (1) and (2) into the formula above, we can get the relationship of retinal coordinate system and world coordinate system

$$\begin{aligned} Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} 1/dx & s' & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \\ & \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} a_u & s & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ & = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tilde{\mathbf{X}} = \mathbf{P}\tilde{\mathbf{X}} \end{aligned} \tag{5}$$

Where $a_u = f/dx$, $a_v = f/dy$, $s = s'f$. $[\mathbf{R}\ \mathbf{t}]$ is totally determined by the position of camera in world coordinate system, which is called external argument matrix. $\mathbf{K}$ is only relative with the internal structure of camera, called internal argument matrix. (u_0, v_0) denotes the coordinate of the principle point. a_u and a_v are the dimension factor on the u axis and v axis respectively. s is the argument describing the incline between two coordinate axes. $\mathbf{P}$ is a 3*4 matrix, named projection matrix, which is the transformation matrix from world coordinate system to retinal coordinate system formed by $\mathbf{K}$ and [$\mathbf{R}$ $\mathbf{t}$]. Therefore, for any point in the space, given the projection matrix $\mathbf{P}$ and its three-dimension coordinate, we can get its coordinate on the retinal plane (u, v).

III. Proposed Method

A. General framework of our proposed method

In our proposed method, we aim to transform the original distorted image into a result image which can reflect the real positions of objects correctly. Our key idea is to utilize some reference information to estimate the transformation metric. The reference information R in image is problem specific.

Specifically, based on the discussion on the background knowledge, there is a mapping function which can map each pixel in the original image to a specific point in the result image, i.e.,

$$x' = \Psi(x) \tag{6}$$

where x denotes the coordinate of the pixel in the original image and x' denotes the coordinate of corresponding pixel in the result image. The mapping function Ψ is a transformation matrix which contains a set of unknown arguments $S = \alpha_i | i = 1, 2, ..., d$, where d is the total number of arguments. These parameters are determined by the parameters of the camera. To find out the optimal mapping function, we need to find out the optimal S. Our idea is to utilize some reference information to optimize S. More specifically, we intend to use DE to search for the best S based on the reference information. The reference information is used to define the fitness function φ . Example reference information can be the length of an object in the image or something else. Suppose the reference information in the original image is R, and corresponding information in the results image is T. Then, given a candidate mapping function Ψ, we can calculate the transformed information by $R' = \Psi(R)$. In this way, we can define the fitness function as:

$$\varphi(S) = D(T, R') \tag{7}$$

where T is the true result, and R is the result obtained based on S, and D is the distance measure to calculate the distance between T and R. In this way, we can formulate the problem of finding the best Ψ as a parameter optimization problem: Given R and T, find the best S* that minimize φ , i.e.,

$$S^* = \arg\min_S \varphi(S) \tag{8}$$

In this paper, we adopt DE as the basic algorithm to search for the optimal S* that minimizes the fitness function. Based on the definition above, the procedure of our solution will be as follow.

Step 1: Design the reference information based on the specific scenario, e.g., the length of an object.

Step 2: Define the distance measure D.

Step 3: Define the fitness function φ based on the reference information and D.

Step 4: Use DE to search for the best S based on φ .

Step 5: Get the mapping function based on S, and use the mapping function to estimate the positions of objects in the original image.

B. DE Algorithm

DE algorithm is a powerful evolution algorithm for continuous optimization [10]–[12]. Due to its high efficiency, in this paper, we adopt DE to search for the optimal S that form the mapping function. As an evolution algorithm, DE uses iterations to search for the optimal solution. Each iteration is a generation, named as G, which generates a population of individuals. Each individual encodes a candidate solution of the problem. To have a stronger connection with our problem, we define the individual in DE as a d-dimension vector to represent S. We denote the number of population as N. So, the individuals in generation G can be represented as $G_{i,G}(i = 1, 2, ..., N)$. DE mainly includes three stages, mutation, crossover and selection. DE generates the new vectors by mutation and crossover, and a selection operation is performed to select individuals with better fitness value to be survival in the next generation. These three operations are described as follows.

Authorized licensed use limited to: Victoria University of Wellington. Downloaded on February 17,2022 at 08:04:19 UTC from IEEE Xplore. Restrictions apply.

1) Mutation: In each iteration, the mutant vector, named as Q, will be given birth to the population according to

$$Q_{i,G+1} = S_{k_1,G} + F(S_{k_2,G} - S_{k_3,G}) \tag{9}$$

Where k_1, k_2, k_3 with $k_1 \neq k_2 \neq k_3 \neq i$ is the indexes randomly picked from $[1, N]$. The formula also indicates that those indexed must be different with the current index I . To satisfy this condition, N should be at least 4. F is an important factor of DE, indicating the amplification of vector $(S_{k_2,G} - S_{k_3,G})$. Commonly, F belongs to $[0, 2]$. During the mutation process, if one dimension of an individual exceed upper bound or the lower bound, its value will be set as a bound value.

2) Crossover: Crossover is a procedure simulating the biological behavior which mixed the mutant vector with the parent vector. After crossover, we can get new vectors

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, ..., u_{di,G+1}) \tag{10}$$

where i is the number of each individual. DE implements the crossover by

$$u_{ji,G+1} = \begin{cases} Q_{ji,G+1}, \text{if} r(j) \leq CR \text{ or } j = rn(i) \\ S_{ji,G}, \text{otherwise} \end{cases} \tag{11}$$

for $j = 1, 2, ..., d$. $r(j) \in [0, 1]$ is the jth evaluation of a uniform random generator number. CR is another important factor which has great influence on the performance of DE and it is a user-defined constant $\in [0, 1]$. $rn(i) \in (1, 2, ..., d)$ is a random index guaranteeing $u_{i,G+1}$ gets at least one component from $Q_{ji,G+1}$ so that there must be some updates in next generation.

3) Selection: In the selection operation, the fitness function is used to determine which vector can survive

$$S_{i,G+1} = \begin{cases} u_{i,G+1}, \text{if} \varphi(u_{i,G+1} < \varphi(S_{i,G}) \\ S_{i,G}, \text{otherwise} \end{cases} \tag{12}$$

For $i = 1, 2, ..., d$. That is, if the new vector has a better fitness value than the parent, the parent will be updated by the new vector. Otherwise, the new vector is removed and the parent survives to the next generation.

When the termination condition is met, the best individual is decoded as the final solution.

IV. EXPERIMENT STUDIES

In this section, we test the effectiveness of the proposed method by using two real world datasets. The two datasets contain moving trajectories of people in a train station and a street crossing scenario, respectively. Since the trajectories were not recorded in perfect top-down view, the positions of trajectory points in the original data contain distortions. What is worse, the parameters of the cameras that record the data are not available. Thus, we apply our method to estimate the real positions of trajectory points in the data sets without using the camera parameters.

A. Case 1 study: New York Central Station

In the first case study, we apply our method to a dataset of the New York Central Station scenario. The dataset is obtained from a 33-min video (The original video and extracted trajectories are downloaded from http://www.ee.cuhk.edu.hk/ xgwang/grandcentral.html). Figure5 shows a frame of the video. The dataset includes

Figure 5: A frame of the video in the first scenario.

40000 people moving track record tuples (more than 6000000 tuples). Each tuple consists of ID, x-value, y-value, the time stamp. Our reference information in this case study are the coordinates of four corners in the station square. We first approximately estimate the coordinates of the four corners based on the trajectories, i.e.,

$$\begin{cases} C_1 = C_{right,top} = (544, 431), \\ C_2 = C_{right,bottom} = (695, 0), \\ C_3 = C_{left,top} = (170, 431), \\ C_4 = C_{left,bottom} = (25, 0) \end{cases} \tag{13}$$

The corresponding real positions should be

$$\begin{cases} R_1 = R_{right,top} = (34, 48), \\ R_2 = R_{right,bottom} = (34, 0), \\ R_3 = R_{left,top} = (3, 48), \\ R_4 = R_{left,bottom} = (3, 0) \end{cases} \tag{14}$$

Based on the discussion in Section II, the mapping function in our problem have 18 parameters to determine. The 18 parameters in the transformation matric are described as:

$$TM_1 = \begin{bmatrix} x_0 & x_1 & x_2 \\ 0 & x_3 & x_4 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$TM_2 = \begin{bmatrix} x_5 & 0 & 0 \\ 0 & x_5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$TM_3 = \begin{bmatrix} x_6 & x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} & x13 \\ x_{14} & x_{15} & x_{16} & x17 \end{bmatrix} \tag{17}$$

Authorized licensed use limited to: Victoria University of Wellington. Downloaded on February 17,2022 at 08:04:19 UTC from IEEE Xplore. Restrictions apply.

And our fitness function is defined as

$$\varphi(S) = \sum_{i=1}^{4} \|R_i - C_i\| \tag{18}$$

We use DE to search for the optimal values of the 18 parameters so as to minimize Eq.(18). To improve the robustness of DE, F and CR are randomly selected from $[0, 1]$ for each generation and individual.

$$TM_1 = \begin{bmatrix} -4253.97 & -229.863 & 132.709 \\ 0 & -2654.14 & 262.479 \\ 0 & 0 & 1 \end{bmatrix} \tag{19}$$

$$TM_2 = \begin{bmatrix} 1.88374 & 0 & 0 \\ 0 & 1.88374 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$TM_3 = \begin{bmatrix} 6267.09 & 758.97 & 1.26e+15 & -43442 \\ 2.49e-13 & 5464.5 & -1.72e+14 & -121989 \\ 4.74e-12 & -38313.1 & -3.28e+12 & -2.32e+6 \end{bmatrix} \tag{21}$$

The final transformation matrices are expressed in Eq(19)-(21). Figure 6 shows the trajectories in the original data and those after transformed respectively. It can be observed that our method transforms the original retinal plane figure, like a trapezoid, into a rectangle which can reflect the 2-dimension information more directly as we look down from the right above. The above results indicate that our method is effective to estimate the real positions of objects in images.

B. Case study 2

In the second case study, we apply our method to a dataset of a street crossing Scenario. The scenario data was obtained from a video recorded in Japan, as described in [9]. The video lasts for 60s and involves about 150 pedestrians. The footway region for pedestrian crossing is a $11m \times 31m$ rectangle and the positions of pedestrians are recorded every 0.5s. Figure7(a) and (b) shows a frame of the video and the original dataset respectively.

In this case study, the reference information is two edges of the zebra stripe region. Our objective is to find the correct parameters in the transformation matrices so as to transform the distorted trapezoid into a standard rectangle. Hence, the objective function is defined as the sum of Euclidian distance between all points on two edges in image coordinate system and their corresponding coordinate in real world coordinate system, i.e.,

$$\varphi(S) = \sum_{i} |X_i - X_i'| \tag{22}$$

where X_i is a point on the two edges in image and X_i' is the corresponding point in real world coordinate system. As in the first case study, we use DE to search for the best parameters that minimize the objective function. The settings of DE is the same as in the first case study.

We also use the linear fitting method proposed in [9] to estimate the real positions of the trajectory points. Figure 8 shows the transformed trajectories offered by the linear fitting method and our method respectively. It is clear that the two edges of the zebra stripe rectangle are not parallel in the results of the linear fitting method, while those of our method are almost parallel. It seems that our method can generate more realistic results.

In general, with sufficient reference information, our model can estimate the real position of object more accurate and retain the expected profile. It should be noted that the reference information has significant influence on the performance of the algorithm. If the reference information is not sufficient or inadmissible, the result image will be distorted again.

V. Conclusion

In this paper, we have proposed an evolution algorithm based method to estimate the real positions of objects in images. Our key idea is to utilize some reference information of the image to estimate the transform matrix that converts the image positions to real world positions. We formally define the problem as an optimization problem and then develop a differential evolution to solve the problem. We test our method on two real-world data sets and the experiment results have demonstrated the effectiveness of our method. As for future work, we plan to apply our method to more real world cases. In addition, utilizing our method to improve the existing data-drive modeling approaches is also a promising research topic.

Acknowledgment

The research reported in this paper is financially supported by the National Natural Science Foundation of China (Grant No. 61602181).

References

[1] A. Lerner, C. Yiorgos, and L. Dani."Crowds by example," *Computer Graphics Forum*. Vol. 26. No. 3. Blackwell Publishing Ltd, 2007.

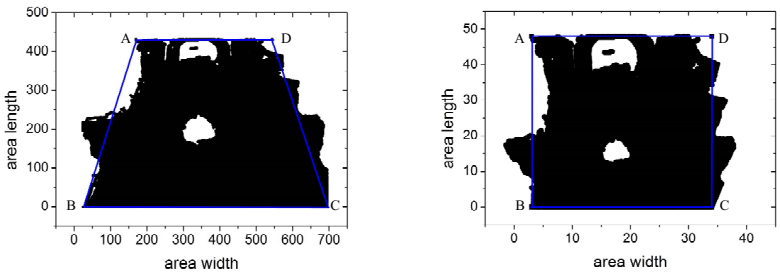
[2] J. Zhong, and et al. "Learning behavior patterns from video for agent-based crowd modeling and simulation," *Autonomous Agents and Multi-Agent Systems*, 2016, 30(5): 990-1019.

[3] K. H. Lee, M. G. Choi, et al. "Group behavior from video: a data-driven approach to crowd simulation," *In Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 2007, pp.109-118.

[4] C. Chen, et al."Distributed modeling in a mapreduce framework for data-driven traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 14.1 (2013): 22-33.

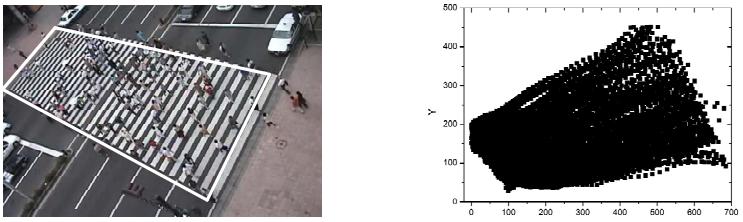
[5] D. Wilkie, S. Jason, L. Ming,"Flow reconstruction for data-driven traffic animation," *ACM Transactions on Graphics (TOG)*. 32.4 (2013): 89.

Authorized licensed use limited to: Victoria University of Wellington. Downloaded on February 17,2022 at 08:04:19 UTC from IEEE Xplore. Restrictions apply.



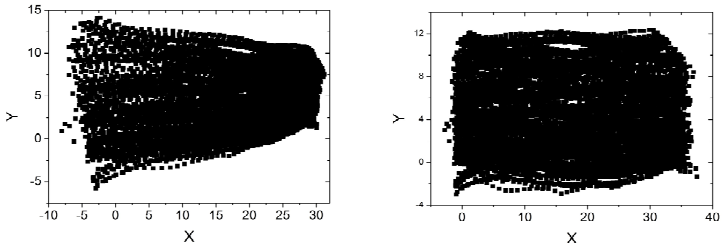
(a) the original extracted points (b) the transformed points

Figure 6: Final results of the proposed method on the first scenario.



(a) A frame of the video (b) Extracted trajectories

Figure 7: A frame of the video in the second scenario and the Dataset.



(a) Results of the linear fitting method (b) Results of our method

Figure 8: Final results of the linear fitting method and the proposed method.

[6] D. Ryan, et al. "Crowd counting using multiple local features," *In Digital Image Computing: Techniques and Applications*, 2009. DICTA'09., pp. 81-88. IEEE, 2009.

[7] B. Zhou, et al."Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," *In 012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2, pp. 2871-2878. IEEE, 2012.

[8] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine visio," *In Proceedings of IEEE conference on computer vision and pattern recognition*, 1986, pp. 364374), Miami Beach, FL.

[9] K. Teknomo, "Application of microscopic pedestrian simulation model," *Transportation Research Part F: Tranc Psychology and Behaviour*, 2006, 9 (1):15-27.

[10] R. Storn, K. Price "Differential evolutiona simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization* 11, no. 4 (1997): 341-359.

[11] J. Zhong, et al."A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem," *IEEE Transactions on Evolutionary Computation*, 17(4), pp.512-527.

[12] S. Das, P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art,"*IEEE transactions on evolutionary computation*, 2011, 15(1): 4-31.

Authorized licensed use limited to: Victoria University of Wellington. Downloaded on February 17,2022 at 08:04:19 UTC from IEEE Xplore. Restrictions apply.