

Systematic Parameter Optimization and Application of Automated Tracking in Pedestrian Dominant Situations

Date of submission: 2014-08-01

Dariush Ettehadieh*
M.Sc. Student,
Polytechnique Montréal,
2500, Chemin de Polytechnique, Montreal
phone : 1-514-266-5544
dariush.ettehadieh@polymtl.ca

Bilal Farooq
Assistant Professor,
Polytechnique Montréal
2500, Chemin de Polytechnique, Montreal
phone : 1-514-340-4711 ext. 4802
bilal.farooq@polymtl.ca

Nicolas Saunier
Associate Professor,
Polytechnique Montréal
2500, Chemin de Polytechnique, Montreal
phone : 1-514-340-4711 ext. 4962
nicolas.saunier@polymtl.ca

5029 Words + 4 Figures + 3 Tables = 6779

Submitted for presentation to the 94th Annual Meeting of the Transportation Research Board and publication in Transportation Research Record: Journal of the Transportation Research Board.

*Corresponding author

ABSTRACT

Though a wealth of data exists for the characterization of pedestrian movement, a majority of said data originates from experimental settings owing to the current state of trackers for real-world scenarios. While these trackers are steadily improving, they remain insufficiently reliable for the accurate, microscopic tracking of individuals, particularly in cases of occlusion or higher density, complex scenes. We propose the use of evolution algorithms in the systematic calibration of the parameters of existing trackers in order to further optimize their performance – evaluated by tracking accuracy and precision metrics – in complex cases, with an initial focus on two tracking methods designed for multimodal analysis. Two real test cases were used a) a confined corridor in a public building and b) a subway station entrance during morning rush hour. Current results demonstrate a halving of tracking errors over both default and manually-calibrated parameters, as well as a strong correlation in performance between similar cases. For applications, flow characterization and directional counting are demonstrated.

INTRODUCTION

In recent years, active modes of transportation, especially walking, have been the focus of attention in transportation research. Conventional data gathering and surveying methods are limited in their scope for providing detailed information on pedestrians' walking behavior. Therefore, computer vision-based automated data-gathering techniques have been used. These techniques are primarily utilized to extract trajectories and count data. Pedestrian tracking is, however, a difficult problem, particularly when seeking data of sufficient accuracy for the calibration of pedestrian flow models (1). While video tracking, unlike many other methods, has the potential to allow extraction of data for each and every pedestrian in the field of the video frame, it must contend with substantial challenges such as occlusion, grouping, and the myriad of visual effects (e.g. lighting, shadows, distortion, non-human moving objects) which can confuse an automated tracker (2).

Currently, pedestrian model calibration using video data has for the most part relied either on data from experimental settings (3) or on video recorded in similarly favorable conditions (4). Though both sources undeniably provide valuable, accurate trajectory data, the applicability of either to larger, more complex cases is difficult. Furthermore, these tracking methodologies are impractical to apply to the majority of existing footage of public spaces, namely surveillance and other low-angle recordings. Recently, however, more generalized tracking methods, relying on the hybridization of methodologies and/or advanced filtering and human-detection algorithms, have made great strides. Within pedestrian-dominant settings, Measures Of Tracking Accuracy (MOTA, best possible value of 1 – see (5)) of more than 0.80 have been attained (6) though accuracies in the 0.50-0.60 range appear to be more common (7). Such measures are, however, difficult to interpret: tracker performance is dependent not only on a tracker's attributes but also on scene complexity, a feature rarely prominent in their evaluation.

This paper focuses on improving the performance of existing video-based trackers in pedestrian-dominated environments. This is accomplished through the application of evolutionary algorithms to the underlying parameters of the trackers, with emphasis on both fine-tuning for diverse flow scenarios (ranging from low-density, uni-directional flow to high-density movement in multiple directions) and the applicability of the methodology to a variety of trackers. Evolutionary methods have been applied to pedestrian tracking before: (13) applied evolution strategies to improve upon the segmentation stage of a background-subtraction-based tracker, achieving a 25 % decrease in positional error of the resulting tracks. In contrast, instead of targeting specific facets of the problem, our method aims at evolutionary optimization of the entire tracking method through calibration of all underlying parameters, with the goal of achieving solution parameters which are optimal, or near-optimal, for a given range of similar scenes. The flexibility of these simulation-based optimization methods make them favorable for use in highly non-linear, multidimensional search space problems, as at hand. In addition, they are easily transferable to similar cases. This means that the methodology developed in this paper can be applied to a wide range of trackers.

The remainder of this paper is structured as follows. After the introduction, a review of the literature is presented, with particular emphasis placed on the two trackers subjected to the optimization algorithm. The next section describes the developed methodology, including the studied data-sources, the utilized performance metrics and describes the optimization algorithm itself. The presentation of the results follows. The paper ends with a brief overview of the resulting trajectory data, and finally conclusions and recommendations for further research.

LITERATURE REVIEW

Video-based tracking can, within most methods, be subdivided into two primary stages: detection of the objects to be tracked, and the data association of their movement through the image- or world-space (8). Each

stage has been subject to a number of different approaches. A brief summary of recent works on the subject is presented in table 1.

At the detection stage, feature-based tracking (used in (9), (10), (11) and (12)) consists of following distinct features (or groups of pixels, e.g. corners) as they move through the image. As features tend to be small, this allows the continuous tracking of partially-occluded subjects. However, the requirement that features move between frames leads to loss of a target if it remains stationary. Similarly, tracking-by-detection methods also detect pixel groups, though instead of moving features they search for predefined features. Such methods include shape and object detection: as humans tend to change shape during each stride, these shapes tend to be either the head (as in (19) and (20)) or both shoulders and head (18). These methods are therefore able to follow targets even if they stop for an extended period of time (or indeed, do not move at all) though they require an unobstructed view of the shapes they are designed to detect. Finally, background subtraction methods ((13), (14), (15), (16) and (17)) track movement as “blobs” over a static background. They have the advantage of requiring little to no consistency within tracked objects, at the cost of difficulty distinguishing objects that are too close together and making errors when associating trajectories to objects occluding each other.

The subsequent stage consists largely of consolidating the raw observations into consistent, accurate tracks. This can be accomplished through means ranging from the simple deletion of clearly erroneous results (9) (e.g. objects moving too quickly or too erratically to be pedestrians) to the application of particle (17) or Kalman (21) filtering, to track inter- and extrapolation by fitting to a given model (15). Moreover, such methods are increasingly being hybridized, such as in (14) and (19).

While accuracies are consistently improving across all these tracking methods, two primary issues remain, which constitute the focus of this paper. First, there is a tendency to evaluate their accuracy only within a small range of scene complexities. This makes application to extended real-world scenes problematic; a subway station, for instance, may experience a relatively constant, low pedestrian flow rate, punctuated by high-density surges following a train’s arrival. These two periods impose vastly different requirements on the utilized tracker (23). One’s ability to attain high performance on complex scenes is not necessarily indicative of performance in simpler ones: the bacterial foraging algorithm utilized in (14) performed markedly worse on the relatively low-density CAVIAR dataset than in crowded scenes.

The second issue is the limited emphasis placed on the parameterization of the trackers. In the few cases where they are discussed, there is a tendency to either manually set parameter values to intuitive values, or to report results for a small number of such values in order to gauge their sensitivity. When automated calibration does occur, it is habitually focused on a particular facet of the tracking problem (e.g. Adaboost training of the shape detectors in (18) and (20), and evolutionary optimization of the segmentation stage in (13)). Global, systematic optimization of tracker has rarely (if ever) been addressed.

Overview of the optimized trackers

In this paper we selected two trackers for the purpose of parameter optimization. Following is a brief overview of both of them.

Table 1 - Summary of selected pedestrian tracking methods. It should be noted that reported accuracies were evaluated in different scenes and with a variety of metrics, making direct comparison difficult. Pedestrian density in the table corresponds to a visual evaluation of density in the scenes: low density denotes that individuals are generally clearly separated, while high density implies substantial overlap and/or grouping of pedestrians. DR: Detection Rate, the ratio of accurate tracks as defined by each author.

Tracking method	Authors	Year	Novel contributions	Test sequence characteristics	Accuracy (metric)	Parameterization	Notes
Feature-based	Saunier & Sayed (9)	2006	Application to transportation systems	Multimodal; low pedestrian density	0.23-0.65* (MOTA)	Manually set	Designed for multimodal traffic
	Rabaud & Belongie (10)	2006	Pedestrian counting in dense crowds	High density, complex scenes	0.78-0.93 (DR)	Partial training of detector	Designed specifically for counting
	Ismail et al. (11)	2010	Automated video analysis for before/after safety evaluations	Multimodal; low pedestrian density	0.70 (DR)	Optimization of a selection of tracker parameters	
	Khanloo et al. (12)	2012	Hybridizes feature-based and Histogram of Oriented Gradients	Low density, multiple directions	<i>Unreported</i>	Manually set	PETS dataset
Background subtraction	Pérez et al. (13)	2006	Evolutionary optimization of the segmentation stage	Single pedestrian	<i>Unreported</i>	Evolutionary optimization	Optimization of only one tracker stage
	Berclaz et al. (14)	2011	K-shortest paths optimization	Low density, multiple directions	0.58-0.86 (MOTA)	<i>Undisclosed</i>	CAVIAR dataset
	Nguyen & Bhanu (15)	2012	Bacterial foraging optimization	Varied	0.35-0.71 (DR)	Manually set	Best performance in high-density scenes
	Jodoin & Saunier (16)	2013	Use of feature-points for object identification	Multimodal; low pedestrian density	0.68-0.93 (MOTA)	Manually set	Designed for multimodal traffic
	Guan et al. (17)	2013	Particle filter approach	Low density, uni- and bi-directional	0.63-0.92 (MOTA)	Manually set	
Tracking by detection	Sidla et al. (18)	2006	Searches image for shoulder-and-head shapes	High density, complex scenes	0.89 (DR)	Manually set	Focused on pedestrian counts through tracking
	Singh, Wu & Nevatia (19)	2008	Completes tracks using low-confidence traces	Low density, multiple directions	0.73 (DR)	Manually set	CAVIAR dataset
	Ali & Dailey (20)	2009	Confirmation-by-classification	High density, bi-directional	0.77 (DR)	Adaboost	Head detection for dense crowds
	Jiang et al. (21)	2010	Hybridizes Histogram of Oriented Gradient (HOG) detection with color tracking	Low density, multiple directions	<i>Unreported</i>	Manually set	CAVIAR dataset
	Andriyenko & Schindler (22)	2011	Tracking using energy minimization	Low density, multiple directions	0.33-0.85 (MOTA)	Manually set	Tested with multiple sets of parameters

*Measurements published in (16)

Traffic Intelligence

The open-source Traffic Intelligence (TI) project is an implementation of feature-based tracking, described in (9). Initially designed for the monitoring of road traffic, TI is used for multimodal tracking of the complex movements within intersections, notably including conflict detection between vehicles and pedestrians. Utilizing a feature-based tracking method (specifically, the Kanade-Lucas-Tomasi method as implemented in OpenCV (24), (25)) it can cope with partial occlusion by following distinguishable elements of a moving object rather than the object as a whole, resulting in good accuracy when used in its intended cases (MOTAs between 0.6 and 0.85 (16)). Such cases, however, primarily involve vehicles: MOTA calculated for pedestrians alone tends to be lower (near 0.50) despite pedestrian density typically being very low.

Urban Tracker

Like TI, Urban Tracker (UT) (16) was designed primarily for road traffic, though in contrast to the previous tracker it utilizes background subtraction for moving-object identification and tracking, and seeks to make as few assumptions about the tracked objects as possible. Built around the ViBe background subtraction algorithm (26), UT expands the tracking method by identifying features of each tracked object (similarly to TI) allowing reacquisition in cases of fragmentation, grouping or occlusions. Direct comparisons with TI have demonstrated significantly higher accuracies in pedestrian tracking (MOTAs ranging from 0.70 to 0.90) in the same sequences as examined above, though again pedestrian density was low.

METHODOLOGY

Test cases

Test data was collected from two locations. The first is a central hallway within Polytechnique Montreal, serving as the primary means of movement between the two buildings of the school, and including a stairway and access to auxiliary hallways and classrooms. This area was recorded from two angles, at separate ends of the corridor. The second location is the exterior of the subway station/bus terminal, in Montreal, with cameras covering both entrances as well as the entirety of the exterior terminal. From the latter location, an entrance with access from all four cardinal directions was selected as the primary focus for optimization due to its complexity. In both cases, multiple hours of video were recorded using wall-affixed wide-angle cameras at a resolution of 1280x720 pixels, during periods selected to include the busiest portions of the day.

Tracks (ground truth) were entered manually for four separate one-minute sequences (figure 1) each selected to contain a range of densities as well as some cross- or bidirectional flow. In the Polytechnique sequence, these correspond to the end of morning courses, whereas in the case of the subway station they follow the simultaneous arrival of a bus and the outflow of arriving subway passengers. Two sequences were drawn from each location, covering the same area (from alternate angles in the case of Polytechnique); in each case, one sequence is used for calibration, and the second to test the generalizability of solutions to similar situations. Densities were estimated to range between 1 and less than 0.1 pedestrian per square meter, reaching the upper limit primarily in the subway station test sequence. All sequences, with the exception of the Polytechnique calibration sequence, include one or more loitering individuals whose limited movements served as an additional challenge for the tracker.

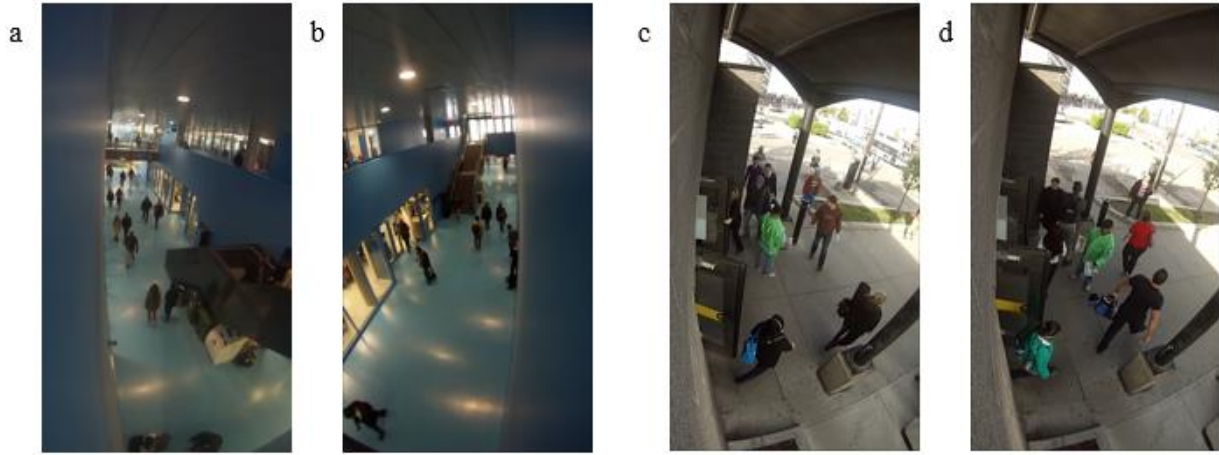


Figure 1 - Example frames taken from the four studied videos. a. and c. represent the calibration sequences recorded in Polytechnique Montreal and the subway station, respectively. b. represents the same location and movement complexity as a., but recorded at a different time by a camera installed at the opposite end of the hallway. d. represents footage taken from the same camera as c., yet recorded during the arrival of a bus, resulting in higher pedestrian volume and densities near the door.

Fitness measure: MOTA & MOTP

The primary fitness measure used in the optimization of TI is the tracking accuracy, as measured by MOTA. This metric requires a preliminary matching step between the tracks output by the tracker and the ground-truth. To count as a match within a given frame, a track must first be the closest track to the ground-truth object. Second, it must be within a maximum distance of said object; given our interest in microscopic model calibration, this maximum distance was set at one meter, based on average human build and step size. It should be noted that this, as well as the following, methodologies are a direct application of the CLEAR-MOT metric as defined in (5). Though this matching methodology can lead to arguably avoidable mismatches (e.g. at the intersection of two tracks, as was corrected for in (16)) the base method was maintained both for its reproducibility and to avoid interfering with any post-processing that may occur within the tracker.

While the ground-truth was first established for all individuals visible in each scene, initial tests revealed the inability of the tracker to reliably track pedestrians smaller than 20-30 pixels without incurring substantial over-detection of larger/closer objects due to the increased sensitivity. In order to attenuate this problem, matching was therefore limited to the closer areas with higher complexity in both scenes (in proximity to bottlenecks, obstacles and multidirectional flow); these areas correspond to the approximately 35 meters of hallway between the two cameras in the Polytechnique sequences, and to within twelve meters of the camera in the subway station videos. To avoid unjustly penalizing the tracker, tracks more than one meter outside these areas were ignored.

Once the matches have been computed, MOTA is calculated as:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

Where m_t , fp_t and mme_t are the number of misses, false positives and mismatches, respectively, at frame t , and g_t the number of ground truth objects in the same frame. Although MOTA has a maximum possible value of 1 (representing perfect tracking accuracy) negative values are possible if sufficient errors (false positives) are made.

The Measure Of Tracking Precision (MOTP) represents the average distance between computed and real tracks across all frames and objects. As defined in (6), it ranges from zero (no error) to a maximum

corresponding to the maximum matching distance. However, in order to facilitate its use alongside MOTA in the optimization algorithm via weighted average, MOTP was normalized to vary on the same scale and in the same direction according to the equation below:

$$MOTP = 1 - \frac{\sum_{i,t} d_t^i}{T \cdot \sum_t c_t}$$

Where T , d_t^i and c_t represent the maximum matching distance, position error and total number of tracker points, respectively.

Evolutionary optimization algorithm

A simulated annealing algorithm was developed specifically for video-tracking optimization (27); its basic structure is summarized in figure 2. This algorithm was selected because it stochastically allows movement to less optimal solutions, permitting nimble avoidance of local maxima, particularly during the initial iterations.

Every iteration i begins by running the tracker on the test sequence with a given set of parameters, or state. The resulting tracks are compared to the pre-established ground truth in order to establish MOTA and MOTP, of which the energy V' is the weighted average. This is then compared to the prior energy V , and the algorithm moves to the new position with a probability P :

$$P = \min\left(1, \frac{e^{T \cdot x \cdot V'}}{e^{T \cdot x \cdot V}}\right)$$

T is the current temperature of the algorithm, defined by:

$$T = \lambda \ln(1 + i)$$

λ and x are constants which, together, determine the energy difference for which a move to a lesser energy state is likely at any iteration. In order to fix maximum downward movement to 0.05, x was set to 10. λ controls the rate at which said maximum decreases between iterations; given that trackers vary in number and complexity of parameters (as well as running time) this constant was manually adjusted for each case.

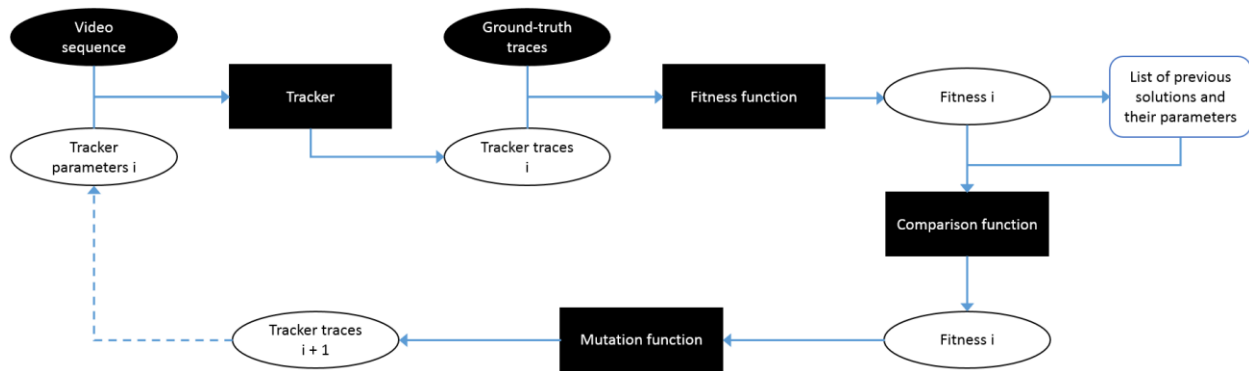


Figure 2 - Flow diagram of the parameter optimization algorithm.

From the selected state, a neighbor solution is generated. The new state is generated by randomized selection of one to three parameters, followed by equally random – but bound – addition/subtraction. Boundaries were specified owing to the varying natures of the parameters, and were decreased whenever

Table 2 - Parameters targeted for optimization in the two tested trackers.

	Parameter name	Type	Min.	Max.	Description
TRAFFIC INTELLIGENCE					
	feature-quality	float	0	1	Minimum quality of corners to track.
	min-feature-distanceklt	float	0	10	Minimum distance between features, in pixels.
	window-size	int	3	10	Distance within which to search for feature in next frame, in pixels.
FEATURES	pyramid-level	int	1	-	Maximum pyramid level for feature tracking.
	ndisplacement	int	2	4	Number of displacements to test minimum feature motion.
	min-feature-displacement	float	0	0.1	Minimum displacement of features between frames (pixels)
	acceleration-bound	float	1	3	Maximum ratio of speeds between frames.
	deviation-bound	float	0	1	Maximum cosine of feature trajectories between frames.
	smoothing-halfwidth	int	0	11	Number of frames to smooth positions.
	min-tracking-error	float	0.01	0.3	Minimum error to reach to stop optical flow.
	min-feature-time	int	5	25	Min. time (in frames) a feature must exist to be saved.
	OBJECTS	mm-connection-distance	float	0.5	2
mm-segmentation-distance		float	0.1	1.9	Segmentation distance, in meters. Must be less than connection distance.
min-features-group		float	1	4	Minimum number of features required to create a group.
HOMOGRAPHY	elevation-1	float	0	1.5	Elevations relative to ground-level of each of the four points used to calculate the homography matrix, in meters.
	elevation-2	float	0	1.5	
	elevation-3	float	0	1.5	
	elevation-4	float	0	1.5	
	homography-correction	float	-0.5	0.5	Elevation difference between tracker and ground-truth homographies, in meters.
URBAN TRACKER					
BACKGROUND SUBTRACTION	bgs-minimum-blob-size	int	10	-	Min. size of blobs, in pixels.
	max-lost-frame	int	1	-	Max. number of frames to continue searching for a lost object.
	max-seg-dist	float	0	1	Max. distance between two blobs to be considered on object, as a ratio of blob diameter.
	max-hypothesis	int	1	-	Max. frames to consider an object hypothesis.
	minimum-match-between-blobs	int	1	-	Min. number of matching features to establish two blobs as the same object.
FEATURE DETECTION	brisk-threshold	int	1	20	Threshold determining minimum quality of features to detect.
	brisk-octave	int	1	5	Number of layers to use in feature detection for each frame.
	match-ratio	float	0	1	Min. matching ratio between second-best and best match for a given object.
FUNCTIONS	urban-isolated-shadow-removal	boolean			Automated shadow removal.
	verify-reentering-object	boolean			Verifies whether entering objects correspond to preexisting ones.
	bgs-remove-ghost	boolean			Retroactively removes blobs if they are not associated to an object.
HOMOGRAPHY	elevation-1	float	0	1.5	Elevations relative to ground-level of each of the four points used to calculate the homography matrix, in meters.
	elevation-2	float	0	1.5	
	elevation-3	float	0	1.5	
	elevation-4	float	0	1.5	

the algorithm stagnated for a sufficient number of iterations in order to strike an acceptable balance between convergence time and an optimal final solution. Once new parameters are selected, they are fed back to the tracker, and the following iteration begins.

Application of this framework to a given tracker requires only two adjustments to the underlying functions. The first of these is to ensure that the tracker's output is compatible with the ground truth, itself defined as the center of the bounding box for each individual. Given methods that identify particular features of pedestrians (a notable example being head-detecting methods) some correction is required in order to avoid unjustly penalizing the tracker's evaluated performance.

While UT computes bounding boxes and therefore requires no particular adjustment, feature-based tracking methods such as TI exhibit higher detection rates on the relatively static upper body. When computing positions within the video-frame, the correction factor for the latter tracker would therefore consist of lowering the position of the detected object. In the cases tested, in contrast, where tracking was performed in the world-coordinates via a pre-established homography, correction was instead achieved by automated repositioning of the points in the image-space used for calculation of the homography matrix. In both cases, the correction factor was included as an additional parameter for optimization.

The second required adjustment to the algorithm is that of the function regulating the parameter state generation function between iterations. The parameters for the tested trackers are presented in table 2. In TI, 14 parameters affect the tracking process and were therefore optimized. These parameters can largely be divided into two primary functions: 11 influence feature detection and tracking (notably, minimum quality of features to track and bounds on acceptable movement between frames) and 3 influence the grouping of detected features into objects (i.e. how many features are required to create an object, and how similar their behavior must be to be considered as belonging to the same pedestrian). In UT, 11 parameters manage tracking behavior, functionally subdivided into three groups: background subtraction (e.g. minimum size of blobs to track), feature detection, and three Boolean values regulating the use of specialized sub-functions, such as automatic shadow removal for outdoor scenes.

To these, four additional but functionally identical parameters were added. Given that pedestrians are generally taller than wide and that video sequences focused on pedestrians were expected to be recorded at close range and low angles, establishing the homography matrix through the use of points at ground level (as is typically done with TI for road safety analysis) is problematic. Therefore, in order to accommodate both the difficulty of predicting the elevation at which pedestrians would ultimately be detected and that of establishing that elevation in the video frame, point correspondences were entered as four in-frame vertical lines, and the elevations to use in computing the homography were included as four additional parameters for optimization.

RESULTS

In order to form a basis for comparison, MOTA and MOTP were evaluated for parameters calibrated manually on the calibration scene by the authors of the respective trackers. The algorithm was run three times from different initial states; only the last of these is reported here, though the first two are discussed in the following section. In all cases, performance was significantly higher after algorithm optimization than using manual calibration (table 3). In the case of UT, a relatively simple pedestrian-only scene used in the original paper was made available, alongside the utilized parameters. Applying the algorithm to this scene also demonstrated substantial improvement (MOTA of 0.94 vs. 0.86) implying that the observed amelioration is not solely a result of lax parameter selection.

Table 3 - MOTA/MOTP (in meters) after both algorithm and manual optimization (performed by one of the tracker's authors) on the calibration cases. UT consistently crashed during tracking when applying the manually-selected parameters to the subway station test scene (a known error within the tracking software) so no results could be reported.

Tracking parameters:	Traffic Intelligence		Urban Tracker	
	Calibration scene	Test scene	Calibration scene	Test scene
Polytechnique Corridor				
- Manual	0.28/0.52	0.48/0.49	0.70/0.71	0.10/0.71
- Algorithm-calibrated	0.59/0.53	0.52/0.59	0.89/0.70	0.42/0.72
Subway Station Entrance				
- Manual	-0.01/0.49	-0.22/0.54	-1.62/0.61	-
- Algorithm-calibrated	0.51/0.56	0.26/0.56	0.54/0.67	0.38/0.66

Performance of both trackers deteriorates when the calibrated parameters are applied to the test sequences, but remains better than manual calibration on the same test scenes. Within the Polytechnique sequence, this may be a result of over-specialization for the point of view of the test scene, and is more marked in the case of UT. Between the two subway station scenes, the higher density test sequence appears to be substantially more difficult than the calibration sequence, likely a result of the increased pedestrian grouping involved. Performance differences are however not limited to the algorithm, as the manually-calibrated parameters demonstrate proportionally larger variations.

Traffic Intelligence

When applied to TI, the algorithm converged to a solution within approximately 900 iterations (roughly 30 hours of processing on a 2.67GHz Intel i5 processor) in both cases. TI's performance appears more dependent on the complexity of the pedestrian movement than on geometry or camera angle. The two Polytechnique sequences, ostensibly of similar complexity but with opposite camera angles, demonstrate comparable MOTAs and proportions in their errors. Conversely, application of the subway-station-optimized parameters to the superficially identical test sequence (which in fact represents footage from the same video, recorded only two minutes prior to the calibration sequence) yields half the accuracy and a markedly higher number of misses (typical observed errors are examined in greater depth in the next section). As similar behavior occurs in UT, this is likely attributable to increased difficulty in the test sequence, a hypothesis supported by the increased densities visible in the video.

Urban Tracker

UT is a markedly slower tracking method than TI, with run-times per iteration ranging from one to three hours, depending on the utilized parameters and hardware. The algorithm was therefore terminated after only 90 iterations (more than 100 hours) and included some manual tuning, specifically fixing the Boolean parameters once their effect appeared to be established. Fortunately, convergence was markedly faster than TI's, likely owing to the reduced search-space resulting from a lower dimensionality. The resulting pedestrian tracks displayed both higher accuracy and precision than the former tracker in all but the Polytechnique test sequence. Comparisons with the manually-set parameters and between successive iterations of the algorithm, however, exhibit increased sensitivity with regards to both the input parameters (particularly those affecting background subtraction) and to camera position.

DISCUSSION

As noted previously, the presented results are those of a third run of the optimization algorithm, which used the manually-calibrated parameters as the starting point. The first and second were initiated from the “generic” (taken from examples provided by the authors) and randomized parameters, respectively, and in both cases tracker performance improved slowly: after 3000 iterations, tracks produced by TI had MOTAs of less than 0.30 when initial parameters were randomly set.

Visualizing these tracks before and after optimization in the initial runs suggests that the difference is not primarily a consequence of the algorithm itself, but a result of a qualitative difference in the original tracks leading to alternative optimization strategies. Tracks generated using random parameters consist largely of noise (particularly in the case of TI) and optimization of said parameters seems to lead to heuristic strategies; for example, decreasing feature-detection sensitivity and increasing the grouping range, leading to accurate detection of groups, but also the inability to distinguish the individuals within them. In contrast, manually-calibrated TI tracks movement relatively well, limiting the problem to, primarily, one of grouping and homography. This advantage also translates into the faster observed convergence (a half-correct global optimum has better performance, and is therefore easier to detect, than “half-correct” noise) and more generalizability (as the tracker’s performance is not as tied to the specifics of the calibration scene).

The energy function also proved problematic during the initial runs. While MOTA can only be improved through error reduction and better association of detected-to real objects, the simple matching heuristic utilized in calculating MOTP means measured precision can be improved simply by increasing the number of potential matches. At relatively low values of both metrics, it is therefore easier to improve their weighted average through increased noise than by the desired overall improvement of tracks, effectively trapping the algorithm in local maxima with good precision at the expense of accuracy. In order to attenuate this effect, MOTP is only included in the energy function once overall performance is judged near-optimal, and then only with a relative weight of 10 %.

Despite the performed optimizations, certain types of error remained common in both trackers, as can be seen in figure 3. In the Polytechnique corridor sequences (Fig. 3.a and 3.b) while pedestrian detection was excellent and very few misses occurred, over-detection was problematic in both trackers, namely TI associating two foreground objects to a single pedestrian and UT identifying two individuals as both two distinct objects and an additional object encompassing both. It should be noted that in TI’s case, this over-detection in the foreground was accompanied by an increased number of misses in the background, whereas UT displayed no such phenomenon. This likely accounts, at least in part, for the latter’s increased accuracy, and implies that TI was partially being optimized for a given range corresponding to the center of the tracked area. This type of error is equally penalizing to MOTA as misses would be, but is arguably preferable given that they may be eliminated through additional filtering (e.g. eliminating objects too close to others in the TI example and objects that are too large in UT). UT already includes some such filtering, though as it relies on two objects merging before – or diverging after – a period of grouping, it fails if no such events can be observed.

In the subway station sequence, conversely, the higher pedestrian density results in both trackers occasionally over-grouping pedestrians, as shown in figures 3.c and 3.d. In TI, this can lead to the rather aberrant traces such as exemplified in the figure, as the tracker attempts to ascertain the position of an object based on features from two distinct individuals. Unlike over-detection, over-grouping unfortunately does not lend itself to post-processing solutions akin to filtering, and such errors therefore represent strict limitations of the trackers themselves.



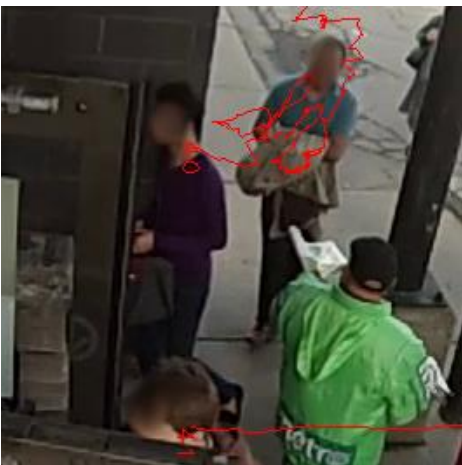
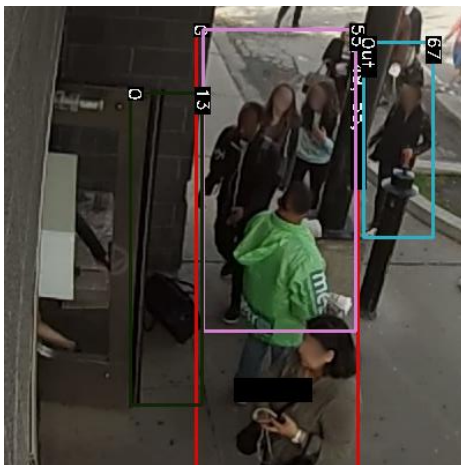


Traffic Intelligence	Urban Tracker
 <p data-bbox="483 730 511 751">a.</p>	 <p data-bbox="1104 730 1131 751">b.</p>
 <p data-bbox="483 1234 511 1255">c.</p>	 <p data-bbox="1104 1234 1131 1255">d.</p>
 <p data-bbox="483 1743 511 1764">e.</p>	 <p data-bbox="1104 1743 1131 1764">f.</p>

Figure 3 - Typical errors observed after parameter optimization. a. and b.: Over-detection in the foreground of the hallway sequence. c. and d.: Grouping errors in the subway sequences. e. and f.: Detection of the subway station doors generated extra tracks.

A final notable error, which lead to a large number of false positives in the subway station sequences, is detection of the movement of the station's doors (figures 3.e and 3.f). Such errors are likely unavoidable given the tested trackers inability to distinguish humans from other objects of similar size and speed (without simply excluding the doors' arc from the tracked area, at least) but, much like over-detection, such tracks could be deleted in post-processing as they are distinguishable by their limited length.

TI's sensitivity to the distance of tracked objects from the camera suggests that it may be even more finely tuned if the tracking area is reduced. If the tracker can be optimized to track only specified narrow bands, it may serve as an accurate and reliable pedestrian counter using only typical surveillance footage, particularly given that the precision and matching requirements could be relaxed.

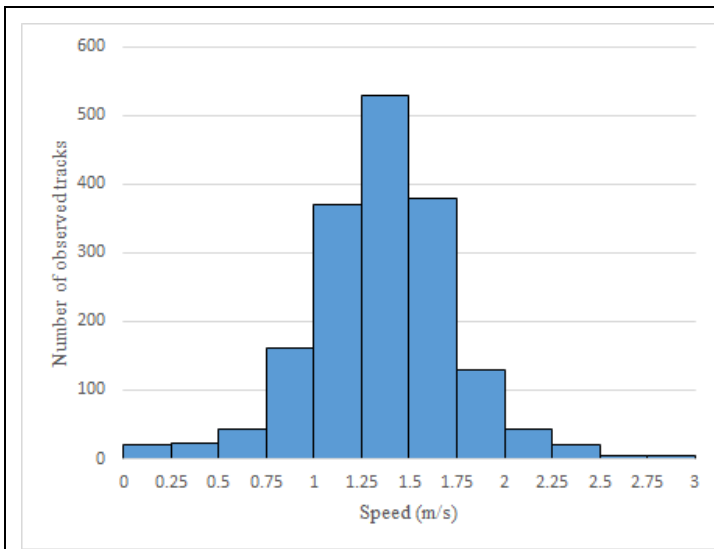
APPLICATIONS

Despite the errors which persisted after optimization, parameter calibration for specific scenes allowed improved automated extraction of flow characterization and cordon counting. Examples of such data were extracted from the full two hour video recorded in the Polytechnique corridor utilizing algorithm-optimized TI (UT was shown to provide higher quality tracks, yet TI's substantially faster tracking was selected for convenience). Said examples are presented in figure 4.

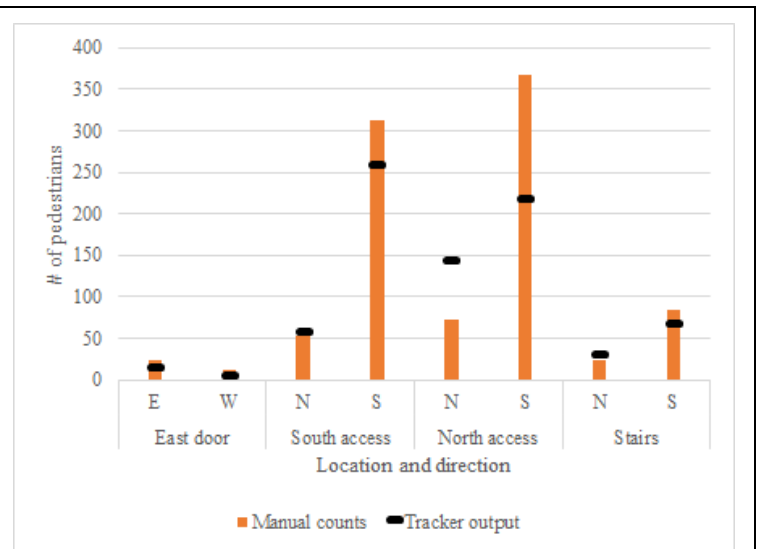
Pedestrian speed is estimated by averaging the instantaneous speed from every second (or 30 frames) of a tracks' existence. The resulting speed distribution (figure 4.a) and median speed of 1.38 m/s are comparable to results presented in (28), with outliers at lower speeds contributable to punctual density increases.

Counts (figure 4.b) are measured through the establishment of pairs of parallel lines set side-by-side; the sequence in which a track crossed these lines establishes direction. These lines were placed on four of the five accesses to the centre of the corridor (figure 4.d), excluding the hallway to the west due to the camera angle precluding vision of its entrance. The average error when compared to manual counts was 38.2 %, though there is an apparent negative correlation of accuracy with distance from the camera: error in the background/northern access was 67.6 %, whereas in the foreground/south access it was as little as 9.2 %. For any future counting applications, it is suggested that a steep camera angle and proximity be maintained to the region within which counting is required.

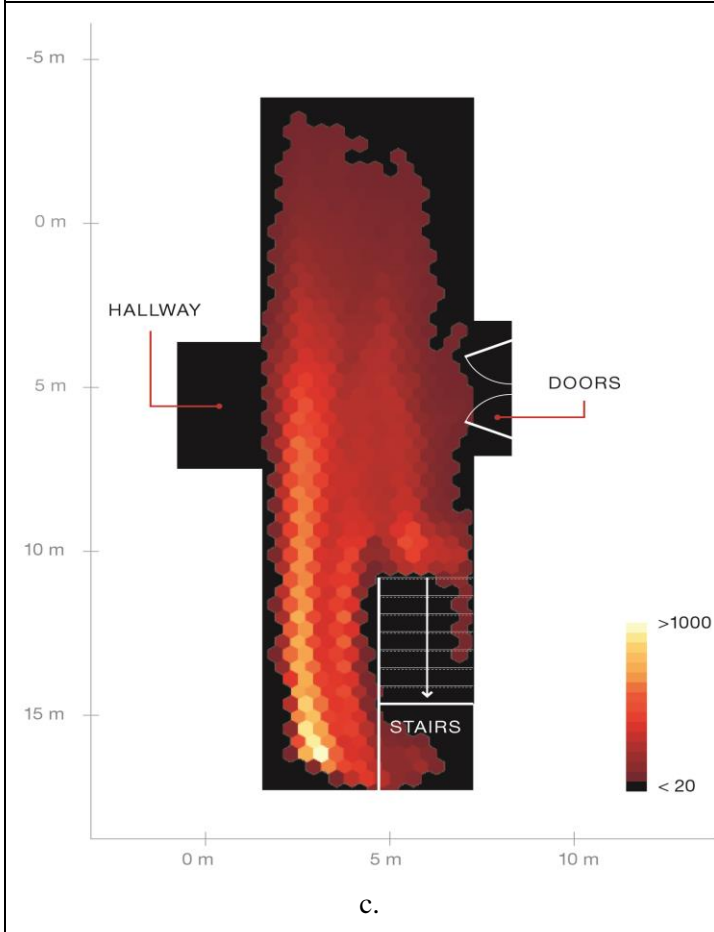
This under-performance at longer viewing distances is also observed in densities (figure 4.c) represented by a heat map of pedestrian traces (i.e. detections at each frame). Pedestrian detection decreases towards the north to an extent which is unlikely solely a result of the widening of hallway. In contrast, the more accurately tracked southern access display flow behaviour highly consistent with both the manual counts and observations (a majority of pedestrians in the scene are walking towards the tunnel access in the south and tend to keep to their right) as well as the geometry of the corridor, with two distinct primary paths leading to and from the two out-of-frame doors.



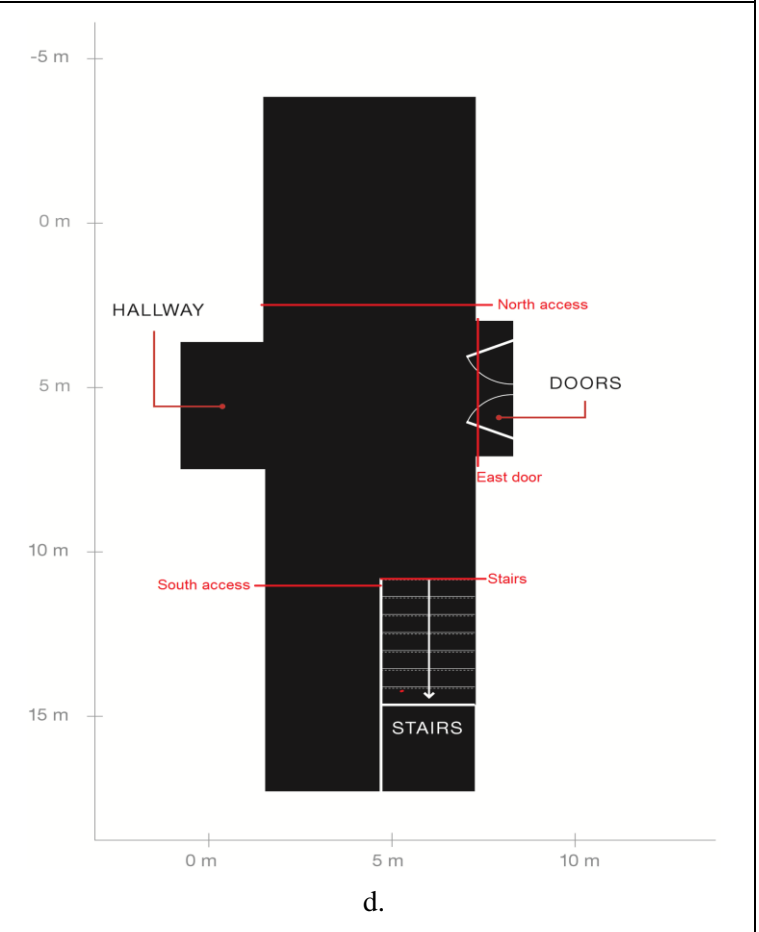
a.



b.



c.



d.

Figure 4 - Data extracted from the full video sequence of the Polytechnique corridor. a.: speed distribution of tracks; b.: tracker and manual cordon counts of pedestrians traversing the lines defined in d.; c.: heat map of tracks in the tracked space, divided into 0.4 m hexes. In both c. and d., the figures are oriented so that north is at the top of the map.

CONCLUDING REMARKS

The improvements offered by the algorithm show promise, particularly given that the sequences tested were chosen specifically for their complexity. An obvious next step will be the comparison of optimized TI and UT with other trackers on a same dataset. More interesting, however, will be the use of the developed optimization algorithm with other candidate trackers; indeed, the modularity of the optimization code greatly facilitates the swapping of one tracker for another, the only fundamental change being the modification of the neighbor-solution generator. It will be interesting to note whether the consistency in relative performance is a feature of sufficient optimization or solely one of the studied tracking method. To facilitate broader usage of the code on optimization, flow characterization and counting, it is available upon request and will soon be released as an open source project (the optimized parameters are also available on request and will be made available on a public website).

In addition to the state generation and functional form of the optimization function, the starting set of parameters greatly affects the search process in the evolutionary algorithms. Here we experimented with three different starting points i.e. default set of parameters suggested by the trackers' authors, random set of parameter values, and parameter set suggested by an expert after evaluation of the scenes. Search based on the last set of starting parameters greatly outperformed the other two. This is probably because of a weaker state generation function. Note that in our implementation, the generation function is not taking into account any correlations that may exist between parameters. Furthermore, the highly nonlinear nature and multidimensionality of the search space may necessitated a good starting point.

One final option afforded by the use of an evolutionary algorithm is the hybridization of trackers. By both combining the outputs of two or more trackers and sufficiently parameterizing the merging, addition, and/or interpolation of tracks, the existing algorithm could be adapted to "learn" tracking methods superior to the sum of their parts.

The paper reported two applications in the real case studies. In first we demonstrated the flow characterization using heat maps and speeds distribution from the pedestrian trajectories extracted by optimized trackers. While in the second application we demonstrated the extraction of cordon counts for pedestrians. The validation process gives us the confidence to use the code developed in this paper as a generic framework for automated data collection on pedestrian movement.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the funding of NSERC (Discovery Grant) and Polytechnique Montreal for this study, and to thank the Société de Transport de Montréal for access to their facilities and the permission to use the data extracted therein, as well as Kian Ettehadieh and my colleagues for their contributions to the data collection and extraction.

REFERENCES

- (1) Mehran, R., Oyama, A., & Shah, M. (2009, June). Abnormal crowd behavior detection using social force model. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009*, (pp. 935-942). IEEE.
- (2) Forsyth, D. A., & Ponce, J. (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.
- (3) Hoogendoorn, S. P., Daamen, W., & Bovy, P. H. (2003, January). Extracting microscopic pedestrian characteristics from video data. In *Transportation Research Board 2003 Annual Meeting, CD-ROM, Paper (No. 477)*

- (4) Johansson, A., Helbing, D., Al-Abideed, H. Z., & Al-Bosta, S. (2008). From crowd dynamics to crowd safety: a video-based analysis. *Advances in Complex Systems*, 11(04), pp. 497-527
- (5) Bernardin, K., Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008
- (6) Fan, X., Mittal, S., Prasad, T., Saurabh, S., & Shin, H. (2013). Pedestrian detection and tracking using deformable part models and Kalman filtering. *Journal of Computer-Mediated Communication*, 10. Pp. 960-966
- (7) Ellis, A., Shahrokni, A., & Ferryman, J. M. (2009, December). Pets2009 and winter-pets 2009 results: A combined evaluation. In *Twelfth IEEE International Workshop on. Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009* IEEE. pp. 1-8
- (8) Trucco, E., & Plakas, K. (2006). Video tracking: a concise survey. *Oceanic Engineering, IEEE Journal of*, 31(2). pp. 520-529
- (9) Saunier, N., & Sayed, T. (2006, June). A feature-based tracking algorithm for vehicles in intersections. In *The 3rd Canadian Conference on Computer and Robot Vision, 2006*. (pp.59-59). IEEE.
- (10) Rabaud, V., & Belongie, S. (2006, June). Counting crowded moving objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006* (Vol. 1, pp. 705-711). IEEE.
- (11) Ismail, K., Sayed, T., & Saunier, N. (2010). Automated analysis of pedestrian-vehicle conflicts: a context for before-and-after studies. *Transportation Research Record: Journal of the Transportation Research Board*, 2198(1). pp. 52-64.
- (12) Khanloo, B. Y. S., Stefanus, F., Ranjibar, M., Li, Z. N., Saunier, N., Sayed, T., & Mori, G. (2012). A large margin framework for single camera offline tracking with hybrid cues. *Computer Vision and Image Understanding*, 116(6), pp. 676-689.
- (13) Pérez, O., Patricio, M. A., Garcia., J., & Molina, J. M. (2006). Improving the segmentation stage of a pedestrian tracking video-based system by means of evolution strategies. In *Applications of Evolutionary Computing* (pp. 438-449). Springer Berlin Heidelberg.
- (14) Berclaz, J., Fleuret, F., Teretken, E., & Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9). pp. 1806-1819.
- (15) Nguyen, H. T., & Bhanu, B. (2012, September). Real-time pedestrian tracking with bacterial foraging optimization. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSSS), 2012* (pp. 37-42). IEEE.
- (16) Jodoin, J. P., Bilodeau, G. A., & Saunier, N. (2013). Urban Tracker: multiple object tracking in urban mixed traffic.
- (17) Guan, Y., Chen, X., Wu, Y., & Yang, D., (2013, June). An improved particle filter approach for real-time pedestrian tracking in surveillance video. In *2013 International Conference on Information Science and Technology Applications (ICISTA-2013)*. Atlantic Press.
- (18) Sidla, O., Lypetsky, Y., Brandle, N., & Seer, S. (2006, November). Pedestrian detection and tracking for counting applications in crowded situations. In *IEEE International Conference on Video and Signal Based Surveillance, 2006. AVSS'06* (pp. 70-70). IEEE.
- (19) Singh, V.K., Wu, B., & Nevatia, R. (2008, January). Pedestrian tracking by associating tracklets using detection residuals. In *IEEE Workshop on Motion and video Computing, 2008*. (pp. 1-8). IEEE.
- (20) Ali, I., & Dailey, M. N. (2009, January). Multiple human tracking in high-density crowds. In *Advanced Concepts for Intelligent Vision Systems* (pp. 540-549). Springer Berlin Heidelberg.
- (21) Jiang, Z., Huynh, D. Q., Moran, W., Chalia, S., & Spadaccini, N. (2010, December). Multiple pedestrian tracking using colour and motion models. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2010* (pp. 328-334). IEEE.

- (22) Andriyenko, A., & Schindler, K. (2011, June). Multi-target tracking by continuous energy minimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011* (pp. 1265-1272). IEEE.
- (23) Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., & Xu, L. Q. (2008). Crowd Analysis: a survey. *Machine Vision and Applications, 19*(5-6), pp. 345-357
- (24) Lucas, B.D., & Kanade, T. (1981, August). An iterative image registration technique with an application to stereo vision. In *International Joint Conferences on Artificial Intelligence* (Vol. 81, pp. 674-679).
- (25) Tomasi, C., & Kanade, T. (1991). Detection and tracking of point features. Pittsburgh: School of Computer Science, Carnegie Mellon Univ.
- (26) Barnich, O., & Van Droogenbroeck, M. (2011). ViBe: a universal background subtraction algorithm for video sequences. In *IEEE Transactions on Image Processing, 20*(6), pp. 1709-1724.
- (27) Ross, S. M. (1997). Simulation, statistical modeling and decision science. *Harcourt Academic Press*.
- (28) Daamen, W., & Hoogendoorn, S. P. (2003). Experimental research of pedestrian walking behavior. *Transportation Research Record: Journal of the Transportation Research Board, 1828*(1), pp.20-30.