Long-term tracking of multiple interacting pedestrians using a single camera

Mogomotsi Keaikitse*, Willie Brink† and Natasha Govender*
*Modelling and Digital Sciences
Council for Scientific and Industrial Research
Pretoria, South Africa
†Department of Mathematical Sciences
Stellenbosch University
Stellenbosch, South Africa

***Abstract*—Detection and tracking are important components of many computer vision applications including automated surveillance. Object detection should overcome challenges such as changes in object appearances, illumination, and shadows. In our system, Gaussian mixture models are used for background subtraction to detect moving objects. Tracking is challenging because measurements from the object detection stage are not labelled and could originate from false targets. Our system uses multiple hypotheses tracking to solve the measurement origin problem. Practical long-term object tracking should have re-identification capabilities to deal with challenges arising from tracking failure and occlusions. To this end, each tracked object is assigned a one-class support vector machine (OCSVM), which learns the appearance model of that object. The OCSVM is trained online using HSV colour features. As a result, objects that were occluded or left the scene can be re-identified and their tracks extended. Standard, publicly available data sets are used to test the system.**

I. Introduction

Closed circuit cameras are becoming widespread and prevalent in cities and towns around the world, indicating that surveillance is an important issue. This increase is not only driven by commercial institutions like banks and airports, but also by governments through law enforcement departments. As the cost of these cameras decreases, the labour cost required to monitor these systems is increasing [1]. Meanwhile, the volume of video recordings generated by these systems makes it impossible to monitor every frame. In fact, most of the video recordings are used mainly as forensic evidence, being called upon to verify the facts after an event has occurred [1]. Moreover, there are situations of targeted monitoring where operators decide to pay close attention to a camera feed based on the appearances of pedestrians, rather than their behaviour [2].

The monitoring of surveillance systems calls for a scientific solution, which is offered by computer vision in the form of active surveillance. Active surveillance "attempts to detect, recognize and track certain objects from image sequences, and more generally, to understand and describe object behaviour" [3]. Thus, the ultimate goal is to automate the entire surveillance process. This technology has applications in diverse areas including access control, flux statistics and congestion analysis, and anomaly detection and subsequent alerting of personnel. These are high level functions which involve the description and understanding of object behaviours. The low level functions required for these capabilities are modelling of environments, object detection, classification, recognition and tracking, and the retrieval and fusion of data from multiple cameras.

II. Background

Collins et al. [1] implemented one of the most complete automated surveillance systems. It uses multiple, different sensors such as video and thermal cameras to achieve cooperative tracking. Moreover, their system distinguishes different types of objects like people, groups of people and cars. Another state-of-the-art system is the Knight system by Shah et al. [4] which can detect, track and categorize objects such as people and vehicles in an environment covered by multiple cameras. It also flags abnormal events such as a person in danger and presents a summary in terms of key frames and a textual description of observed activities. This summary is presented to a human operator for final analysis and decision making.

Our goal is to design and implement a system that can detect and track multiple interacting pedestrians using a single static camera. A review of complete systems pointed out challenges that we must solve and issues that we must take into account in order to realize our system. Firstly, background subtraction does not work in crowded scenes. Secondly, tracking algorithms can fail and result in fragmented tracks. Therefore, a method must be devised to connect the fragments into complete tracks. Thirdly, a data association method is required to assign detected objects to tracks. Finally, we should explicitly detect and handle merge and split events.

A. Object detection

Object detection is an important first stage of a surveillance system because it focuses the attention of subsequent stages such as tracking and classification on dynamic regions of the image and scene. Techniques for object detection may be classified as either background subtraction [5], optical flow [6] or machine learning [7]. Background subtraction and optical flow methods rely on the motion of objects to detect

them. The goal of background subtraction is to maintain an image that is representative of the scene covered by a camera. Optical flow methods, particularly dense flow methods, can be computationally expensive and thus not suitable for real-time systems [1]. Machine learning approaches to object detection learn the generic appearance and shape of objects, for them to be detected in images and videos [8]. Most of these methods must be trained off-line using large labelled data sets. They do not adapt to the changes in the appearance of objects as it is not possible to learn all the appearances of all the objects in a class. Moreover, it is especially difficult to make viewpoint or scale-invariant models. Algorithms have been proposed to learn the appearance of objects online but they rely on robust tracking and/or selective updating of the models [7]. This is a drawback because incorrectly labelled samples can corrupt the learnt models. As a result we use background subtraction in our system to detect moving objects.

Background subtraction should handle challenges such as gradual and sudden changes in illumination, shadows and dynamic background objects like escalators. Wren et al. [9] model each pixel with a single Gaussian distribution to allow for small variations. This type of model cannot handle multi-modal events such as waving trees or flickering monitors which occur in uncontrolled environments [5]. Grimson et al. [5] extend the model by modelling each pixel as a mixture of K Gaussian distributions, where K is fixed and is the same for all pixels. The algorithm relies on the assumption that the background is visible more frequently than any foreground object and that it has modes with relatively narrow variances [10].

The major drawbacks of the model by Grimson et al. are the initialization and slow stabilization of the parameters [10]. Moreover, the number of components in a mixture is the same and fixed for all pixels. Zivkovic [11] proposes an improvement that determines at runtime the optimal number of Gaussian distributions required to model the values of each pixel, in addition to estimating the parameters of each distribution in the mixture. Another drawback is that the noise in the images is assumed to have a Gaussian distribution. A viable solution is to model the variations in the intensity of a pixel using adaptive kernel density estimation [12]. Algorithms in this class estimate the density function directly from the data without making any assumptions about the underlying distribution.

In this paper, moving object detection is performed using the improved mixture of Gaussian distributions algorithm as outlined by KaewTraKulPong and Bowden [10]. Their improvements to the original method by Grimson et al. [5] solve the issues related to the initialization and stabilization of the model parameters.

B. Tracking

Tracking is a crucial component for automated surveillance. It seeks to consistently label objects of interest in every frame of a video sequence. Tracking can be a complex problem as a result of noise, cluttered environments, illumination changes in the scene, object and camera motions, non-rigid and articulated objects, and occlusions. Requiring that a tracking system runs in real-time presents a further challenge, as this renders many solutions infeasible due to high computational costs [13]. Tracking may also require the use of multiple cameras either to handle occlusions or to cover large areas. In this case the challenge is reconciling the different identities of an object as seen from the fields of view of different cameras. In our case a single static camera is used.

Yilmaz et al. [14] classify tracking algorithms into point, kernel and silhouette tracking methods. Point tracking methods include the Kalman and particle filters. Our goal is to track multiple interacting objects. Tracking silhouettes is not ideal as they are sensitive to occlusions. Moreover, they provide more detail than is required for our purpose. Kernel-based methods such as tracking-by-detection [15] and the mean-shift tracker [16] require an external method for initialization. This can be provided by an object detection method such as background subtraction [5]. The next issue is that of initializing the search. Cominaciu et al. [16] start searching where the pattern was found on the previous time step. However, they suggest incorporating a filtering algorithm to better predict the starting position. The appearance of objects may change due to variations in illumination and viewpoint, and the non-rigidity of objects. Kernel-based tracking methods must account for these changes. One approach is to adapt the appearance of objects models. An example is the mean-shift tracker [16] which considers the current appearance of the tracked object as the target appearance. This adapted template could be corrupted because either of background regions in the template or occlusions.

Recent approaches use machine learning methods to learn the appearance of objects online [15]. Even in this case, a strategy must be devised to search for regions in the next frame that are confidently explained by the classifiers. An alternative approach is to pair online learning methods with particle filter methods to predict prospective object locations [17]. We also note that online learning of object-specific appearances may corrupt the learnt model if incorrectly labelled samples are used.

We use point tracking methods, particularly filtering methods, to track pedestrians. Filtering methods assume a one-to-one relationship between measurements and tracks. However, measurements from the object detection stage are not labelled and could be from valid objects, false alarms or clutter. In the case of multiple target tracking the measurements could also be from new targets. Data association is required to solve this measurements origination problem.

The simplest data association algorithm is the nearest neighbour tracker which updates a track with the measurement that is closest to the predicted state of the track [18]. This tracker may result in one track stealing the measurement of another especially when the targets are close together. An improvement is the global nearest neighbour (GNN) method which minimizes the sum of the distances between predicted states of the tracks and measurements [18]. GNN works well

when there is no clutter or track contention, and it cannot handle the appearance and disappearance of objects.

The joint probabilistic data association (JPDA) filter is more robust to clutter and track contention [19]. It is the extension of the probabilistic data association filter to multiple target tracking. The JPDA filter assumption is that the target may not have generated the closest measurement. As a result, the average of the measurements is used to update the state of the target. One drawback of JPDA is that the tracks of closely spaced targets tend to drift towards each other because the same subset of measurements is used to update both targets [20]. Also, the number of tracked objects must be known and fixed.

JPDA introduces the concept of the probability of track-measurement association to multiple target tacking. This concept is crucial to multiple hypothesis tracking (MHT) which is a deferred logic method [21]. MHT is an exhaustive method for enumerating all possible track-to-measurement associations. Ultimately, an optimal set of disjoint tracks, referred to as a hypothesis, must be retained. Two approaches to MHT are the hypothesis-oriented MHT [21] and the track-oriented MHT [22]. The original hypothesis-oriented MHT yields joint probabilities of measurement-to-track association hypotheses. The probabilities of individual tracks may then be obtained by marginalization. It is a top-down approach and the reverse of track-oriented MHT.

This work uses multiple hypothesis tracking for data association. It implicitly provides facilities for track initialization, continuation and termination [23]. It also explicitly models both spurious measurements and constraints on measurements. The MHT approach is memory and computation intensive but techniques such as gating and track clustering are available to improve the situation.

C. Learning object appearances

Standard tracking approaches like mean-shift [16] and the Kalman filter assume that the object of interest is never completely occluded. This assumption and unsuitable motion models can result in tracking failure. These methods do not directly address what happens after tracking failure. Instead, new tracks are initialized after tracking failure or when objects reappear. In this paper machine learning algorithms are used to learn object-specific appearances which are then used to uniquely re-identify objects when they reappear or after tracking failure.

At least three aspects are essential for learning the appearance model of an object for re-identification purposes. First, the features used to represent the appearance of objects must be discriminative. In the case of recognizing people, biometric features such as the face, iris and gait could be used to re-identify people, but most surveillance video have low resolution or are difficult to segment. As a result it is necessary to model the global appearance of each object. This leads to the second aspect which is that models must be learned online because discriminative appearances of tracked objects cannot be known in advance.

Lastly, a strategy must be devised to decide which samples to use to update the model. Each update can introduce errors which can lead to the classifier not learning the appearance of the intended object [13]. The errors may be due to the inaccuracies in segmenting the object. Moreover, some of the background will be treated as part of the foreground no matter how tight the bounding box.

III. System components

This section provides more detail on the major components of the system.

A. Track-oriented MHT

The major component of track-oriented MHT is the calculation of the probability that a measurement was generated by a track with which it is associated. The preferred approach is to use Bayes' theorem and combinatorial analysis of the data association problem to derive the joint probability of a hypothesis. The result is the probabilistic expression presented by Reid [21]. For the purpose of implementation, it is easier to use the mathematically equivalent log-likelihood ratio proposed by Sittler [24]:

$$
\begin{aligned}
L_i = & \ln \frac{\lambda_0 \lambda_s}{\lambda_N (\lambda_s + \frac{1}{\tau_0})} + \sum_{k=2}^{m_i} \ln \frac{\lambda_s e^{-(\lambda_s + 1/\tau_0)\, t_k}}{\lambda_N |2\pi(\epsilon_k^2 + t_k\, \mathbf{S}_{ik})|^{\frac{1}{2}}} \\
& - \frac{1}{2} \sum_{k=2}^{m_i} (\mathbf{z}_{ik} - \mathbf{H}_{ik} \hat{\mathbf{x}}_{ik})^T (\epsilon_k + t_k \mathbf{S}_{ik})^{-1} (\mathbf{z}_{ik} - \mathbf{H}_k \hat{\mathbf{x}}_{ik}) \quad (1) \\
& + \ln \left[\frac{\lambda_s e^{-(\lambda_s + 1/\tau_0)(T - v_i)} + \frac{1}{\tau_0}}{\lambda_s + \frac{1}{\tau_0}} \right],
\end{aligned}
$$

where t_k is the time between the $(k-1)$th and the kth measurements (the last received and current measurements). The parameter m_i is the number of measurements associated with track i and ϵ_k is chosen such that

$$
\left[\ln \frac{\lambda_s e^{-(\lambda_s + 1/\tau_0)\, t_k}}{\lambda_N |2\pi(\epsilon_k^2 + t_k\, \mathbf{S}_{ik})|^{\frac{1}{2}}} \right]_{t_k = 0} > 0. \quad (2)
$$

An assumption made in deriving the score is that the numbers of new objects and false alarms may be modelled using Poisson distributions with parameters λ_0 and λ_N, respectively. λ_0 is the average number of new objects per unit time per unit area of the region under surveillance. Similarly, λ_N is the number of false measurements per unit time per unit area. The observations on a single object are assumed to follow a Poisson process with the average rate λ_s.

Each object is assumed to persist independently through a length of time that has an exponential distribution with time constant τ_0. This is the mean track length. The kth measurement associated with track i is $\mathbf{z}_{ik}$. The predicted state of the track is $\hat{\mathbf{x}}_{ik}$ and H_{ik} is the Kalman filer measurement matrix. $\mathbf{S}_{ik}$ is the Kalman filter innovation covariance matrix. T is the current time and v_i is the last time a measurement associated with track i was received. A detailed analysis of this formula and its application to tracking may be found in [24] and [25]. We note that tracks with larger log-likelihood ratios are preferred.

B. One-class SVM

One-class support vector machines are used to learn the appearance of objects of interest. In this case only positive training samples represented in the hue-saturation-value feature space are used. The goal is to obtain a function $f(\mathbf{x})$ that demarcates a region $\mathcal{S} \subseteq \mathcal{R}^D$ in the input space representative of the training data such that

$$f(\mathbf{x}) \geq 0, \text{ if } \mathbf{x} \in \mathcal{S}, \qquad f(\mathbf{x}) < 0, \text{ if } \mathbf{x} \notin \mathcal{S}.$$

Kivinen et al. [26] derive the optimization problem:

$$\min_{\mathbf{w},b} \frac{\nu}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\left[\max(0, b - \mathbf{w}\cdot\phi(\mathbf{x}_i)) - Nb\nu\right].$$

The function $\phi(\mathbf{x})$ transforms the input data $\mathbf{x}$ from $\mathcal{R}^D$ into $\mathcal{R}^P$ where a hyperplane optimally separates the data from the origin. The parameters of the hyperplane are the offset from the origin b and the normal $\mathbf{w} = \{w_1, w_2, \ldots, w_P\}$. N is the number of training samples and $\nu \in (0, 1)$ is a parameter chosen by the user.

Training a support vector machine means solving this optimization problem, which must be done online. Online learning is achieved using stochastic gradient descent (SGD) which uses a single sample at each iteration. This is in contrast to Newton gradient descent which uses all samples at each iteration. The goal is to find a sequence of parameters $\{(\mathbf{w}_n, b_n)\}_{n=1}^{N+1}$, hence a sequence of decision functions $\mathbf{f} = \{f_1, f_2, \ldots, f_{N+1}\}$. The initial parameter set $(\mathbf{w}_1, b_1)$ is arbitrary and $(\mathbf{w}_n, b_n)$, $n > 1$, is obtained after observing the $(n-1)$th training sample, that is

$$\mathbf{w}_n = \sum_{i=1}^{n-1} \alpha_i^n \phi(\mathbf{x}_i), \tag{3}$$

$$f_n(\mathbf{x}) = \mathbf{w}_n \cdot \phi(\mathbf{x}_n) - b_n. \tag{4}$$

A superscript is used for the Lagrange multipliers α_i^n, $i = 1, 2, \ldots, n-1$, to emphasize that they evolve as more samples are used to train the SVM. Application of SGD yields the following set of iterative equations:

$$\alpha_i^{n+1} = (1 - \nu\eta_n)\alpha_i^n, \quad i = 1, 2, \ldots, n, \tag{5}$$

$$\alpha_{n+1}^{n+1} = \begin{cases} \eta_n, & \text{if } f_n(\mathbf{x}_n) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

$$b_{n+1} = \begin{cases} b_n - (1-\nu)\eta_n, & \text{if } f_n(\mathbf{x}_n) < 0, \\ b_n + \nu\eta_n, & \text{otherwise.} \end{cases} \tag{7}$$

IV. System integration

We have identified four major components of the system, which are the mixture of Gaussians to background subtraction for object detection, MHT for data association, and the Kalman filter and OCSVM for online learning of object appearances. For the purpose of integrating these components into a complete system, we introduce the pedestrian, the integer-programming problem-solver (IPPSolver) and the single camera system (SCS) components. The pedestrian component represents what we are tracking, as well as its track. The IPPSolver converts the MHT problem into an integer-programming problem and then solves it[25]. The SCS coordinates the interaction between the various classes.

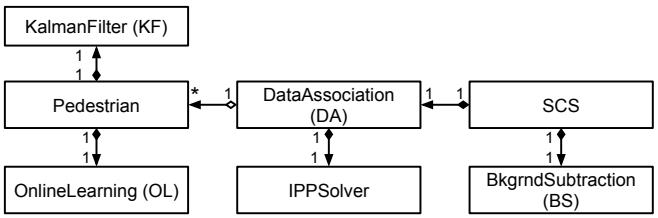


Fig. 1: The interaction of the system components.

Figure 1 shows the high-level interactions between the components. It indicates that the SCS object must have one and only one of the each of the DA and BS objects as data member. The same holds for the relationship between the Pedestrian object and the KF and OL objects. This is the case because the Pedestrian object represents what is being tracked as modelled by the OL object, as well as the track which is obtained using the KF object. The DA object contains a list of Pedestrians. The DA object contains a list of Pedestrians.

V. Results

We consider scenarios that test the ability of the system to handle various tracking problems. These include object re-identification, tracking two people walking side-by-side and crossing paths. We also provide examples of tracking failures in the system. The datasets collected by Baltieri et al. [27], [28] and Rasid and Suandi [29] are used. These videos are chosen because they meet the assumptions of the system which are that each pedestrian occupies a small fraction of the frame, the camera looks down on the pedestrians and the only moving objects are pedestrians.

We use the Jaccard similarity coefficient (JSC) to measure the similarity between the system output S and ground truth G bounding boxes:

$$J(S, G) = \frac{|S \cap G|}{|S \cup G|}, \tag{8}$$

where $|A|$ denotes the area of the bounding box A. The track length and normalized mean squared error (NMSE) are also used to measure the performance of the system. A person is said to be correctly tracked in a given frame if JSC ≥ 0.65. The ground truth was generated using the MATLAB toolbox developed by Dollàr [30].

For each scenario, the first experiment that we perform is determining the optimal parameter T for the background subtraction method. For a given pixel, N of the M sorted densities in the mixture with weights $w_i, i = 1, 2, \ldots, M$ represent the background if

$$\sum_{j=1}^{N} w_i \leq T \tag{9}$$

where T is the minimum fraction of the data that should account for the background. This is to ensure that the bounding

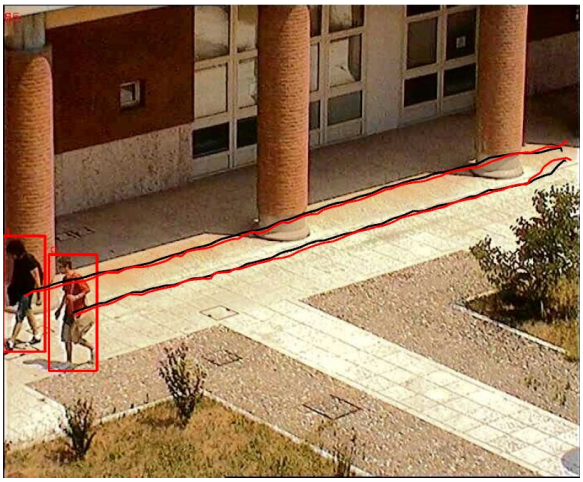


Fig. 2: Tracking results for scenario 1. The system generated tracks in black and the corresponding ground truth tracks in red.

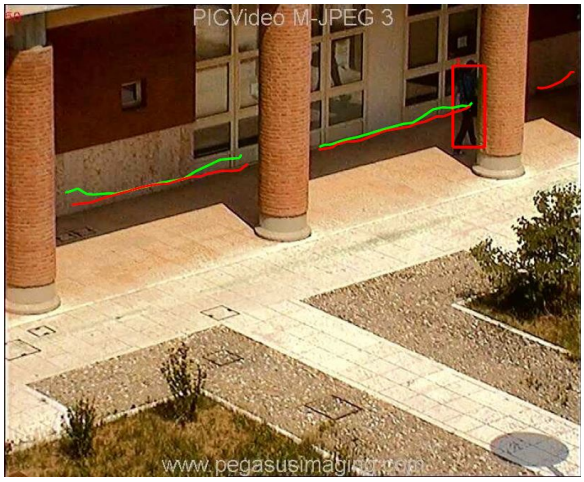


Fig. 3: Tracking results for scenario 2. Green indicates the system generated track, and the ground truth track is shown in red.

boxes enclose as much of the pedestrians as possible. All other parameters are set empirically.

A. Scenario 1: Two people walking side-by-side

The first scenario tests the ability of the system to track two people walking side-by-side. The first experiment that is performed is to determine the optimal parameter T for the background subtraction method. The value $T = 0.65$ is used in subsequent experiments because it results in the highest true positive rate. In the second experiment, the system is used to track two people as they walk across the scene. Figure 2 shows the tracks generated by the system in black as well as the respective ground truth tracks in red. Pedestrian 1 was correctly tracked for 78 of the 79 frames he was in the scene with an NMSE of 3.79 pixels. Pedestrian 9 was correctly tracked for 75 of the 77 frames he was in the scene with NMSE of 4.55 pixels.

B. Scenario 2: Re-identification

In this section we test the re-identification ability of the system. The first experiment that we perform is to determine the optimal value of the background subtraction parameter T. The values of $T \in (0.5, 0.75)$ yield the same true positive rate. However, the values of $T \in (0.6, 0.75)$ yield better precision rates. As a result, $T = 0.65$ was used in the experiments.

In the second experiment, a tracked pedestrian is completely occluded by a pillar and the system manages to re-identify him when he reappears and extend his track. Figure 3 shows the track generated by the system in green as well as the ground truth track in red. The pedestrian then disappears behind the third pillar (on the right) and is not re-identified when he reappears. This is due to poor illumination in that region which changes the appearance of the pedestrian. This highlights issue that an object can only be re-identified if its current appearance is similar to one of the previously seen appearances. The system successfully tracked the pedestrian in 38 of the 48 frames with an NMSE of 9.72 pixels which is slightly higher than the NMSEs in the first scenario.

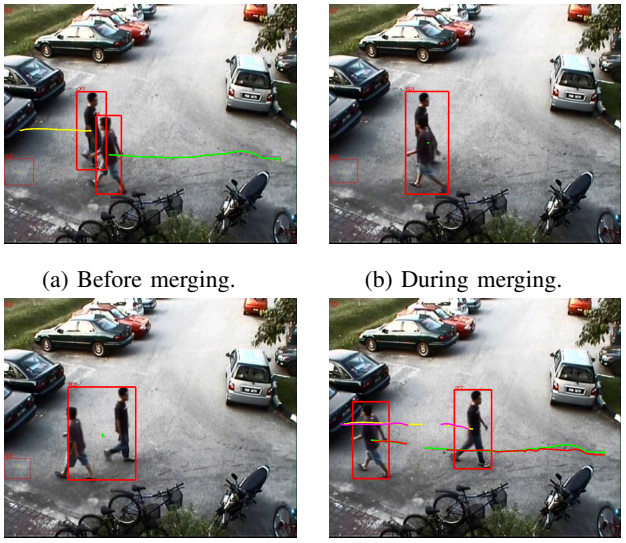
C. Scenario 3: Crossing paths

Next we test the ability of the system to track two pedestrians whose paths cross. The first experiment is to determine the optimal value for the background subtraction parameter T. The values of $T \in (0.55, 0.70)$ yield the same results for all measures. A value of $T = 0.65$ is used in the experiments to match those used in previous experiments.

The second experiment demonstrates how the system handles merging and splitting tracks. Figure 4a shows the tracks three frames before the merger takes place. The pedestrians are still tracked individually even though they are not detected as separate entities in those frames. This is because a track that is not associated with a measurement in a given number of consecutive frames, in this case three, is assumed to have left the scene and is deleted. This is used to manage the number of hypotheses in the MHT tracking algorithm.

Figure 4b shows that the merged group is being tracked as a single object. This track was created when the two pedestrians first merged but had to be supported by additional measurements to ensure that it is not a false track. Figure 4c shows that the group is still tracked as a single object even though the pedestrians have finally separated. It will only be deleted after three consecutive missed detections. At this point the splitting event has been detected and the measurements are used for re-identification.

The pedestrians are successfully re-identified as shown in Figure 4d. where the system generated tracks and the associated ground truth tracks are shown in red and magenta. The pedestrian on the left was correctly tracked for 51 of the 53 frames he was in the scene with an NMSE of 8.89 pixels. The pedestrian on the right was correctly tracked for 30 of the 31 frames he was in the scene with an NMSE of 6.91 pixels. Note that our system tracks overshoot the ground



(a) Before merging. (b) During merging.

(c) After splitting. (d) System and true tracks.

Fig. 4: Tracking results of scenario 3, of two people whose paths cross.

truth tracks substantially when going into the merger. This is because tracks are retained for at most three consecutive frames even though the tracked objects are not detected in those frames.

D. Scenario 4: Re-identification failure

This section demonstrates a scenario where the system fails because of the similar appearance of pedestrians. In Figure 5 the pedestrian on the right was occluded by the middle pillar when the pedestrian on the left entered the scene. This meant that the track of the leading pedestrian could be used for re-identification. As a result, the lagging pedestrian was mistaken for the leading pedestrian because their appearances are similar. This can be resolved by using different or additional features that preserve the shape information of the pedestrian such as the histogram of orientation gradients. Re-identification can also be improved by using a rule that a pedestrian cannot move move faster than some bound and thus cannot jump significantly in position from one frame to the next.

VI. Conclusions

The goal of this paper was to design and implement a system that can detect and track multiple interacting pedestrians over a prolonged period of time. Such a system is necessary because of the challenges emerging from the growing number of closed circuit cameras, which expensive to monitor. Our focus is only part of a complete system that would include understanding and describing the behaviour of pedestrians. The components of the system that were identified are object detection, motion estimation using filters, data association, and learning the appearance of pedestrians.

Gaussian mixture models were used to detect moving objects. Long-term tracking is achieved using data association, filtering and online learning. Data association is realized using multiple hypothesis tracking. One-class support vector machines are used to learn the appearance of tracked objects and re-identify those object once tracking fails. The components must interact to produce a complete system which required the introduction of additional components.

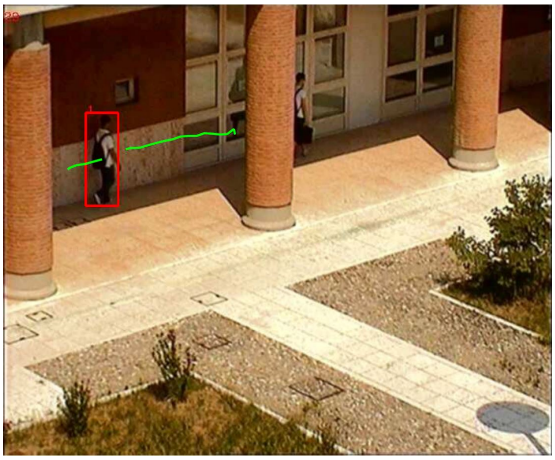


Fig. 5: Tracking results for scenario 4. The system fails because the two pedestrians have similar appearances.

The system performs well as can be judged visually from the output. This observation is also supported by quantitative measures. The fact that a similar set of parameters was used for all scenarios indicates that the system is fairly robust.

There are improvements that can be made to the system. The first improvement would be to use a number of features to represent objects. Colour features are discriminative but, as used in this paper, do not incorporate shape. This was highlighted in one of the experiments when one pedestrian was mistaken for another with a similar colour histogram. Complementing the colour histogram with a histogram of oriented gradients may solve the problem.

Re-identification can also be improved by using bounds on the perceived speeds of objects, as mentioned in the previous section. Another improvement would be to extend the system to handle multiple merges and splits. Currently the system can handle a single merge and split event at the same time. All that is required is a data structure to keep track of the merges and splits.

Finally, a set of hypotheses in the same tree are maintained separately when transforming the track-oriented MHT problem into an integer programming problem. This simplifies the implementation but increases the memory requirement. We could optimize the implementation by using a tree structure.

References

[1] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A system for video surveillance and monitoring," Carnegie Mellon University, Technical Report CMU-RI-TR-00-12, 2000.

[2] K. Macnish, "Unblinking eyes: the ethics of automated surveillance," *Ethics and Information Technology*, vol. 14, no. 2, pp. 151–167, 2012.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviour," *Transactions on Systems, Management and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.

[4] M. Shah, O. Javed, and K. Shafique, "Automated visual surveillance in realistic scenarios," *Transactions on Multimedia*, vol. 14, no. 1, pp. 30–39, 2007.

[5] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition*, 1999, pp. 246–252.

[6] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–466, 1995.

[7] H. Grabner and H. Bischof, "On-line boosting and vision," in *Computer Vision and Pattern Recognition*, 2006, pp. 260–267.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[9] C. R. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: real-time tracking of the human body," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[10] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, 2002, pp. 135–144.

[11] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *International Conference on Pattern Recognition*, 2004, pp. 28–31.

[12] A. Elgammal, R. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European Conference on Computer Vision*, 2000, pp. 751–767.

[13] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: a survey," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[14] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, p. 13, 2006.

[15] H. Grabner, C. Leister, and P. Fua, "Semi-supervised on-line boosting for robust tracking," in *European Conference on Computer Vision*. Springer, 2008, vol. 5302, pp. 234–247.

[16] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[17] Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *European Conference on Computer Vision*, 2008, pp. 678–691.

[18] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a tracking algorithm using global nearest neighbor approach," in *Computer Systems and Technologies*, 2003, pp. 290–295.

[19] Y. Bar-Shalom and X. Li, *Multitarget-multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[20] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[21] D. B. Reid, "An algorithm for tracking multiple targets," *Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[22] Y. Bar-Shalom, S. S. Blackman, and R. J. Fitzgerald, "The dimensionless score function for multiple hypothesis tracking," *Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 392–400, 2007.

[23] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's MHT algorithm and its evaluation for the purpose of visual tracking," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[24] R. W. Sittler, "An optimal data association problem in surveillance theory," *Transactions on Military Electronics*, vol. 8, no. 2, pp. 125–139, 1964.

[25] A. S. Keaikitse, "Long term tracking of multiple interacting pedestrians," Thesis, Stellenbosch University, 2014.

[26] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[27] D. Baltieri, R. Vezzani, and R. Cucchiara, "SARC3D: A new 3D body model for people tracking and re-identification," in *International Conference on Image Analysis and Processing*, 2011, pp. 197–206.

[28] D. Baltieri, R. Vezzani, and R. Cucchiara, "3DPes: 3D people dataset for surveillance and forensics," in *Joint ACM Workshop on Human Gesture and Behavior Understanding*, 2011, pp. 59–64.

[29] L. N. Rasid and S. A. Suandi, "Versatile object tracking standard database for security surveillance," in *International Conference on Information Science, Signal Processing and Their Applications*, 2010, pp. 782–785.

[30] P. Dollár, "Piotr's Image and Video Matlab Toolbox (PMT)," http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html, Sept 2013.