



Long-range trajectories from global and local motion representations[☆]



Eduardo M. Pereira^{*}, Jaime S. Cardoso, Ricardo Morla

INESC TEC, Campus da FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:

Received 16 September 2015

Revised 18 April 2016

Accepted 22 June 2016

Available online 25 June 2016

Keywords:

Long trajectories

Motion representations

Flow information

ABSTRACT

Motion is a fundamental cue for scene analysis and human activity understanding in videos. It can be encoded in trajectories for tracking objects and for action recognition, or in form of flow to address behavior analysis in crowded scenes. Each approach can only be applied on limited scenarios. We propose a motion-based system that represents the spatial and temporal features of the flow in terms of long-range trajectories. The novelty resides on the system formulation, its generic approach to handle scene variability and motion variations, motion integration from local and global representations, and the resulting long-range trajectories that overcome trajectory-based approach problems. We report the results and conclusions that state its pertinence on different scenarios, comparing and correlating the extracted trajectories of individual pedestrians, manually annotated. We also propose an evaluation framework and stress the diverse system characteristics that can be used for human activity tasks, namely on motion segmentation.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Nowadays, almost any public space has a CCTV (Closed Circuit Television) system installed. This has fostered the implementation of (semi-)automatic systems to interpret real-world scenes by monitoring individuals and their activities, detecting common motion patterns, and identifying unusual behaviors. The key insight for this type of system is to exploit spatiotemporal relationships among subjects and motion patterns, while retaining structural information of the scene. Underlying motion representations such as trajectories are, by nature, intuitive and useful to build up solutions for such problems. To avoid ambiguity with related work and clarify our aim, throughout the paper the term **global motion trajectories** refers to trajectories that translate typical paths of pedestrians, normally used for video scene analysis [1–3].

Long-duration trajectories offer several advantages [4] over short-range tracklets [5–7] for visual analysis tasks such as activity discovery and learning of semantic region models for event recognition. However, they imply to overcome occlusions, camera motions, and nonrigid deformations issues. In this paper, we provide a system for computing long-range global motion

trajectories based on the combination of a rich global flow representation, that accurately captures spatiotemporal continuity and transitions of motion, with a local flow energy field in terms of entropy, which measures the degree of motion variability revealing salient regions for flow topology analysis.

The type of scenarios addressed in this work are inherent to public spaces under CCTV systems. We aim to prove that the proposed system can approximate global motion trajectories for different pedestrian scenarios such as crowds, with structured and unstructured movement, and multi-tracking, with sparse and dense groups. Our contributions can be summarized as: (i) integration of local motion information, based on information theory principles, with global motion information, based on temporal integration of flow, to capture longer spatial and temporal changes in the scene; where (i.i) a new global technique for flow vector outliers removal is proposed; (i.ii) and a fine-to-coarse quantisation of flow vectors and analysis of its impact on temporal flow integration is presented; (ii) a complete and dynamic motion-based system that automatically extract long-range motion trajectories from different scenarios, that accounts with (ii.i) a re-correlation algorithm to link broken streamlines and accurately form long-range streamlines which correspond to the global motion trajectories.

The paper's outline is as follows. In Section 2, we survey the related work. Next, in Section 3, we briefly present the relevant theoretical concepts behind our system. The following Section 4, summarizes an overview about the system's foundations. A

[☆] This paper has been recommended for acceptance by M.T. Sun.

^{*} Corresponding author at: INESC TEC, Campus da FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal.

E-mail addresses: ejmp@inesctec.pt (E.M. Pereira), jaime.cardoso@inesctec.pt (J.S. Cardoso), ricardo.morla@fe.up.pt (R. Morla).

description about the main system steps is presented next, in Section 5. The experimental setup and results are reported in Section 6. Finally, we formulate the conclusions and future work in Section 7.

2. Related work

Considering various types of scenes, from dense crowd context to multi-tracking with sparse groups, the literature can be subdivided depending on the scene density and the object size. Indeed, human motion can be described at different levels from micro to macro scale. Each one implies specific motion analysis since their underlying relationships among individuals and space-context behavior differ. Normally for high density scenes and low object resolution, motion is modeled at a global level and patterns are inferred [8–10]. On the other side for scenes with a small number of objects, multi-tracking approaches [11,12] are preferred since they track objects individually and describe motion by their spatial position. The computer vision community has been addressing several research problems related to each scenario independently.

Crowded scenes present two types of categories, structured and unstructured, depending if the movement of objects are defined by physical constraints or if they move freely in any direction, respectively. Related work focuses on modeling scene structures and on recognizing the co-occurrences of crowd behaviors. For instance, Saad et al. [8] proposed a framework that implements a Lagrangian Particle Dynamics to advect a grid of particles and use them for motion interpretation in the form of physically and dynamically distinguishable motion segments. This type of approaches overcomes the lack of optical flow in capturing long-range temporal dependencies, and do not suffer from problems faced by object-tracking-based approaches. However, they do not consider spatial changes, cause time delays, and imply high computational effort.

For structured scenes, motion patterns are the most salient features that help understanding the scene [12]. Some approaches [13,11] consider the division of the video in spatiotemporal cuboids to identify prototypical motion pattern representations and variations within each one. It is not usual to extract motion trajectories from this type of conditions. However, to the best of our knowledge, there are some works [14–16] that try to approximate the extraction of motion patterns in terms of *super tracks*, but they did not measure their similarity with traditional object tracks and they did not test their approach on low density scenes.

For unstructured scenes, the concept of *coherent motion* emerges. It describes the free collective movement of individuals in groups and try to infer collective behaviors. This type of approach models the crowd dynamics focusing on individuals and interactions among them [17]. These approaches can follow two types of taxonomy: (i) macroscopic studies [18], that consider groups as a collective and homogeneous block where the individual is transformed by the group and (ii) microscopic approaches [19], which analyze groups as the composition of individual agents that interact with each other and with the environment. Macro models have statistical meaning and not physical, while micro present physical validation but are difficult to scale up to macro scale.

Scenarios with sparse and dense groups follow a single or multi-tracking approaches. Both present difficulties related to target's size, number of similar objects, and occlusions [12]. For crowded scenes, tracking-based models disregard the correlation between pedestrians in a close vicinity. Motion trajectory mechanisms can also be performed at feature level [6,7], instead of object level, by tracking interest points. However, they face critical factors to solve: selection of good tracking features, correct mapping between selected features and actions of interest, trajectory discontinuity due to inconsistent point correspondence, among others.

Our work borrows concepts from fluid dynamics to integrate macroscopic behavior, in terms of global flow dynamics, with microscopic approach, in terms of local information theory concepts. From a technical point of view, the proposed system is novel since it integrates, in a pertinent way, different concepts into the pipeline to extract long-range trajectories that approximate the global motion trajectories. From a practical perspective, its pertinence surpass most common methods which can only be applied to limited scenarios.

3. Preliminary concepts

3.1. Flow dynamics

Motion can be described by Lagrangian and Eulerian flow descriptions, which are formulated on different frames of reference and describe coherent structures of temporal dynamics in terms of trajectories. Both provide different insights about the temporal evolution of the flow and reveal either fluid-specific or site-specific flow information, respectively. The Lagrangian coordinate system implies the advection and tracking of particles injected into the flow, and permits the observation of how the flow deforms and rotates the fluid. The Eulerian approach extracts a dense flow coverage, since particles are computed at fixed positions, providing an overview over the entire flow at a specific time-step.

The Lagrangian perspective assumes the tracking of a fluid parcel located at position $\vec{\alpha} = (x, y, z)$. The parcel trajectory, $\xi(\vec{\alpha}, t)$, also called pathline, is obtained through the computation of position of parcels at latter times, $t_i \in [0, \dots, n-1]$. A specific fluid parcel is composed by the trajectory ξ , which is a dependent variable, and by the initial position $\vec{\alpha}$ and time t , which are the independent variables.

The Lagrangian velocity, V_L , is given by the partial derivative with respect to time

$$V_L(\vec{\alpha}, t) = \frac{\partial \xi(\vec{\alpha}, t)}{\partial t} \quad (1)$$

while the Eulerian velocity, V_E , is the velocity of the fluid parcel instantaneously presented at a fixed position

$$V_E(\vec{x}, t) \big|_{\vec{x} = \xi(\vec{\alpha}, t)} = V_L(\vec{\alpha}, t) \quad (2)$$

where \vec{x} is the position, $\vec{\alpha}$ is the initial position, V_E is the dependent variable, \vec{x} and time t are the independent ones.

The flow field generated by human motion can be treated as an aperiodic dynamical system, where a grid of particles is advected. The particle evolution provides a set of dense trajectories that relates initial particles position to their position at a later time.

In a time dependent vector field there are four types of characteristic curves: streamlines, pathlines, streaklines, and timelines. For unsteady flows, directions of flow depends on time as well as on position, therefore those representations are different [20]. In this work, we explore the complementary streaklines and streamlines representations.

3.1.1. Vector field representation and advection

A grid of particles is overlaid on the flow field. The scene's motion is quantified by particles' movement driven by dense optical flow. This advection process considers a video represented by a 3-dimensional array $W \times H \times T$, where T is the number of frames, W frame's width, and H frame's height, and by the corresponding optical flow $(u_w(t), v_h(t))$, where $w \in [1, W]$, $h \in [1, H]$, and $t \in [1, T-1]$. The particle position $(x_w(t), y_h(t))$ at grid point (w, h) at time t is achieved by solving

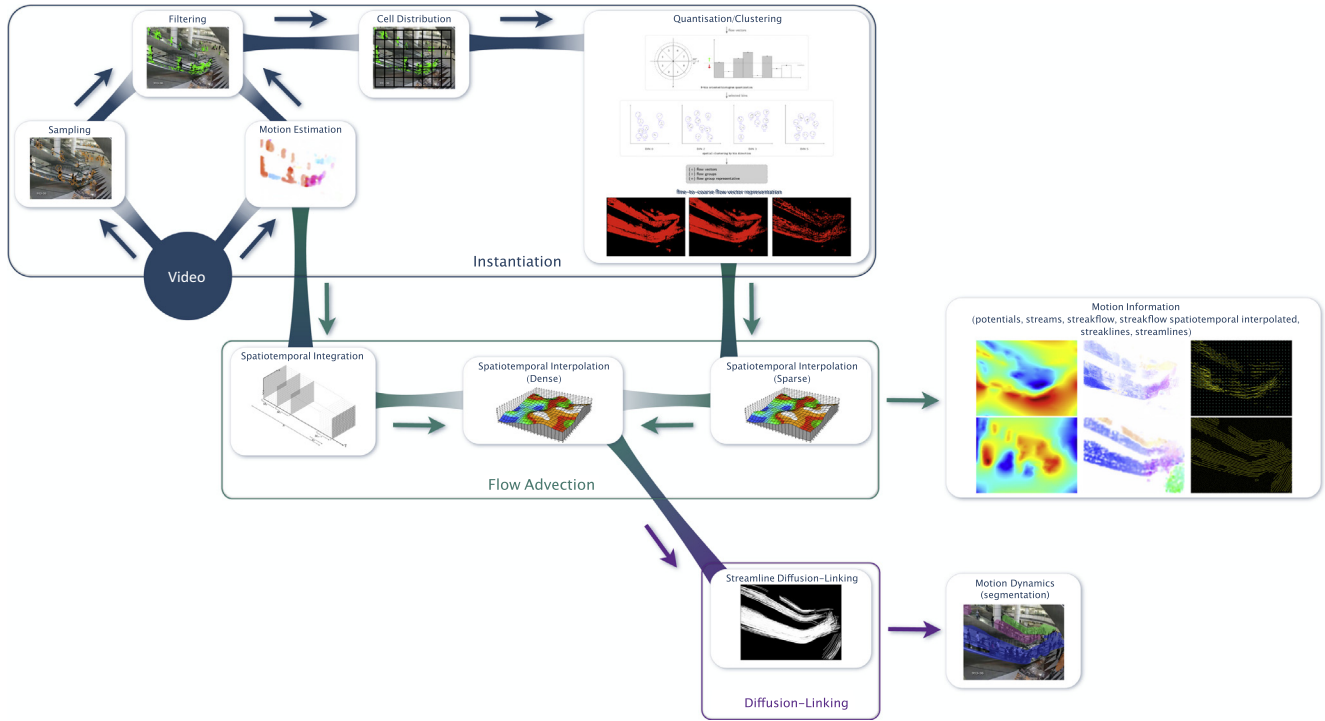


Fig. 1. System workflow.

$$\begin{aligned} x_w(t+1) &= x_w(t) + u(x_w(t), y_h(t), t) \\ y_h(t+1) &= y_h(t) + v(x_w(t), y_h(t), t) \end{aligned} \quad (3)$$

The repetition of this process at each frame yields a family of curves that represents the particle trajectory set. Since human motion creates unsteady flow, each point can be represented by a set of pathlines, streaklines, and streamlines.

3.1.2. Streaklines: computation and derived information

Streaklines are the locus of points that connect all the particles that had been originated from the same initial point in the past at a given time, and can be computed from the spatial and temporal gradients of the flow map. Streaklines should not get too long due to shape inconsistency with the flow and instability on numerical integration solution. Indeed, they propagate velocity changes given by instantaneous optical flow at the time of initialization. Spatial and temporal changes in the scene can be obtained using a streakline technique that extract both flow and potential functions based on integration of the velocity field and Helmholtz decomposition, respectively.

Mehran et al. [10] revealed that streaklines are the most informative flow representation when compared with optical flow and particle flow. In fact, they have important properties than can be used to characterize motion in spatial and temporal domains, such as: (i) *streak flow*, time integration of the velocity field results on streak flow representation, where motion information of the flow for a period of time is retained; this representation fills the gaps of optical flow and captures faster immediate dynamic flow changes than traditional particle flow representation [10] and (ii) *potential functions*, feature analysis of vector field can be performed through *Hodge-type* decomposition which leads to vector field separation into divergence-free, rotation-free, and harmonic components; this variational approach could locate and distinguish extremal points from the potential of each component such as sources, sinks, and vortices [21].

3.1.3. Streamlines: computation and derived information

Streamlines can be obtained by bidirectional numerical integration of vector field using an autonomous ODE (Ordinary Differential Equation) system. They can be described as curves tangent to the vector field at every point in the flow [22]. The integration starts from a seed point and ends when it: reaches another streamline's neighbor or a critical point, hits the domain boundary or forms a closed path. Several streamline placement algorithms have been proposed on the literature including flow topology based methods [23], evenly-spaced streamline placement method [24], and a hybrid flow topology-evenly-spaced streamline algorithm [25]. All of them share three common stages: (i) seed placement, (ii) diffusion process, and (iii) stopping criteria.

Flow feature analysis over streamlines permits to create rich semantic representations of trajectories. The combination of those local features forms the shape of a trajectory that encodes local motion patterns. Such shapes can be associated by similarity (different basis can be used: pixel-by-pixel, point-to-point or segment-to-segment) in order to define boundaries regions that segment motion, and to provide cues for abnormal behavior detection.

4. System overview

Traditional approaches for motion analysis that consist of detecting moving objects or features, tracking them, and analyzing their tracks, miss the estimation of long-term motion representations that bring important cues for scene understanding [26]. Our approach is able to extract long-range motion trajectories that encode spatial and temporal changes in the scene, as well as local motion statistics around each trajectory point in the form of discrete distributions. It follows a Lagrangian perspective to integrate motion through temporal domain under an Eulerian view, similar to the *extended particle* technique defined in [10]. Fig. 1 shows the overall system workflow.

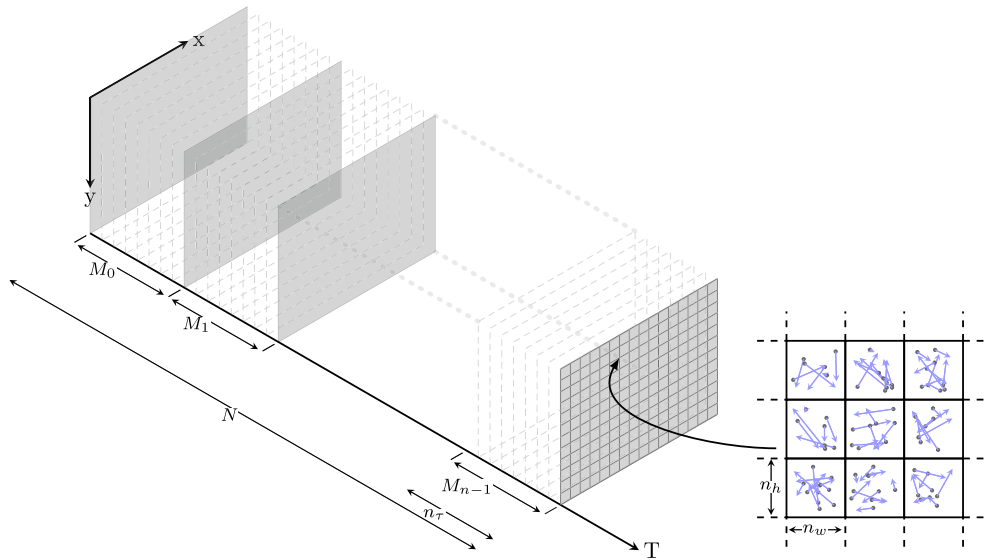


Fig. 2. Spatiotemporal video volume representation.

The system subdivides temporally the video in a series of mini-batches without overlap, and subdivides spatially each frame in a static grid. In this way, spatiotemporal cells are created (see Fig. 2). On each frame, the flow vectors are formed from the combination of a sparse sampling strategy with the flow map obtained from an optical flow algorithm. The locations of the sampled key points are used as starting points and a median filtering is applied to the current flow map to obtain approximated and smoothed ending points for each flow vector. All flow vectors are collected by the corresponding enclosing cell for further quantisation and clustering, based on the orientation and spatial information respectively. Such step is performed on each cell at the end of each mini-batch. This dual-operation aims to reduce the number of flow vectors and leads to a fine-to-coarse representation, which defines three levels of granularity: (i) *flow vectors*, all vectors collected during the mini-batch duration; (ii) *flow groups*, dominant groups of vectors obtained after quantisation and clustering; and (iii) *flow group representative*, the most dominant group, which is the one with the largest number of flow vectors within. Flow information of local neighborhood around each cell is acquired and continuously updated. This information permits a richer distribution to accurately infer entropy and energy measures, which help to obtain vector field's characteristics such as topology.

From the advection scheme we can extract streaklines initiated at each particle position. Such process should not be too long to avoid the propagation of flow changes that are inconsistent with the actual flow, which can also be caused by the optical flow algorithms that introduce noisy flow vectors. Human-like motion does not produce so well-defined properties such as vorticity. Therefore, to approximate human-motion scenarios to fluid motion and reduce propagation artifacts, the advection is executed between mini-batches and not between individual frames. This is one of the novel features that distinguish our system from the approach presented in [10].

Since pedestrian flow is unsteady, streaklines and streamlines curves are different in direction and shape. At the end of a set of consecutive mini-batches, so-called *memory cell* size, an averaged streak flow map is combined with one of the fine-to-coarse flow representations to obtain a dense vector field, which is used as input for the streamline diffusion process. However, such discretisation could be insufficient to avoid broken streamlines.

Instead of investigating about diffusion techniques of streamlines and improve them, we optimize globally the local short-range streamlines and obtain global long-range trajectories, which accurately correlate features related to flow decomposition and to the motion model, and can handle local ambiguities by spatiotemporal regularization.

Our system, due to its novel formulation, is able to extract meaningful and correct long-range trajectories that represent either object tracks and motion patterns for different scenarios such as crowds, with structured and unstructured movement, and multi-tracking, with sparse and dense groups. Correct system initialisation and parametrisation of its different steps for the aforementioned scenarios are studied and evaluated in this work.

5. System details

The input of the system is a monocular video sequence I of T frames, where its volume is $W \times H$ (pixels) $\times T$ (frames). The volume is subdivided into N spatiotemporal cells of size $n_w \times n_h \times n_t$, without overlap, where n_w is the width of the cell in pixels, n_h is the height of the cell in pixels, and n_t the number of frames in the cell, which is equal to the temporal duration of each mini-batch (Fig. 2). The video is composed by a set of mini-batches, $\mathcal{M} = \{M_n\}$. The output primitive is a trajectory $\Gamma = (\mathbf{S}(n), \mathbf{x}(n))$, where $\mathbf{S}(n)$ is the trajectory descriptor, and $\mathbf{x}(n) = (x(n), y(n))$ are the trajectory spatial coordinates at mini-batch n . The temporal coordinate, n , is integral (correspond to mini-batches) and the spatial coordinates, $\mathbf{x}(n)$, are in sub-pixel accuracy. The set of detected trajectories is denoted by $\mathcal{T} = \{\Gamma_i\}$. The Algorithm 1 summarizes the relevant steps of the system.

5.1. Instantiation

The beginning of this stage is composed by the sampling strategy and by the motion estimation represented by flow maps, which are indeed instantaneous velocities. Both information are combined to create the flow vectors. After, they undergo a filtering step to remove noise and outliers, and then are distributed by the enclosing cells, where are locally quantised and grouped. These steps form the *instantiation* block of the system and are executed at each frame, excepting the quantisation and clustering operation.

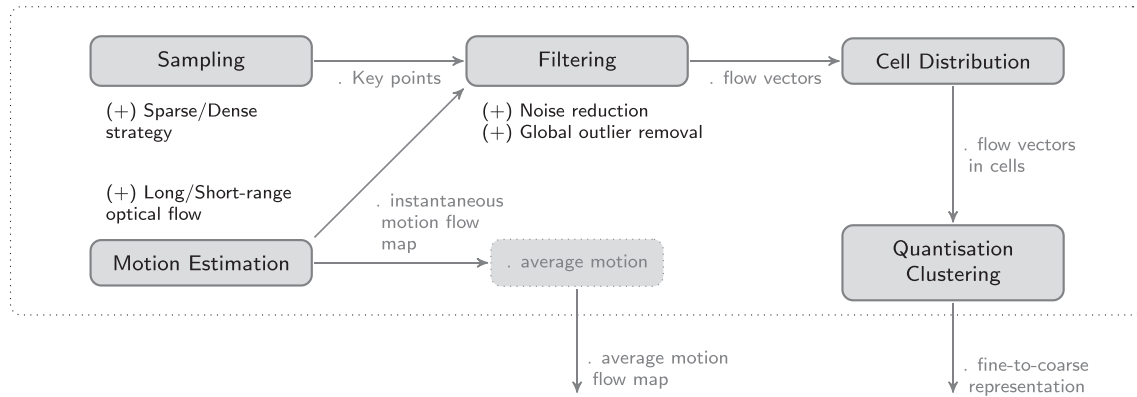


Fig. 3. System instantiation step.

Its output is: (i) an averaged flow map from the instantaneous flow maps of the current mini-batch and (ii) a global fine-to-coarse flow vector representation (see Fig. 3). All the operations taken in this stage are executed within each mini-batch.

Algorithm 1. Main Loop algorithm

```

1 Procedure Main()
2   for  $f = 1$  to  $T$  do
3     // Computes sparse/dense sampling and returns key points.
4      $keyPoints \leftarrow Sampling(f)$ 
5     // Computes instantaneous flow maps from Optical Flow algorithm.
6      $instFlowMap \leftarrow MotionEstimation(f)$ 
7     // Filtering step with outlier removal technique.
8      $flowVectors \leftarrow Filter(keyPoints, instFlowMap)$ 
9     // Computes an averaged flow map for the current mini-batch.
10     $avgFlowMap \leftarrow ComputeAvgFlowMap(f)$ 
11    // Distributes flow vectors through the enclosed cell.
12     $DistributeSpatially(flowVectors)$ 
13    if newMinibatch then
14      // Quantization/Clustering for fine-to-coarse flow vector
15      // representation.
16       $flowRep \leftarrow QuantizeAndClusterCells(cells)$ 
17      // Computes motion advection.
18       $ComputeMotionAdvection(avgFlowMapPrev, avgFlowMapCurr)$ 
19      // Computes an averaged streak map for the set of mini-batches.
20       $avgStreakMap \leftarrow ComputeAvgStreakMap(f)$ 
21      if  $nMinibatches = cellMemory$  then
22        // Extract sparse interpolated flow map from flow vector
23        // representation.
24         $interpFlowMap \leftarrow ComputeInterpFlowMap(flowRep)$ 
25        // Extract dense interpolated streak flow from combination
26        // between averaged streak flow and fine-to-coarse flow
27        // vector representation.
28         $interpStreakFlowMap \leftarrow$ 
29         $SpatioTemporalInterp(avgStreakMap, flowRep)$ 
30        // Compute streamlines from diffusion technique.
31         $streamlines \leftarrow ComputeStreamlines(interpStreakFlowMap)$ 
32        // Link broken streamlines to form representative long motion
33        // trajectories.
34         $trajectories \leftarrow LinkGloballyStreamlines(streamlines)$ 
35      end
36    end
37  end
38 end

```

5.1.1. Sampling and motion estimation

The sampling extracts a set of key points from one of two possible distributions: dense or sparse. The dense sampling is highly computational demanding, and such effort is propagated through the subsequent steps. Also, it introduces noisy points that do not add discriminative value, therefore sparse sampling is

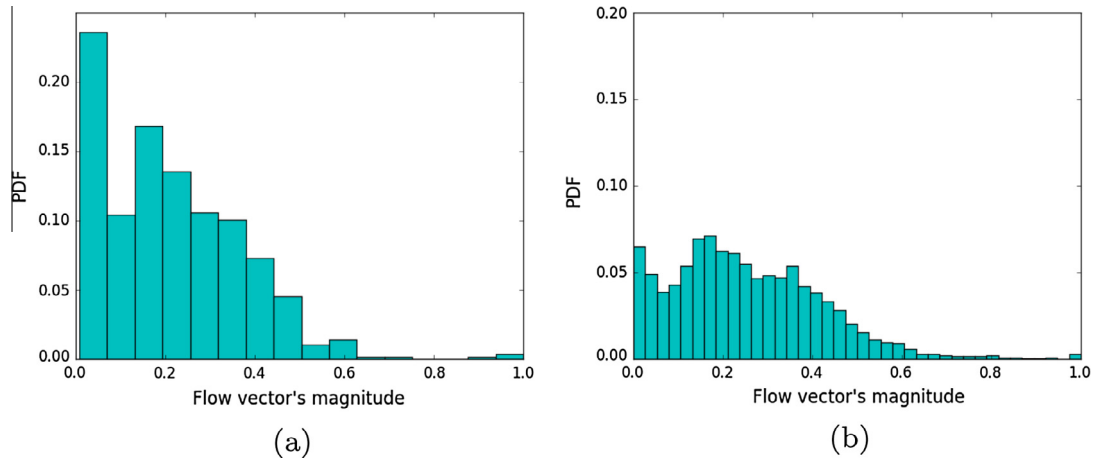


Fig. 4. Flow vector's magnitude distribution (C.U. scenario) in: (a) just one frame and (b) several frames.

preferred. The motion flow is estimated using optical flow algorithms, which differ from either frame-to-frame analysis or larger spatiotemporal displacements. We tested two algorithms: the short-classical Farneback [27], and the large-descriptor matching in variational model (LDOF) [28]. Both present good results but we adopted the Farneback's method due to the trade-off between computational effort and robustness. See [29] for a more detailed description about these results.

5.1.2. Filtering

Each flow vector is represented by $F_i = (x_i, y_i, u_i, v_i)$, where (x_i, y_i) is the sampling point, and (u_i, v_i) are the motion field components in x and y directions, respectively. The filtering step builds each vector flow by assuming the key point location to be the initial vector's position, $P_i(t) = (x_i(t), y_i(t))$, and considering the flow vector's endpoint, $P_f(t) = (x_f(t), y_f(t))$, as the median (component-wise) of the flow field, $f = (u(t), v(t))$, in a neighborhood of size K . Therefore,

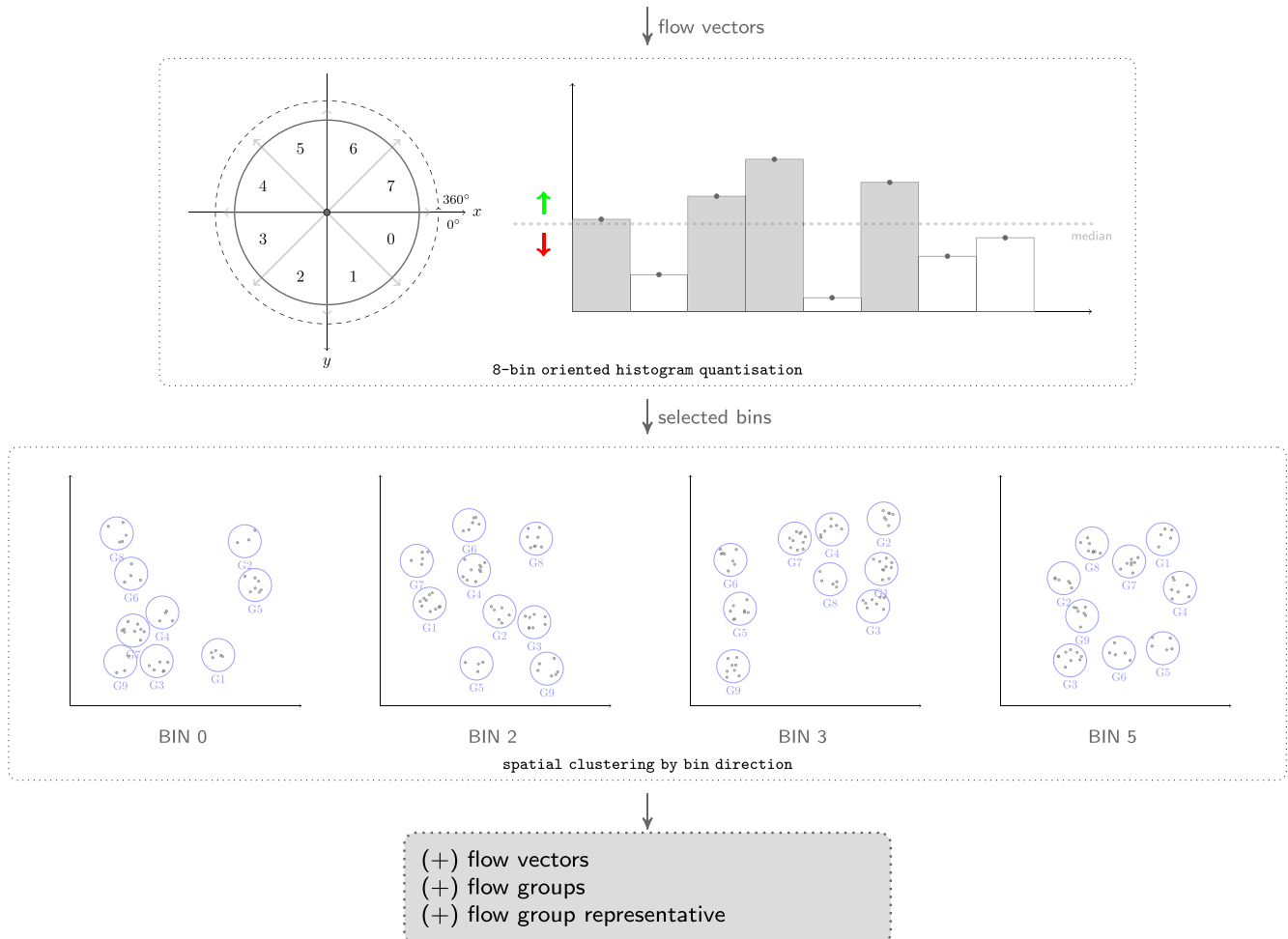


Fig. 5. Quantisation and clustering step by cell.

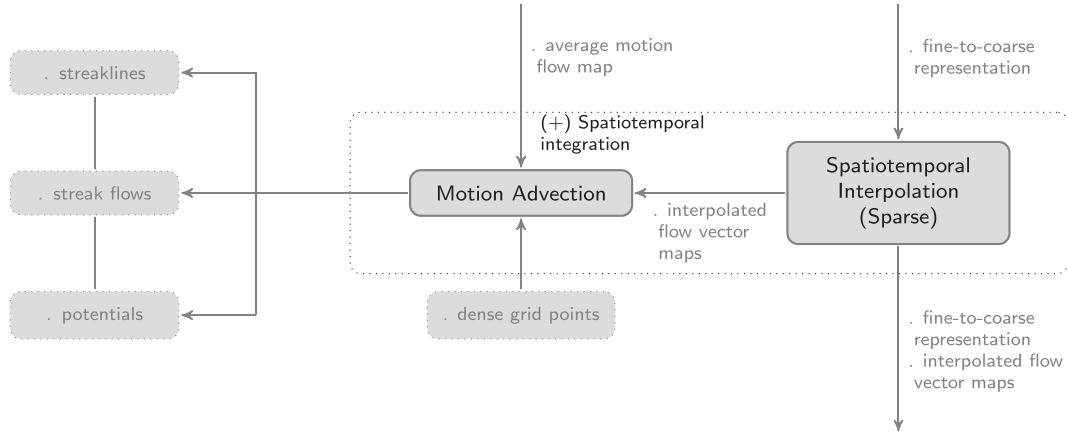


Fig. 6. System flow model advection step.

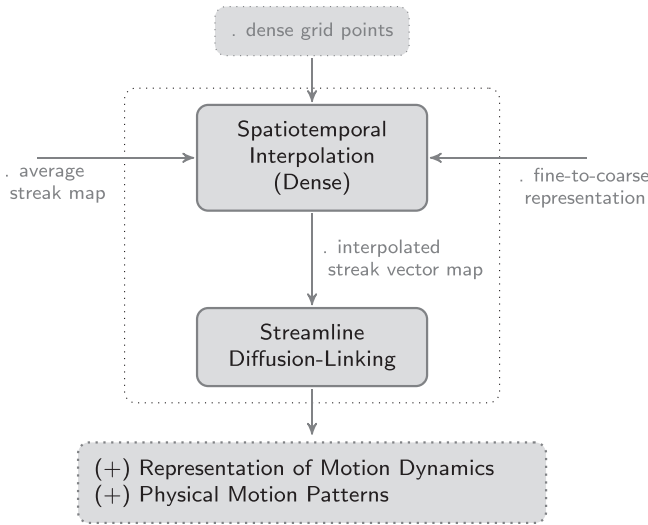


Fig. 7. System streamline diffusion-linking step.

the number of flow vectors is equal to the number of key points. The adopted method is a median filtering kernel that performs better than the bilinear interpolation [30].

A dual-threshold on flow magnitude is applied to remove flow vectors that have little motion information, as well as extremely high magnitudes. However, we empirically verified that such operation is not enough to remove the flow vectors resulting from the background noise. For this reason, a novel outlier removal technique is proposed. It consists on a rough approximation that fits the magnitude of the flow vectors into a unimodal gaussian distribution and estimates both lower and upper bounds inspired on the Chebyshev's theorem and on the skewness measure.

Since the magnitude of the flow vectors depend on the instantaneous velocities and on the kernel size, we expect that most of the flow vector's magnitude to be smaller than the mean magnitude. Therefore the distribution will not be symmetric and will be, normally, skewed to the right, as illustrated by Fig. 4.

The Chebyshev's theorem can be applied to any data set regardless of its distribution. In this case, due to the sharp skewed distribution, the Chebyshev's inequality is not good enough to estimate the non-symmetric, lower and upper, bounds, $\{\ell_-, \ell_+\}$. Our technique estimates two parameters to obtain those bounds, a shift factor, χ , and a scale factor, ρ . The intuition behind is that the nonparametric skew measure gives a shift factor to account with

the difference between the median and the mean, accordingly with the distribution's tendency (positive or negative skew), while the ratio between the number of observations represented by the median and by the mode provides a factor to scale the shift amount (the greater the ratio, the smaller the scale factor, and, consequently, the larger will be the acceptance gap between the threshold limits).

The nonparametric skew measure is given by $\gamma = (\mu - \nu)/\sigma$, where μ is the mean, ν is the median and σ is the standard deviation, with values between $[-1, 1]$. Considering these boundaries, the shift factor is decomposed in the following limits

$$(\chi_-, \chi_+) = \begin{cases} (\gamma, 0), & \text{if } \gamma > 0 \\ (0, |\gamma|), & \text{if } \gamma < 0 \end{cases} \quad (4)$$

The scale factor, $\rho = 1 - \tilde{x}/\tilde{\chi}$, results from the relation between the number of observations represented by the median, \tilde{x} , and the mode, $\tilde{\chi}$, of the log-transformed flow vector's magnitude distribution. We employ the log-transformation to approximate the data to a symmetric distribution before measuring the amplitude relation. We adopt the Freedman-Diaconis rule to obtain the optimal bin width, $Bin_w = 2 \cdot IQR \cdot n^{-\frac{1}{3}}$, where IQR is the interquartile range, and n is the number of observations in distribution. The scale factors are given by $\{s_-, s_+\} = \{\rho \cdot \chi_-, \rho \cdot \chi_+\} \in [0, 1]$, which produces the final non-symmetric bounds, $\{\ell_-, \ell_+\} = \{\sigma \cdot s_-, \sigma \cdot s_+\}$, which are used to estimate the final threshold limits for outlier removal, $\{\lambda_-, \lambda_+\} = \{min + \ell_-, max - \ell_+\}$, where min and max are the minimum and maximum values of the flow vector's magnitude distribution, respectively.

After several experiments on different datasets, this technique presented coherent and better results to remove global outliers on flow vector's data than state-of-the-art methods (see Section 6.2).

5.1.3. Cell distribution

The video volume has a regular spatiotemporal distribution. Spatially each frame is divided by a grid, whose resolution is dependent on the frame size and is set at the beginning. Temporally the video duration is evenly divided. Each spatiotemporal region is denominated a cell, C_i , and contains the flow vectors whose initial positions lay inside it. Each flow vector is encoded by $F_i = (x_i, y_i, L_i, \theta_i, t_i)$, where (x_i, y_i) is the sampling point, L_i is the flow magnitude length, θ_i is the flow angle relative to positive x -axis, and t_i is the frame. This step as well as the previous ones are executed every frame.

5.1.4. Quantisation and clustering

In order to obtain a fine-to-coarse flow vector representation that could permit to model different levels of patterns, a two-step quantisation and clustering approach is applied on each cell at

the end of each mini-batch. This operation considers all the flow vectors collected along the duration of the mini-batch, therefore the number of key points is much greater than the number of cells. The aim is to reduce the number of flow vectors, while maintaining the geometric structure of the flow field, and to obtain different representations of local dominant motion flows in a fine-to-coarse scale. Fig. 5 illustrates this process.

The first quantisation step uses the flow vector angle and considers a full-degree histogram (360°) with 8 bins to represent orientation groups. Only the major groups, with weight above the histogram's median value, are taken into account for next clustering step. This eliminates noisy flow vectors whose orientation fall apart the expected local distribution. The second step uses the flow vector position and applies a spatial clustering on each valid orientation group. A k-means approach with center initialisation [31] is adopted. To select a robust k value, we compute k-means with increasing k until the compactness measure, C_k , satisfies the condition

$$\frac{C_{k+1} - C_k}{C_{k+1}} < t_c \quad (5)$$

with $C_k = \sum_{i=1}^n \|s_i - c_l\|^2$, where s_i is the input sample i , c_l is the l index clustering center to which sample i belongs, and t_c is the compactness ratio threshold, which is normally selected very low (≈ 0.01), and we set $k_{opt} = k + 1$.

Several clusters per orientation group are obtained, the so-called dominant groups, which are weighted by the number of flow vectors that belong to them and are ordered in a descendent-way. These groups represent the local dominant flows, which are described by $L_i = (x_i, y_i, n_i, \theta_i)$, where (x_i, y_i) is the average position, n_i is the total number of flow vectors, and θ_i is the average orientation angle. A fine-to-coarse representation, with three levels of flow vector granularity, is obtained: (i) *flow vectors*, the set of all flow vectors collected and filtered, $\mathcal{F} = \{F_i\}$; (ii) *flow groups*, the prototypes of the set of local dominant groups, $\mathcal{L} = \{L_i\}$; and (iii) *flow group representative*, the principal local dominant group, L_{rep} . They are useful for computational consumption requirements and to investigate their length and time scale impact on the dynamics of the motion advection system. This three-level global flow vector representation is obtained per mini-batch. However, it can be accumulated during the last b mini-batches, so-called *memory cell*, in order to create dense flow field representations for different discriminative levels. In this paper, we explore these representations for streamline diffusion.

At this stage, each cell is able to estimate local region parameters using information from its neighborhood. One of them is the entropy, whose computation might not be reliable if a small number of samples is presented. To overcome this problem, for each cell we collected as samples its flow vectors, the *flow groups* of the cells that belongs to its neighborhood of size $K \times K$, and replicate its flow vectors at boundary cells. The entropy calculation relies on the probability of each angular bin x_i , $p(x_i) = C(x_i) / \sum_{i=1}^n C(x_i)$, where $C(x_i)$ corresponds to the number of vectors in bin x_i . Entropy is then measured and used on this work as an indicative of: (i) degree of vector variation in its local neighborhood, which measures the saliency value and highlights the possibility of being a critical point or belonging to a separation line in the global vector field and (ii) degree of vector variation between subsequent mini-batches, which measures the difference in the information content between both vector fields, considering the distribution shape difference and highlighting a possible new local motion pattern.

5.2. Flow model advection

This stage captures long-range temporal dependencies to represent spatial and temporal features of the flow. It is responsible to

Table 1
Datasets characteristics.

Name	Category	Frame Size	Fps	# Frames
UCF 913-361 ^a	Sparse Crowd Structured (S.C.S.)	480 × 360	25	467
Marathon ^b	Dense Crowd Structured (D.C.S.)	720 × 480	25	150
UMN seq3 ^c	Sparse Crowd Unstructured (S.C.U.)	320 × 240	30	658
IM03 ^b	Dense Crowd Unstructured (D.C.U.)	720 × 480	25	67
PETS2013 S2 L1 Time12-34 View001	Dense Multi-Tracking (D.MT.)	768 × 576	–	794
PETS2013 S2 L3 Time14-41 View001	Sparse Multi-Tracking (S.MT.)	768 × 576	–	240

^a UCF Crowd Segmentation Dataset <http://crcv.ucf.edu/data/crowd.php>.

^b Coherent filtering from crowd clutters dataset <http://mmlab.ie.cuhk.edu.hk/projects/coherentfiltering/>.

^c Unusual Crowd Activity dataset <http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi>.

advect the motion considering an average flow map along the entire mini-batch and a dense grid of particles. It uses the global fine-to-coarse flow vector representation in a twofold way: (i) interpolated to produce sparse vector maps and (ii) in combination with a sampling strategy to enrich the representation. This stage is executed at each mini-batch and its output is a set of motion representations such as streaklines, potentials, streak flow, and interpolated flow vector maps, among others (see Fig. 6).

5.2.1. Motion advection and spatiotemporal interpolation

A dense grid of particles is considered. Each particle has fluid properties and their initial position correspond to each pixel on image. This characteristic follows the assumption that the computation of the streakline vector field needs a dense path line integration [32]. All particles are integrated over time accordingly to an average of the optical flow maps along the current mini-batch, which is restarted at the beginning of the next mini-batch. On each time step a particle on position p is created and all other particles previously initialised on same position follows the flow field. This process is expressed by Eq. (3) and is repeated along the *memory cell* size to obtain the streaklines. The streak flow is extracted from temporal integration of the velocity field. We use the Runge-Kutta-Fehlberg (a.k.a. RKF45) for this purpose.

Considering that each streakline is a collection of particles, we get a set of 3D data points for x and y flow directions. To compute the streak flow, $\Omega_s = (u_s(x, y), v_s(x, y))$, with sub-pixel level accuracy, we adopt a multi-resolution method based on B-spline refinement to approximate scattered data by error minimization on both dimensions. The 3D data points are given as input, the tensor product B-spline surfaces are produced, and the result is a least square approximation to the scattered data with B-splines for each flow direction that represents the streak flow on each direction (x, y) .¹ The same procedure is used to obtain the spatiotemporal interpolated flow maps of each fine-to-coarse representation, considering instead the accumulated flow vectors of each one.

5.3. Streamline diffusion-linking

Due to the ending conditions of the streamline diffusion process, described on Section 3.1.3, and since we are interest in

¹ We used the Least Squares Approximation of Scattered Data with B-splines Library (<http://www.sintef.no/Projectweb/Geometry-Toolkits/LSMG/>).



Fig. 8. Comparison results with and without the proposed outlier removal technique. By row: from left to right, with outliers and after outlier removal. By column: from top to bottom, S.C.S., S.C.U., D.M.T., and S.M.T.

extracting long-range streamlines to represent global motion trajectories, we include a post-processing step that links short streamlines. This stage uses the flow information collected over a sequence of mini-batches, of *memory cell* size (see Fig. 7).

Streamline diffusion requires as input a vector field obtained from a dense flow field and the largest it represents the temporal and local changes over time, the longer the streamlines, and the better they emphasise the global field temporal coherency and the better they describe the topology of the flow. To this end, we use a combination of: (i) the averaged streak flow from the current set of mini-batches; (ii) the set of flow vectors from a specific fine-to-coarse representation collected along the current set of

mini-batches; and (iii) a dense grid of particles. The resulting flow field is formed by B-spline interpolation, as explained on Section 5.2.1, considering as input the set of flow vectors from the fine-to-coarse representation superimposed to the averaged streak flow. This flow field is converted into a vector field using a grid-based discretisation and the filtering step explained on Section 5.1.2. We adopted the state-of-the-art farthest point seeding method [33] as the streamline diffusion technique.²

² Algorithm is implemented on the Computational Geometry Algorithms library (CGAL) <http://www.cgal.org/>.

Table 2
System parameters.

Memory cell size	Minibatch size				
10	2	4	6	8	10
Minibatch size	Memory cell size				
5	3	6	9	12	20

The streamline linking process permits to obtain long-range streamlines. It is formulated as a combinatorial matching problem that considers compatibility in terms of flow appearance, motion, and spatiotemporal regularization among all short-streamlines. We adopted a discrete Markov Random Field (MRF) process to encode association constraints between *query* and *candidate* streamlines, re-correlate the set of short-streamlines and extract an optimal linkage between them. This undirected model was inspired on [34]. Our formulation in terms of probability of linkage, \mathcal{L} , between streamlines is defined by

$$P(\mathcal{L}) = \prod_i \phi_i(l_i) \prod_{ij \in \mathcal{N}(i)} \psi_{ij}(l_i, l_j), \quad (6)$$

where $\phi_i(l_i)$ are the unary potentials that model the compatibility between a query streamline, q_i , and a candidate streamline, c_j ; $\psi_{ij}(l_i, l_j)$ are the pairwise potentials for link regularization, in case of tracking ambiguities, between a pair of query streamlines, q_i and q_j , considering the candidate streamlines that lay in their spatiotemporal neighborhood, $\mathcal{N}(i)$. The global optimization problem, given by Eq. (6), is inferred using a tree-reweighted belief propagation. Under this context, the streamlines taken on the MRF process are called tracks.

The compatibility term is divided into three components: (a) the appearance similarity, ϕ_a , which models the flow properties; (b) the motion similarity, ϕ_m , which takes into account the velocity information; and (c) the prior on motion model, ϕ_p , which approximates a motion model to predict next streamline's position in case of large discontinuities. Appearance and motion similarity terms consider a symmetrically weighted average comparison of features along the last n elements of the query track, q_i , and the first n elements of the candidate track, c_j . The weight is an exponentially decaying factor, $w_t(k) = \alpha^k$, $0 < \alpha < 1$, that works as a confidence parameter.

Instead of considering an individual average information (motion or appearance) for each track and then take their difference in the similarity term, as used in [34], we adopted point-to-point operations. We formulate the appearance term between tracks Γ_i and Γ_j based on the cosine similarity of the streak flow's angle at each track's position given by

$$\mathbf{s}_{ij} = \frac{1}{Z} \sum_{k=0}^{n_a-1} (S_i(t_i^{end} - k)w_o(t_i^{end} - k) - S_j(t_j^{start} + k)w_o(t_j^{start} + k))w_t(k) \quad (7)$$

where $S_i(t)$ is the track's, Γ_i , cosine of the streak flow angle at time t , $w_o(t)$ is an outlier weight that measures how well $S_i(t)$ fits the appearance characteristics of the entire track, Γ_i , which is modeled by a Gaussian distribution of the track's streak flow angles. The same is defined for track Γ_j . The normalization factor is expressed by

$$Z = \sum_{k=0}^{n_a-1} (w_o(t_i^{end} - k) - w_o(t_j^{start} + k))w_t(k) \quad (8)$$

and the appearance similarity is defined by

$$\phi_a = \exp\left(-\frac{1}{\sigma_a^2} \|\mathbf{s}_{ij}\|\right) \quad (9)$$

The motion term considers the velocity variation. Similarly, the velocity difference is taken by a point-to-point track relation stated by

$$\mathbf{v}_{ij} = \sum_{k=0}^{n_v-1} (v_i(t_i^{end} - k) - v_j(t_j^{start} + k))w_t(k), \quad (10)$$

where $v_i(t)$ is the track's, Γ_i , velocity at time t . The same is defined for track Γ_j . The motion similarity is expressed by

$$\phi_m = \exp\left(-\frac{1}{\sigma_m^2} \|\mathbf{v}_{ij}\|\right) \quad (11)$$

The prior on motion model that predicts track's movement on discontinuities considers linear kinematic equations to estimate the closest point of the query track, Γ_i , to the initial point of the candidate track, Γ_j . The motion integration is done until the distance travelled equals the length between the last point of Γ_i and the first point of Γ_j , and is governed by

$$\begin{aligned} \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t) + \frac{1}{2}\mathbf{a}_i(t) + \mathbf{v}_i^{flow}(t) \\ \mathbf{a}_i(t+1) &= \mathbf{v}_i(t+1) - \mathbf{v}_i(t) \end{aligned} \quad (12)$$

where $\mathbf{v}_i^{flow}(t)$ is the flow vector velocity at position $\mathbf{x}_i(t)$. The next velocity, $\mathbf{v}_i(t+1)$, is randomly chosen from a Gaussian distribution of the velocities of the track Γ_i . After this, a weighted distance, that includes spatial and angular values between the last predicted point of the query track and the initial point of the candidate track, is used to obtain the motion discontinuity similarity term

$$\phi_p = \exp\left(-\frac{1}{\sigma_p^2} \left(\alpha \|\mathbf{x}_j^{start}(t) - \mathbf{x}_i^{end}(t)\| + (1 - \alpha) \angle(\mathbf{l}_i, \mathbf{l}_j)\right)\right) \quad (13)$$

where \mathbf{l}_i is the last segment of the query track, \mathbf{l}_j is the first segment of the candidate track, $\angle(\mathbf{l}_i, \mathbf{l}_j)$ is the angle between both segments, and α is a weighted factor (in this case 0.5). For each set of mini-batches, the system extracts a set of streamlines, which are connected with the streamlines obtained from the subsequent set of mini-batches.

Every node in the graph, i.e. every streamline, has an additional state with a predefined cost to represent the terminal state and to avoid a forced linking. The graph just defines unary potentials among streamlines that present a compatibility term, ϕ_i , below a predetermined threshold. Candidate track's formation is evaluated under geometrical constraints: (i) length (d_{thr}), which defines the spatial distance between the last point of the query track and the first point of the candidate track; (ii) continuity direction (θ_{dir}), which returns the angle between the last segment of the query track and the segment that links the last point of the query track with the first point of the candidate track; and (iii) direction difference (δ_{dif}), which states the angular difference between the last segment of the query track and the first segment of the candidate track. Only the ones that satisfy pre-defined thresholds are included in the graph. In the same way, the compatibility term between query tracks, ψ_{ij} , follows a geometrical pruning with the same constraints, excluding the δ_{dif} constraint. In terms of temporal neighboring, only the tracks which belong to the same set of mini-batches are considered in the link regularization step. This pruning process effectively reduces the computational effort without affecting the final results.

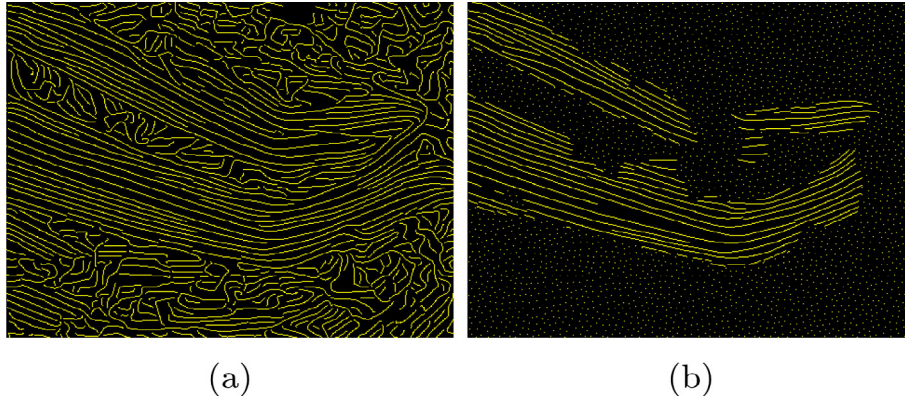
We took further advantage of the system's characteristics and used the cell's entropy to detect the areas with high entropy values. The query tracks whose ending points and candidate track whose starting points fall on these regions are not considered in the linking process. This highly improves efficiency.

Table 3

Sensitivity and specificity of various outlier removal techniques (%).

	Std	Thompson-Tau [35]	Mzscore	Zscore	Adj-Boxplot [36]	ExpSM [37]	Grubbs [38]	Ours
<i>UMN seq3</i>								
TP	97.9 (± 1.3)	18.9 (± 5.5)	27.4 (± 22.6)	98.7 (± 0.9)	58.9 (± 10.7)	68.0 (± 4.6)	92.2 (± 15.9)	79.7 (± 6.5)
TN	1.7 (± 1.2)	19.2 (± 8.1)	16.5 (± 6.7)	1.2 (± 0.8)	12.7 (± 5.5)	21.0 (± 4.6)	3.6 (± 4.4)	82.7 (± 6.0)
TB	49.8	19.1	21.9	49.5	35.8	44.5	47.9	81.2
<i>UCF 913-36l</i>								
TP	96.6 (± 5.7)	13.8 (± 13.6)	29.3 (± 33.7)	97.9 (± 3.6)	58.6 (± 21.7)	77.3 (± 12.1)	76.8 (± 31.6)	79.9 (± 14.2)
TN	2.3 (± 1.8)	24.8 (± 21.9)	17.8 (± 10.7)	1.7 (± 1.3)	21.3 (± 21.1)	16.5 (± 7.2)	7.8 (± 6.2)	89.7 (± 6.4)
TB	48.4	19.3	23.5	49.8	39.9	46.9	42.3	84.8

Bold values are the best results by row.

**Fig. 9.** Effects caused by the outlier removal technique on streamline formation (C.S. scenario): (a) without outlier removal and (b) with outlier removal (single points are seeds that were, correctly, not diffused by lack of meaningful flow field).

6. Experimental results

We present results on several datasets according to the following criteria: (i) crowd scenario, both structured and unstructured movement, which are further subdivided into sparse and dense crowd density and (ii) multi-tracking scenario, both sparse and dense groups (see Table 1).

6.1. System parameters

In our previous work [29], we tested the instantiation stage with satisfactory results. A baseline was reached, namely we adopted the FAST sampling, the median filtering kernel size of $K = (13, 13)$, the LDOF's optical flow algorithm [28], and a spatial cell size of $C = (15, 15)$. In this work, considering empirical research, we fixed the neighborhood size to $\mathcal{N} = (3, 3)$.

We identified four system parameters: (i) cell size, which influences the granularity from which motion is aggregated, filtered, and advected; (ii) kernel size, which conveys a rule for common kernel operations such as flow vector computation and filtering; (iii) *minibatch* size, which sets the length of the streaklines, as well as the flow vectors quantisation; and (iv) *memory cell* size, which sets the number of mini-batches for streamline formation. After some experiences, we verified that the most relevant are the last two. For further analysis, we vary *minibatch* size fixing *memory cell* size, and vice versa according to Table 2.

6.2. Outlier removal

The outlier removal technique assumes a predominant role on system accuracy. Fig. 8 provides a qualitative confirmation. A quantitative comparison of the proposed technique with several outlier removal methods is provided in Table 3, reported under

three metrics, namely true positive (TP) rate, true negative (TN) rate and the mean of both, here called the true balance (TB) rate. Results were computed considering the (S.C.S.) and (S.C.U.) scenarios. For the former, nearly 20 frames were masked, since it is a large video sequence with many persons per frame, while for the latter, all the frames were masked. Pedestrians were manually annotated with bounding boxes and the masks are the inner ellipses within each one. The flow vectors that lay outside the masks are considered the background, i.e. the outliers, which are considered as negative samples.

We have tested with simple and more complex techniques (for most of them, we used the available source code). For instance, the most *naïve* approach, the *Std* method, considers as outliers the samples that distance from the mean more than 3σ , while more robust approaches follow statistical models that deal with skewed data. All techniques are applied on the log-normalized magnitude distribution of flow vectors. Results show that our technique keeps a high TP rate and, more importantly, it demonstrates a higher TN rate, proving its effectiveness on removing background's flow vectors. None of the others techniques are able to achieve satisfactory TN rate, which support our evidence to propose a new outlier removal technique for this problem. The combined TB rate clearly shows the overall supremacy of our technique. We conclude that the combination of the Chebyshev's theorem with the skewness metric brings a stabilization factor to the initial rough approximation of the unimodal normal distribution for the formulation of our outlier removal technique.

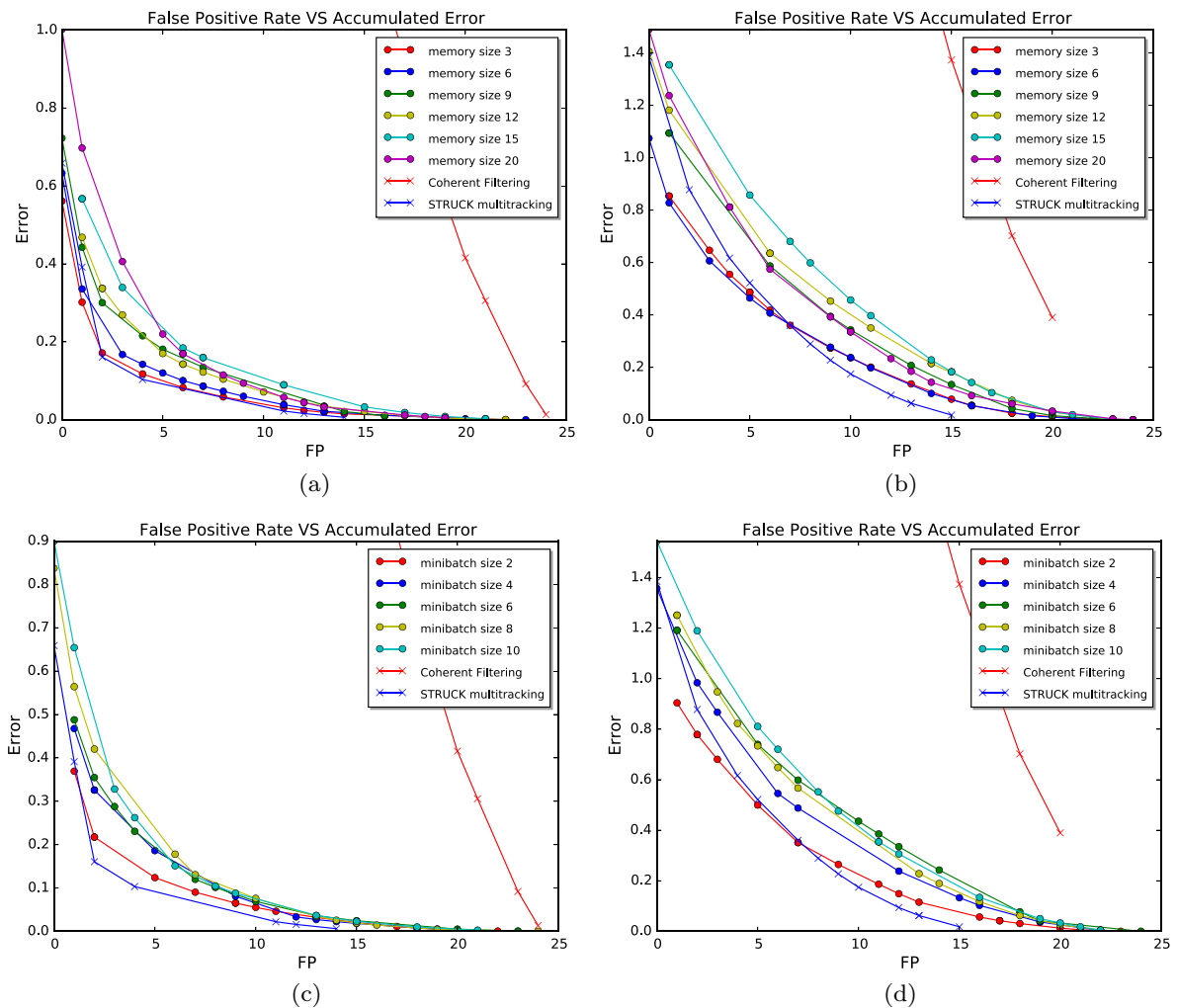
This step is crucial for the motion advection step. Fig. 9 confirms the high perturbation that undesired flow vectors introduce on the streamline formation. The great advantage of the proposed outlier removal technique is that it is data-driven, therefore it automatically adjusts to several scene contexts, as well as to large motion variations during the same scenario. We verified that the percentage of

Table 4

Number of trajectories per system parameters on each dataset at different steps (BP: before pruning; AP: after pruning; AL: after linking).

Dataset	Minibatch size (for memory cell size = 10)														
	2			4			6			8			10		
	BP	AP	AL	BP	AP	AL	BP	AP	AL	BP	AP	AL	BP	AP	AL
S.C.S.	2058	574	434	918	263	222	573	165	143	426	121	106	405	106	89
D.C.S.	391	135	87	154	53	37	95	38	28	45	16	15	48	18	18
S.C.U.	2508	995	169	1381	540	153	891	346	136	770	283	128	580	196	88
D.C.U.	349	107	52	108	32	16	106	31	17	113	33	18	112	30	20
D.MT.	6252	2088	1171	4156	1184	691	3597	1020	566	2695	707	389	2263	616	319
S.MT.	2111	516	247	797	235	127	714	194	128	424	125	87	356	100	60

Dataset	Memory cell size (for minibatch size = 5)														
	3			6			9			12			15		
	BP	AP	AL	BP	AP	AL	BP	AP	AL	BP	AP	AL	BP	AP	AL
S.C.S.	3043	818	598	1372	368	297	879	244	199	618	165	143	497	145	122
D.C.S.	542	186	102	258	88	55	147	53	35	–	–	–	–	–	–
S.C.U.	3514	1398	236	1746	662	175	1180	441	157	877	339	132	716	267	117
D.C.U.	451	125	63	218	63	32	104	29	14	–	–	–	–	–	–
D.MT.	3948	1294	775	5705	1656	1001	4167	1178	706	3521	973	552	2949	814	456
S.MT.	2978	711	351	1260	332	190	741	222	124	704	186	115	347	113	81

**Fig. 10.** FP rate VS error on S.C.S. scenario with median RLS regularization for: (a) *memory* variation and DTW metric; (b) *memory* variation and euclidean metric; (c) *minibatch* variation and DTW metric and (d) *minibatch* variation and euclidean metric.

flow vectors removed is between [65%,75%], but it could varied around [20%,85%], depending on the scene category and the video characteristics.

6.3. Global motion trajectories

Our aim in this paper is to show that our system is able to extract long-range global motion trajectories that approximate the ones obtained from multi-tracking approaches, along with the advantage to be useful and effective in different scenarios. To the best of our knowledge there is no related work with similar goal. However, for comparative reason, we selected two state-of-the-art methods that encompass micro and macro approaches of motion, respectively: (i) *STRUCK* [39], which is a tracker initialized by a search bounding box that outputs the tracked trajectories and (ii) *Coherent Filtering* [17], which is a global motion method that obtains clusters of points given by coherent neighbor invariance, whose temporal continuity is further performed through spatio-temporal relationships; to obtain the trajectories, we connected the points between consecutive pair of frames by nearest neighborhood, while optimizing the all-against-all distance matrix.

Our method obtains the final trajectories by the linking process. After empirical tests, the parameters of the selected farthest point seeding method were fixed to $d_{sep} = 4.3$ and $d_{rat} = 1.3$, both in sub-pixel accuracy and within the optimal ranges stated in [33]. The former controls the spacing distance of the density field, and the latter expresses the saturation ratio to trigger the seeding of a new streamline. It is necessary to execute a pruning step to remove redundant and short streamlines. Such process removes trajectories with a number of points below the mean value of the whole set of trajectories extracted on all set of mini-batches of the entire sequence. We used in this step the number of points since after the diffusion process the points are very close to each other (nearly one pixel), therefore its value is approximately equal to the trajectory length. Table 4 states the significant reduction of trajectories achieved with the pruning and linking processes.

After empirical experiments to obtain the maximum number of linked trajectories with coherent motion continuity, the geometrical constraints for the MRF graph were fixed to $d_{thr} = 45$, $\theta_{dir} = 42^\circ$, and $\delta_{dif} = 40^\circ$. For the remaining parameters used on the similarities terms, we follow [34] with the exception of n_p and n_v , which are the number of points taken for appearance and velocity similarity terms, respectively. They were fixed heuristically considering a percentage of the trajectory with the minimum number of points (n_{min}), $n_p = 0.7 \times n_{min}$ and $n_v = 0.35 \times n_{min}$. The velocity term is lower due to a higher expected variation, therefore less point-to-point measures are considered.

To the best of our knowledge there is not any evaluation framework that deals with comparison between manual trajectories and automatic long-range motion trajectories of pedestrians. We extend our previous work [29] to propose an evaluation methodology that supports clustering of trajectories by similarity to obtain the most representatives, and measures the correspondence between extracted and annotated trajectories. We follow a similar approach to [40] and use a one-to-one distance function between each annotated trajectory and each auto-generated one to obtain a distance matrix, and solve the assignment problem with the Hungarian algorithm [41]. We report the quality of the matching process with the miss detection (FN) and false positive (FP) rates.

The distance matrix passes through a regularization process before applying the Hungarian algorithm. This step favors configurations with small residuals and down-weights large errors, that could otherwise dominate the matching process. We evaluate four alternatives to regularise the distance matrix: (i) *clustering threshold*, where a K-means is applied to the distances and the *max* of the cluster with the lowest values is taken as the desired threshold

to truncate the matrix entries up this value; (ii) *quartile threshold*, where the third quartile of the distances distribution is taken as threshold, and a similar truncation process is applied; (iii) *median RLS*, where the median of the distances is used as the σ parameter for the robust least square (RLS) approach expressed by $\rho(u, \sigma) = u^2 / (\sigma^2 + u^2)$, where u is each distance value; and (iv) *local scaling RLS*, where the same robust least square approach is taken, but instead of using the median value, the mean of a clustering based on local scaling is used. Such value is computed as follows: (1) a K-means is applied to the distances, (2) the *max* of each cluster is used to obtain a distance based on local scaling as stated on [42], and (3) the mean of those distances is considered to be the σ value.

An annotated trajectory is considered correctly matched if its distance to an auto-generated one is below a certain threshold. The threshold is also used to evaluate the relationship between the accumulated error, sum of the distances between the trajectories correctly matched, and the false positive rate. Such threshold is considered to be the *max* of the cluster with lowest values after applying the same K-means process detailed on Section 5.1.4. Before computing the distance matrix, all trajectories are resampled using a cubic spline interpolation scheme, where the resampling value is global and is equal to the length of the trajectory with the minimum number of points.

Next, we present some results for each dataset that will help us to answer the following questions: (i) what is the influence of *memory cell size*; (ii) what is the influence of *minibatch size*; (iii) what is the influence of the regularization matrix distance method on the assignment process; and (iv) what is the most suitable distance function. We took conclusions based on the relation between the false positive rate and the accumulated error, varying the threshold for the incorrectly classified matches. The following results are presented accordingly with the regularization step and the distance function. For instance, the chosen distance metrics between trajectories are Euclidean, Chebyshev, Hausdorff, Hausdorff-Modified, Dynamic Time Warping (DTW) and Longest Common Subsequence (LCS), while the regularization that leads most times to the lowest error is selected. The distance matrix is normalized by the minimum and maximum, therefore the accumulated error for each metric can be compared. The distance functions are calculated considering four feature's trajectory, (x, y, dx, dy) , which are the point coordinates and the normalized vector direction between subsequent trajectory segments.

6.3.1. Sparse Crowd Structured (S.C.S.) scenario

For this scenario a manual annotation of pedestrians was performed. Since the motion is structured, i.e. is represented by common motion patterns, for each one we selected and annotated several persons that undergo a motion pattern. At the end, we obtained representatives trajectories for each pattern, which led to 25 trajectories.

Both *memory cell size* and *minibatch size* have similar behavior, namely lower values have less error and FP rate associated, therefore for this type of scenario *memory* and *minibatch size* should be kept low. We also notice that *memory size* is less sensitive than *minibatch size*, therefore *memory size* could be higher than the *minibatch size*. The chosen regularization was the median RLS, since, in general, it presents lower error. The DTW measure presents a steeper monotonically decreasing behavior which shows a better compromise between the error and the FP rate. It also produces less false positives. Euclidean distance also performs well, while LCS measure presents the worst results, since it has almost a linear behavior (see Fig. 10). The *STRUCK* tracker presents competitive results with our method, while the *Coherent Filtering* shows poor performance. These results are in accordance with the qualitative results shown in Fig. 22.

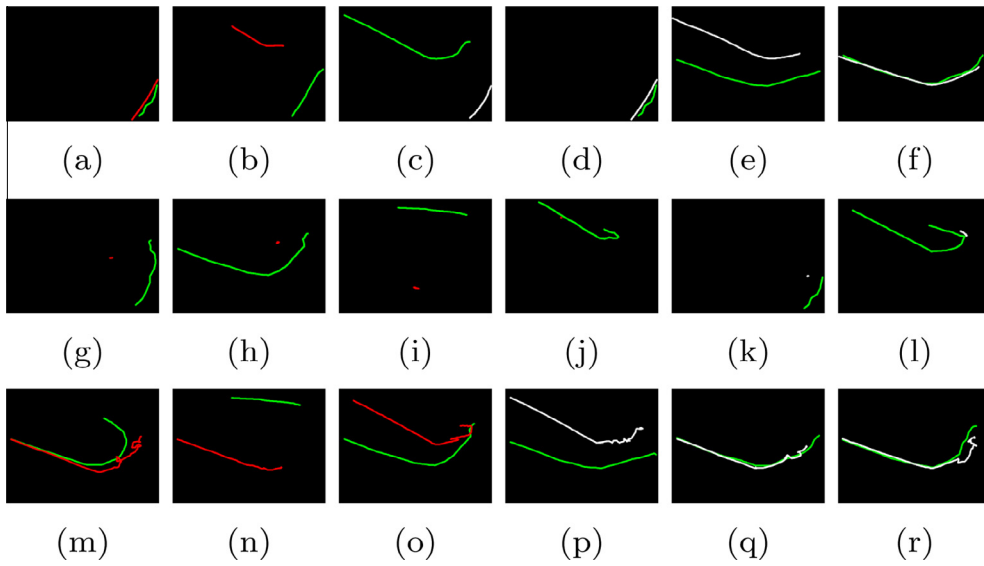


Fig. 11. Assignment using DTW metric: First row – our method ($minibatch = 2$, $memory = 10$): (a) false negative; (b) miss-match detected successfully; (c) false positive not detected; (d), (e), (f) matches detected; second row – *Coherent Filtering*: (g), (h), (i), (j) miss-match detected successfully; (k), (l) false positives not detected; and third row – *STRUCK*: (m) false negative; (n), (o) miss-matches detected successfully; (p) false positive not detected; (q), (r) matches detected.

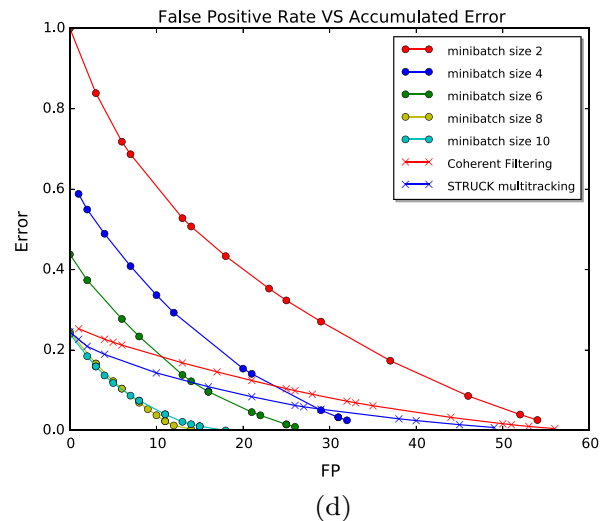
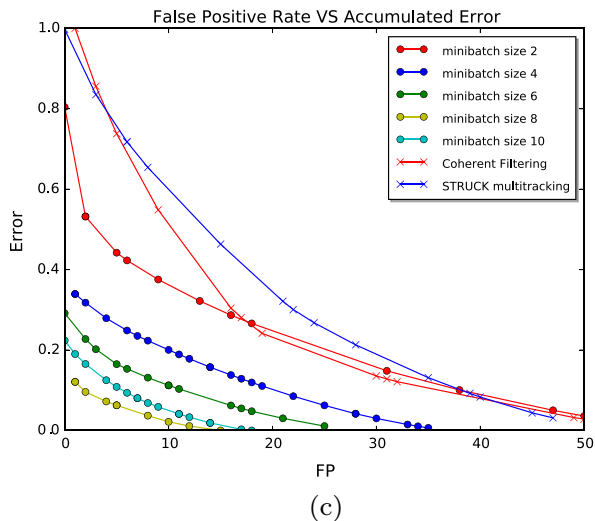
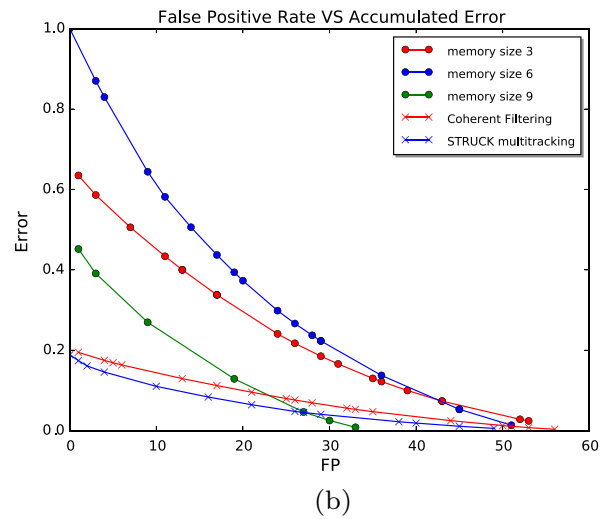
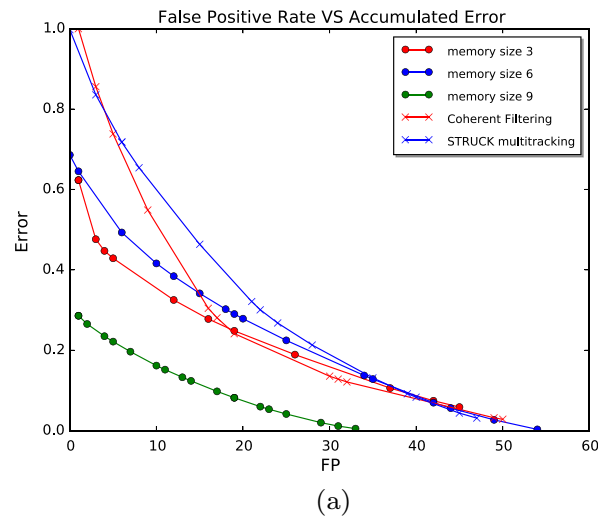


Fig. 12. FP rate VS error on D.C.S. scenario with quartile threshold regularization for: (a) *memory* variation and hausdorff-modified metric; (b) *memory* variation and chebyshev metric; (c) *minibatch* variation and hausdorff-modified metric; and (d) *minibatch* variation and chebyshev metric.

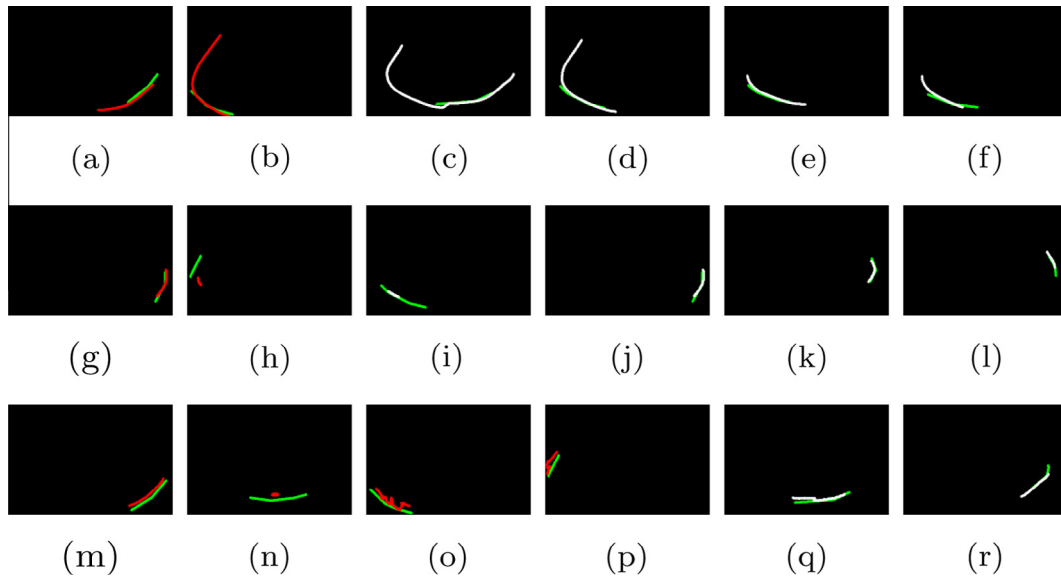


Fig. 13. Assignment using hausdorff-modified metric: First row – our method ($\text{minibatch} = 8$, $\text{memory} = 10$): (a) false negative; (b) miss-match detected successfully; (c) false positive not detected; (d), (e), (f) matches detected; second row – *Coherent Filtering*: (g) false negative; (h) miss-match detected successfully; (i) false positive not detected; (j), (k), (l) matches detected; and third row – *STRUCK*: (m), (p), (o) false negatives; (n) miss-match detected successfully; (q), (r) matches detected.

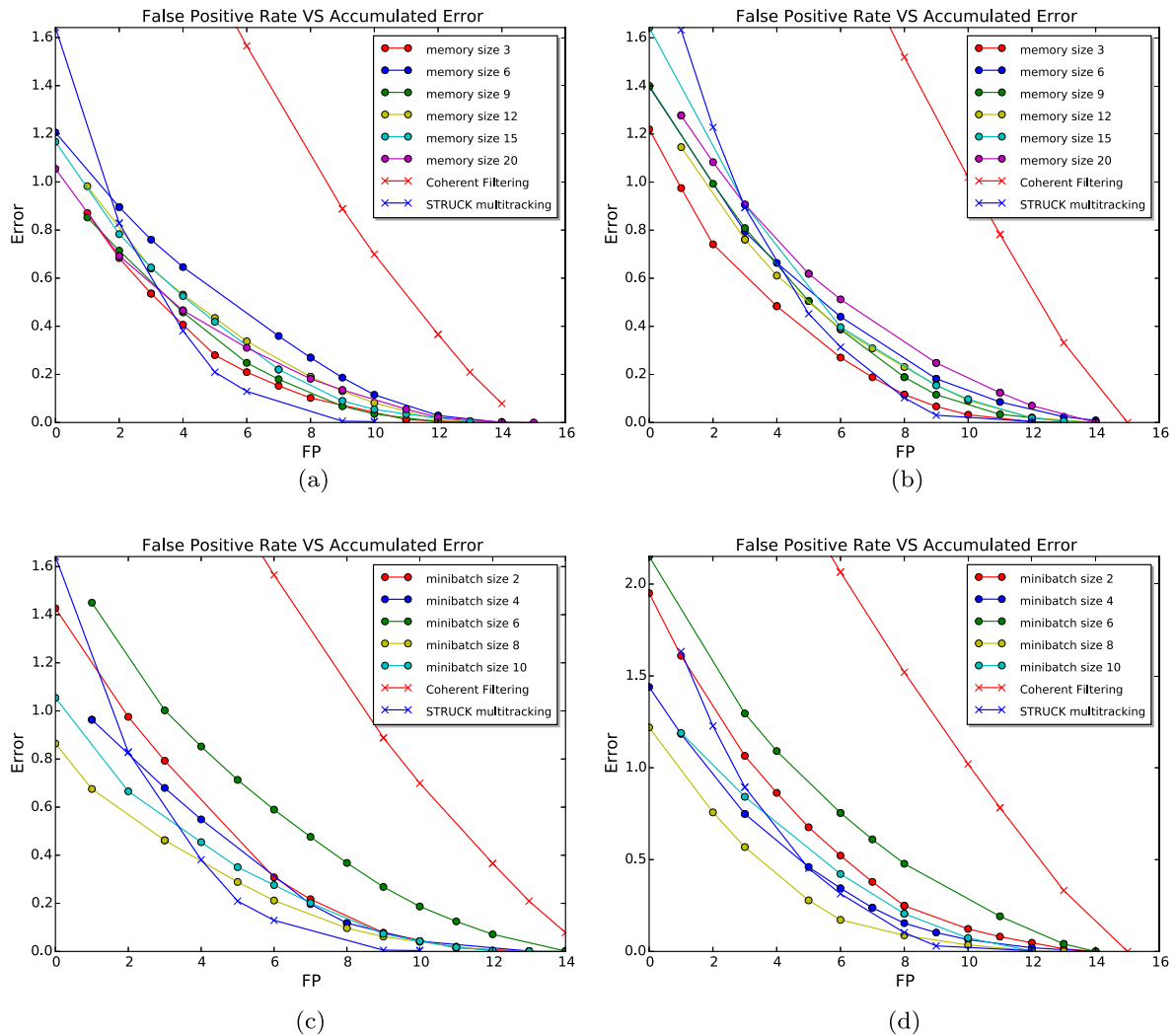


Fig. 14. FP rate VS error on S.C.U. scenario with median RLS regularization for: (a) *memory* variation and DTW metric; (b) *memory* variation and euclidean metric; (c) *minibatch* variation and DTW metric; and (d) *minibatch* variation and euclidean metric.

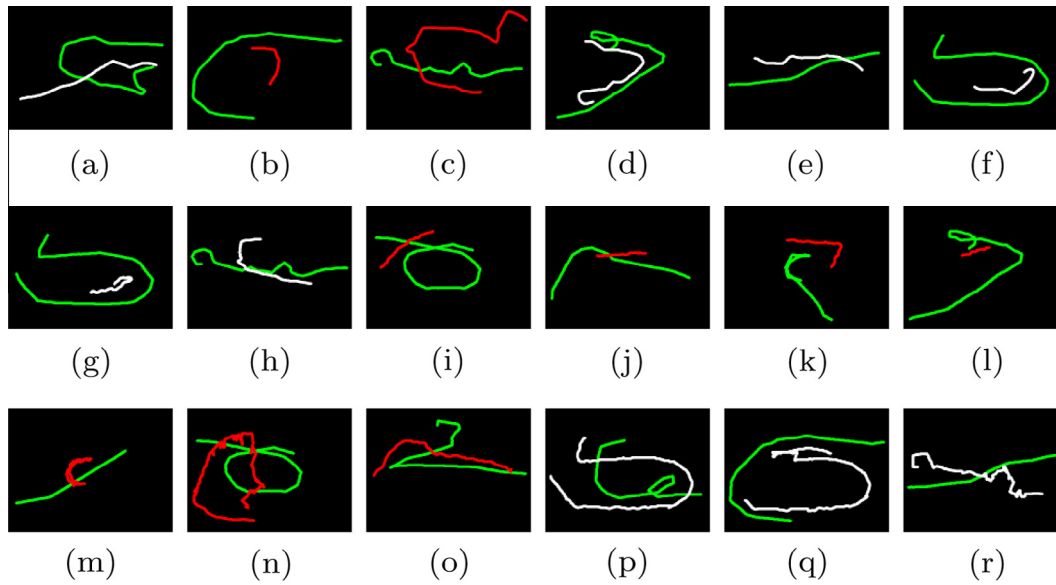


Fig. 15. Assignment using DTW metric: First row – our method (*minibatch* = 8, *memory* = 10): (a) false positive not detected; (b), (c) miss-matches detected successfully; (d), (e), (f) matches detected; second row – *Coherent Filtering*: (g), (h) false positive not detected; (i), (j), (k), (l) miss-matches detected successfully; and third row – *STRUCK*: (m), (n), (o) miss-matches detected successfully; (p), (q), (r) false positive not detected.

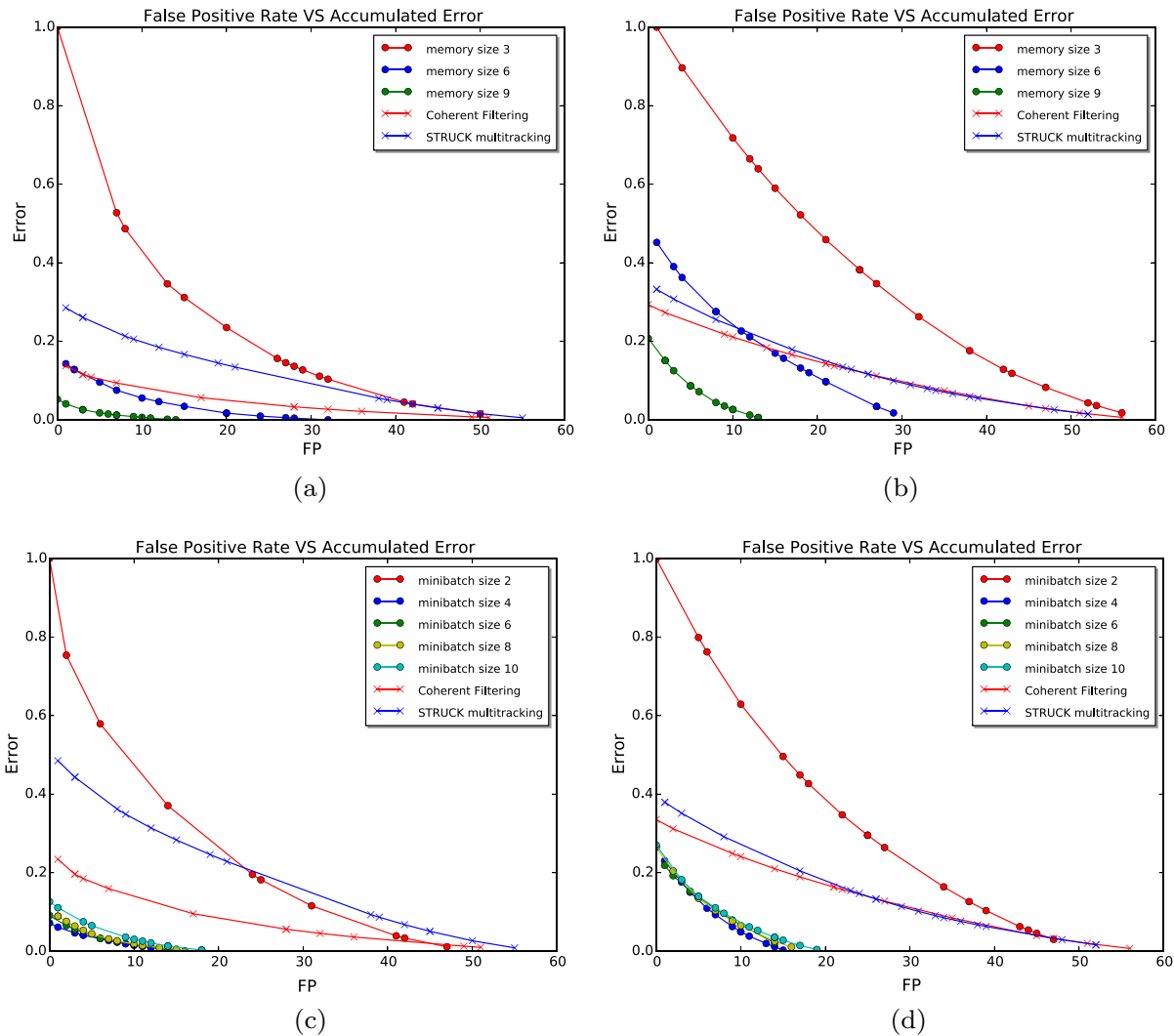


Fig. 16. FP rate VS error on D.C.U. scenario with quartile threshold regularization for: (a) *memory* variation and hausdorff-modified metric; (b) *memory* variation and chebyshev metric; (c) *minibatch* variation and hausdorff-modified metric; and (d) *minibatch* variation and chebyshev metric.

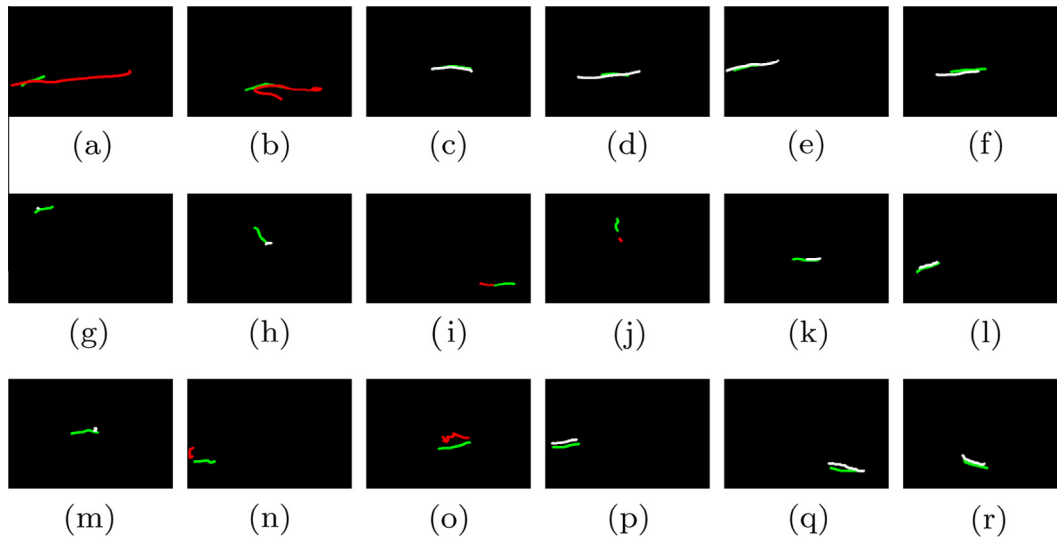


Fig. 17. Assignment using hausdorff-modified metric: First row – our method (*minibatch* = 4, *memory* = 10): (a), (b) miss-matches detected successfully; (c), (d), (e), (f) matches detected; second row – *Coherent Filtering*: (g), (h) false negatives; (i), (j) miss-matches detected successfully; (k), (l) matches detected; and third row – *STRUCK*: (m) false positive; (n), (o) miss-matches detected successfully; (p), (q), (r) matches detected.

Fig. 11 presents some matching results, where manual trajectories are represented by green,³ and auto-generated trajectories are illustrated by red if a false positive is considered, or by white if a correct match is assigned. This nomenclature is kept for subsequent figures. Fig. 11(a) and (d) shows the threshold influence when deciding if it is a false positive or not, the former is a false negative assignment, and the latter is a correct match. Fig. 11(c) presents a wrong match, probably due to the global minimization problem associated with the Hungarian algorithm. However, we state by Fig. 11(e) that our system is capable to generate a possible assignment for the previous miss match. We verify valid assignments and false positive detections. It is important to highlight that just some trajectories were annotated for this scenario, which has hundreds of trajectories that could be more similar to the automatic trajectories. The difficulty to obtain valid and long trajectories by the *Coherent Filtering* method is clearly shown in the second row of Fig. 11. The *STRUCK* tracker is able to obtain valid trajectories, but with some degree of noise.

6.3.2. Dense Crowd Structured (D.C.S.) scenario

This scenario represents a marathon, which is similar to a parade, therefore it is categorized as a dense crowded scene with structured motion. From the original sequence, 150 frames were manually annotated obtaining a total of 60 trajectories.

Clearly, both *memory cell* size and *minibatch* size benefit of large values to obtain low error rates, therefore their sizes should be kept as high as possible (see Fig. 12). The selected regularization was the quartile threshold. Regarding the distance measure, we verify that the hausdorff-modified and the chebyshev present well-defined curves to describe the ratio between FP and error while varying the threshold. However, we notice that the hausdorff-modified metric has a steepest behavior on low thresholds. It also shows that our method outperforms both comparative methods. The chebyshev metric confirms the supremacy of our method for large *minibatch* sizes, while it also stated that comparative methods could perform better than our method if no correct *memory* size is selected. Results for *memory* size superior than 9 are not shown, since not enough information can be acquired from the small number of frames of this dataset

to reliably represent the behavior of our method under such parameterization.

This scenario is highly crowded with well-defined common trajectories, most of them following the same path (u-form with the same source and sink regions, see Fig. 22). Due to the small number of frames, the manual annotation does not include the true long common paths that our method is able to extract. Instead, it represents individual pedestrians within a small temporal gap. Fig. 13 illustrates this observation, where longer trajectories are obtained through our method and are correctly matched, and more similar trajectories to the manual ones are obtained through *Coherent Filtering* and *STRUCK* methods.

6.3.3. Sparse Crowd Unstructured (S.C.U.) scenario

This is the most challenging scenario for our system. It is a low resolution video that represents a crowded scene where pedestrians move randomly in various directions, causing constant occlusions. We manually annotated all the pedestrians, which lead to a total of 15 trajectories.

In this case, *memory cell* size has a less stable behavior assuming the relation between the error and the FP rate. However, in general lower values present better results. *Minibatch* size has even more variability, and we verify that medium and large values perform better than lower values. The selected regularization was the median RLS. In terms of distance function, we reached the same conclusion of the S.C.S. scenario. Once again, our method, with the correct selected parameters, shows superior performance than both comparative algorithms, especially with the *Coherent Filtering* method (see Fig. 14).

Fig. 15 permits to conclude that this scenario is challenging for all the methods. We verified that the assignment method is not effective in some cases, e.g. Fig. 15(q) presents an automatic trajectory similar to the manual trajectory in Fig. 15(f) and they were not matched. After visual inspection of the trajectories extracted by our system, we verified that the most similar were not matched to the manual ones. This factor leads us to conclude that, despite the larger matching differences, the system performs well on such demanding scenario.

6.3.4. Dense Crowd Unstructured (D.C.U.) scenario

This scenario is similar to the previous one, with the advantage that the global motion is denser and less random. However, the

³ For interpretation of color in Fig. 11, the reader is referred to the web version of this article.

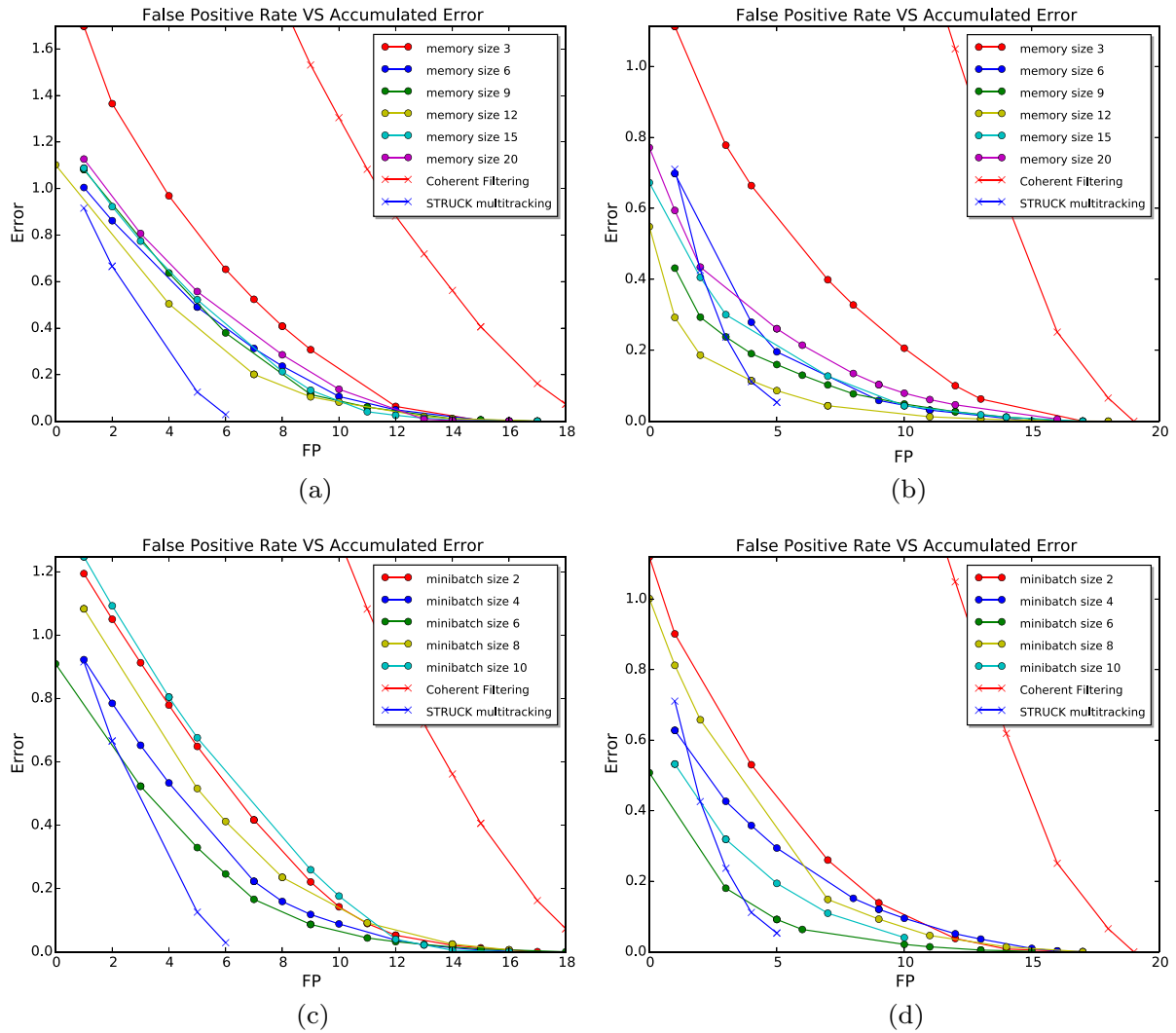


Fig. 18. FP rate VS error on D.M.T. scenario with median RLS regularization for: (a) *memory* variation and DTW metric; (b) *memory* variation and hausdorff metric; (c) *minibatch* variation and DTW metric; and (d) *minibatch* variation and hausdorff metric.

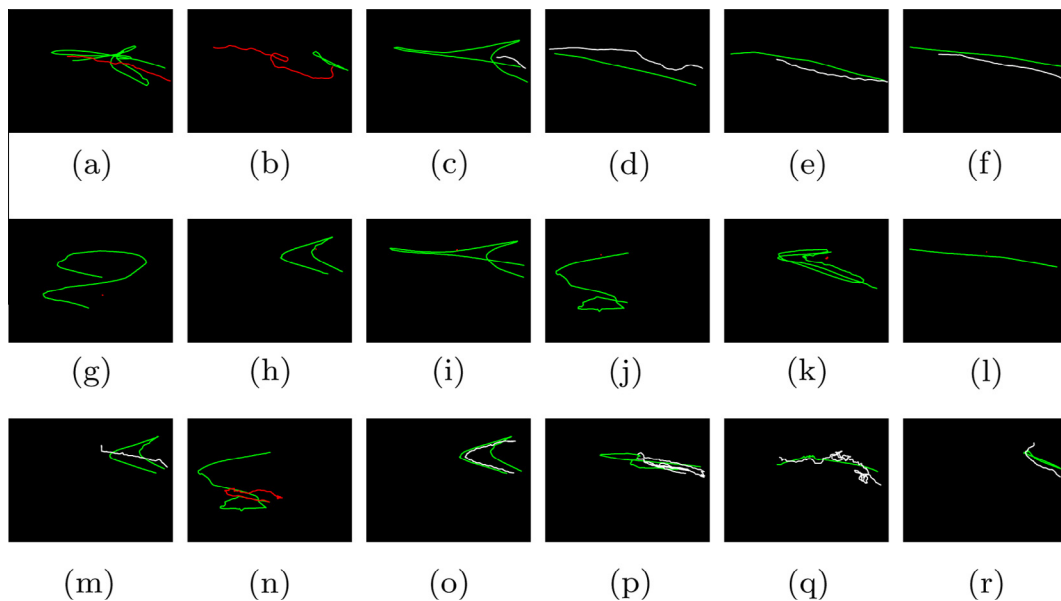


Fig. 19. Assignment using hausdorff metric: First row – our method (*minibatch* = 5, *memory* = 12): (a), (b) miss-matches detected successfully; (c) false positive not detected; (d), (e), (f) matches detected; second row – *Coherent Filtering*: (g), (h), (i), (j), (k), (l) miss-matches detected successfully; and third row – *STRUCK*: (m) false positive not detected; (n) miss-match detected successfully; (o), (p), (q), (r) matches detected.

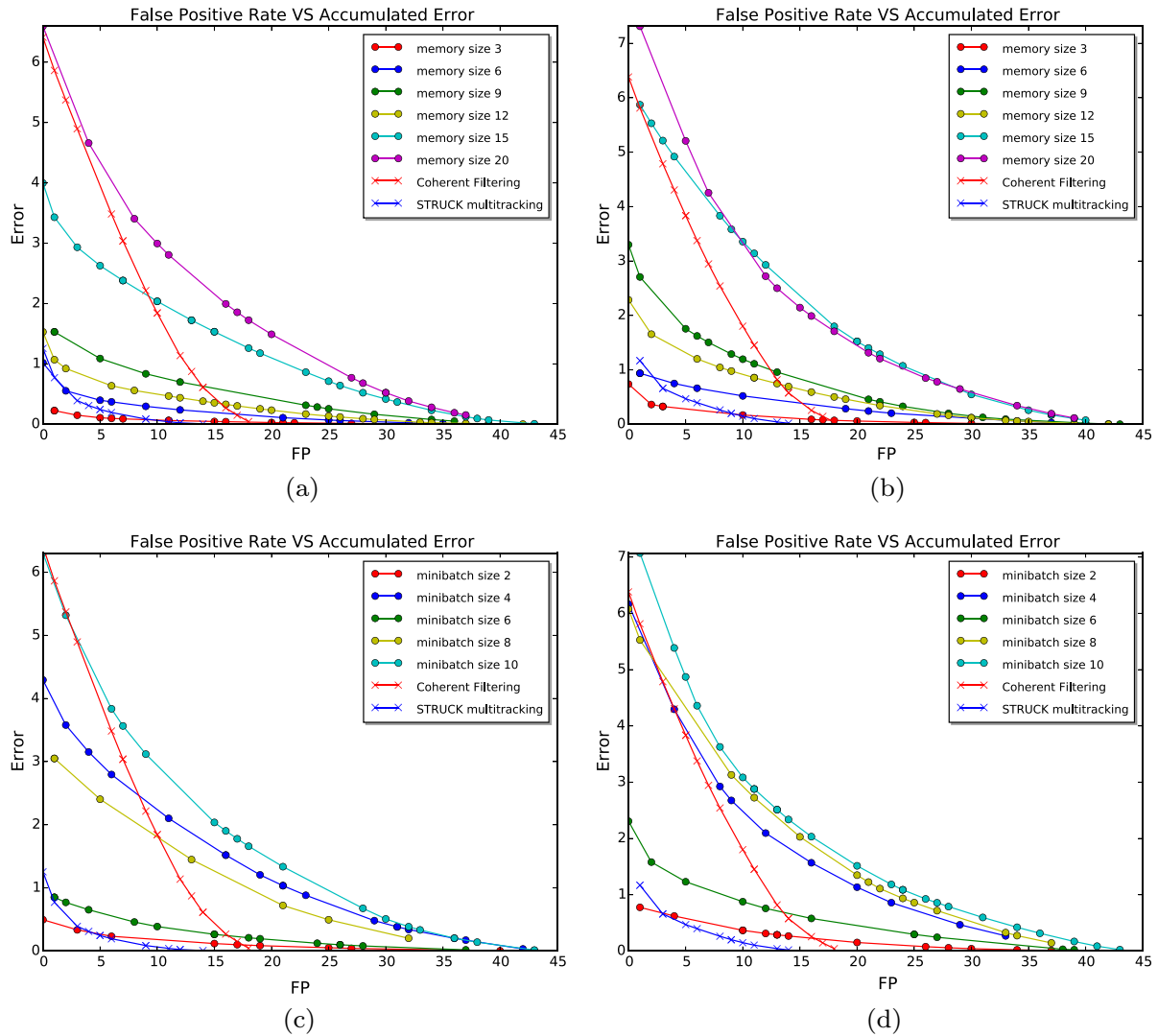


Fig. 20. FP rate VS error on S.M.T. scenario with median RLS regularization for: (a) *memory variation and DTW metric*; (b) *memory variation and euclidean metric*; (c) *minibatch variation and DTW metric*; and (d) *minibatch variation and euclidean metric*.

camera is far away from the pedestrians, which hinders proper motion tracking. It represents an outdoor highly crowded scenario with 67 frames and a total of 60 manually annotated trajectories.

As well as the D.C.S scenario, Fig. 16 does not present results for *memory size* superior than 9, the quartile threshold regularization and both hausdorff-modified and chebyshev metrics were selected. The behavior of our method is also similar to the one presented in the D.C.S scenario, with the exception that the variation of the best *minibatch size* is lower, showing it best performance for medium values.

Simultaneous inspection of Figs. 17 and 22 shows that the three methods are able to extract short trajectories that represent individual pedestrians slowly walking through the crowd. Our method keeps its supremacy results over the comparative algorithms, and as well is able to obtain longer trajectories through temporal integration. This is the only scenario where the *Coherent Filtering* presents better performance than the *STRUCK*, which makes sense since it was created for this type of scenario considering coherent motions within the crowd.

6.3.5. Dense Multi-Tracking (D.MT.) scenario

This scenario is also very challenging. Normally, multi-tracking approaches are applied to it and flow-based approaches conduct to poor results. There are 19 manual trajectories to match.

An analysis shows that low values for *memory cell* and *minibatch size* decrease performance, while medium values perform better. The selected regularization was also the median RLS. Regarding the distance functions, previous conclusions still hold true. The Hausdorff metric reaches the best performance, which presents a low and steeper error for low false positive rates. Results illustrated in Fig. 18 reveals the poor performance of the *Coherent Filtering* in multi-tracking scenarios, while shows the most competitive results of *STRUCK* when compared with our method.

Fig. 19 shows the most difficult trajectories to match in this dataset. In fact, Fig. 19(a) presents a complex manual trajectory, which is very long and has several direction changes. However, it could be divided on two cross sections, where one of them is automatically captured by our system, nevertheless it could not be identified as a correct match. Despite the good results obtained by the *STRUCK* method, Fig. 19 reveals that our method is still able to extract similar trajectories to pedestrian movements.

6.3.6. Sparse Multi-Tracking (S.MT.) scenario

This is another scenario where multi-tracking approaches are normally applied. This dataset has an additional difficulty related to illumination variations that could lead to erroneous flow calculation. There are 44 manual trajectories.

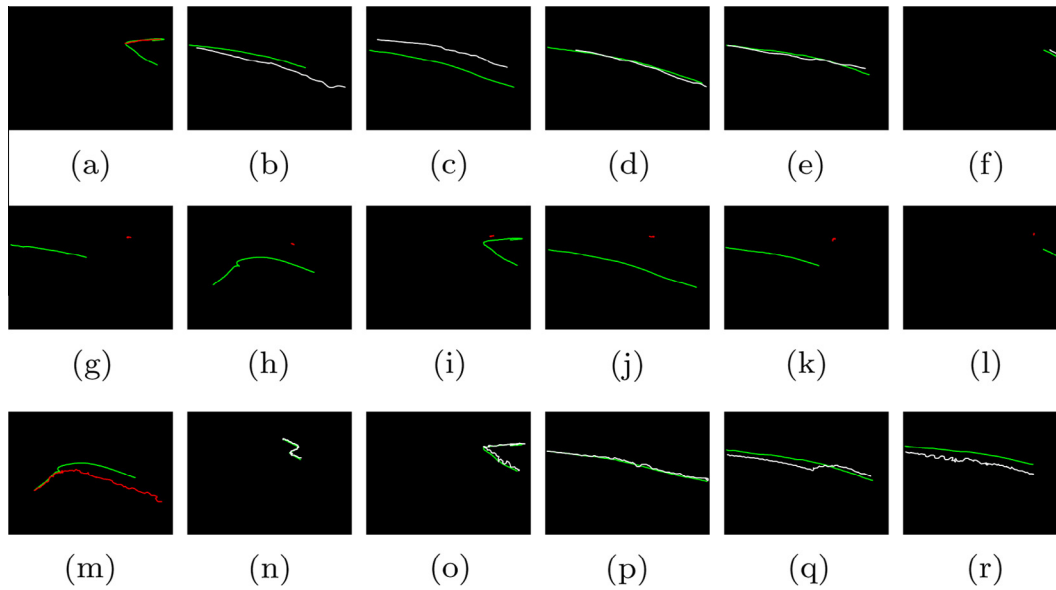


Fig. 21. Assignment using DTW metric: First row - our method ($minibatch = 5$, $memory = 3$): (a) miss-match detected successfully; (b), (c), (d), (e), (f) matches detected; second row - *Coherent Filtering*: (g), (h), (i), (j), (k), (l) miss-matches detected successfully; and third row - *STRUCK*: (m) miss-match detected successfully; (n), (o), (p), (q), (r) matches detected.

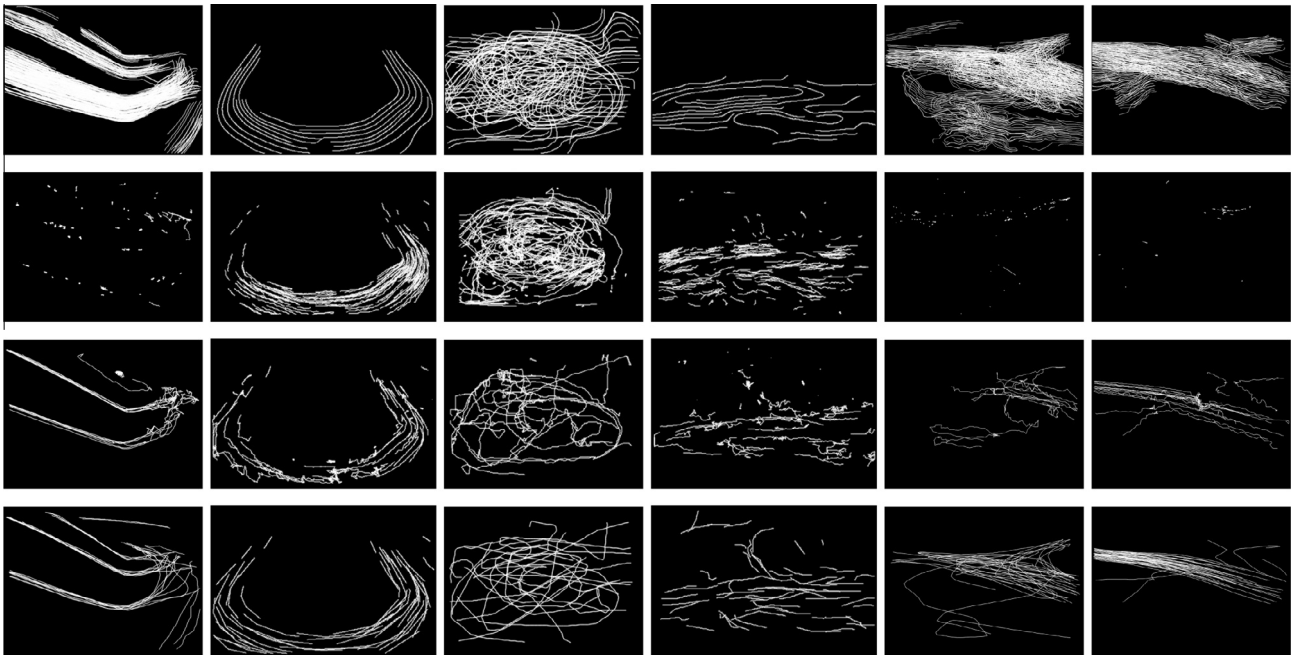


Fig. 22. Comparison between all automatic extracted trajectories and manually annotated trajectories. From top-to-bottom: First row: out method; Second row: *Coherent Filtering*; Third row: *STRUCK*; and Fourth row: manual.

The performance dependence on *memory cell* and *minibatch* is as in the S.C.S. scenario. The regularization of the distance matrix was based on median RLS, and the DTW was, again, the metric which presented the best results (see Fig. 20).

As expected, for this sparse multi-tracking scenario we verified even better performance of the *STRUCK* algorithm, with a lower margin of parameter's optimization for our method. However, this dataset presents the best results (see Fig. 21), which is an excellent conclusion since it proves the versatility of the system. It correctly captures short and long trajectories, as well as false positives.

However, we notice that the matches presented on Fig. 21(b) and (c) could be inversely assigned to obtain a better matching.

As general evaluation, we conclude that low values should be considered for *memory cell* size, with the exception of crowd scenes where higher values should be used, while *minibatch* size can be varied from low to medium values depending on scene category. Therefore, *minibatch* size presents a larger functional interval, while avoiding error propagation derived from numeric calculations of the motion advection scheme. The lower values for *memory cell* size could be explained by the intuition that it

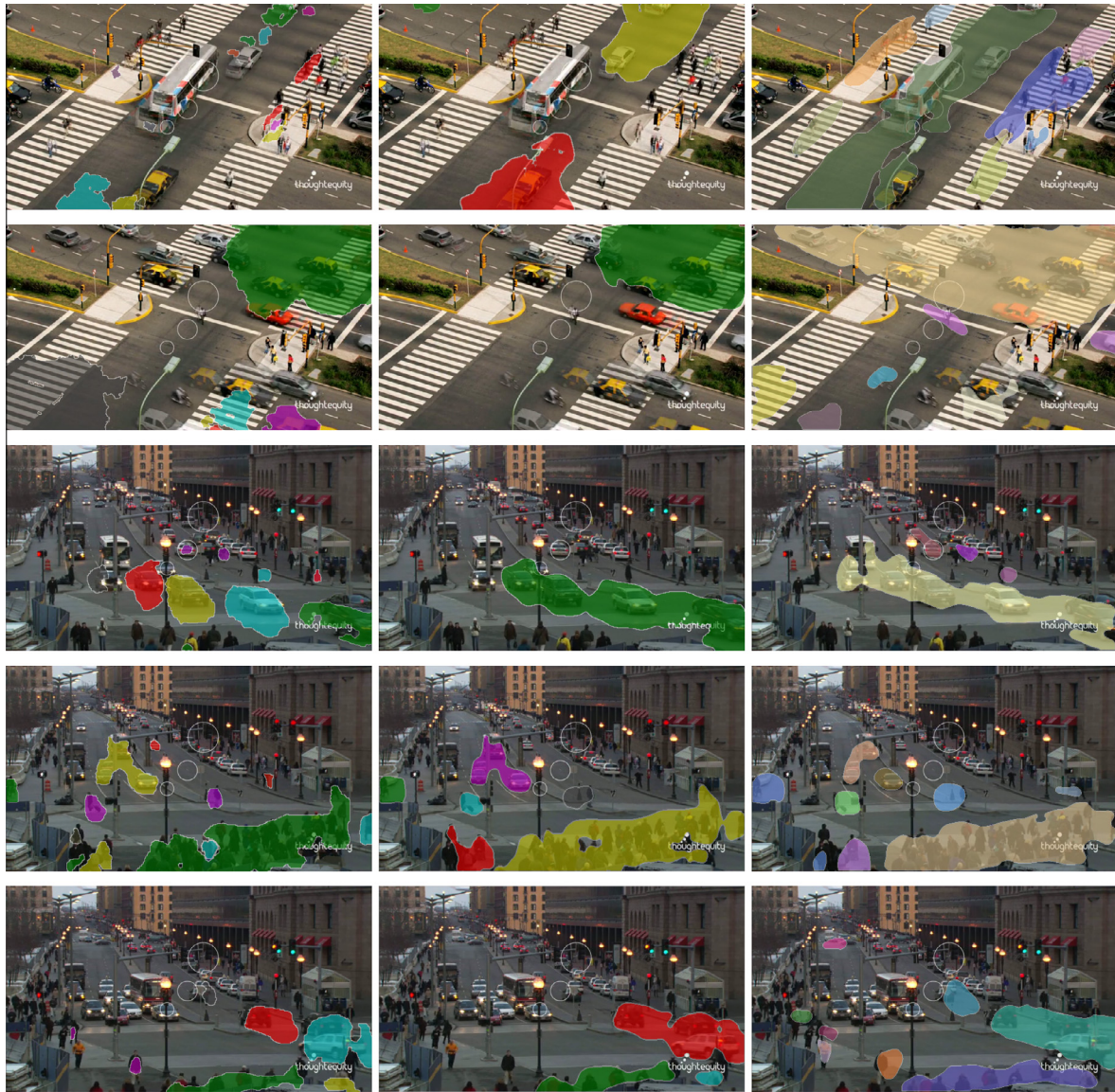


Fig. 23. Qualitative comparison of segmentation results in Argentina and Boston datasets. By row: frame 115 (1st row), and frame 213 (2nd row) of Argentina; frame 40 (3rd row), frame 433 (4th row), and frame 2042 (5th row) of Boston. By column: from left to right, pathlines, streaklines, and our approach.

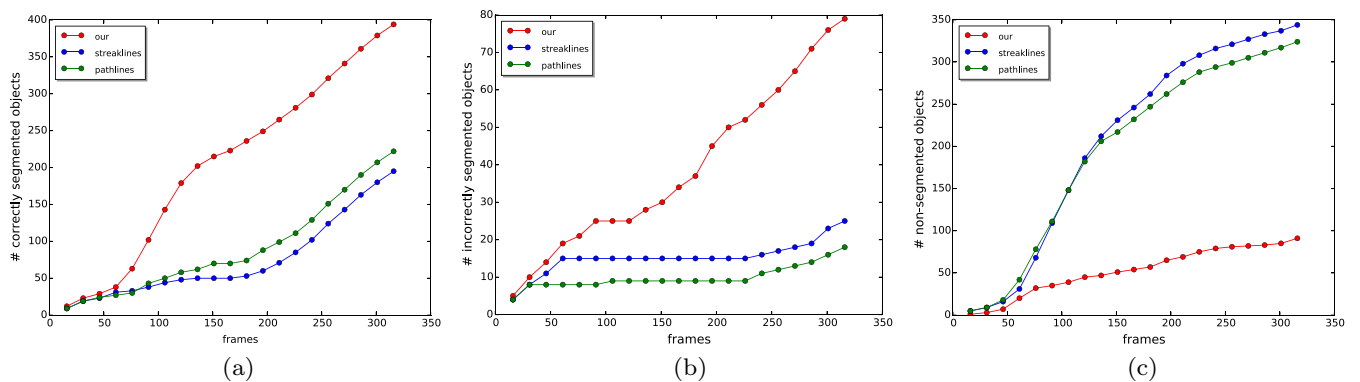


Fig. 24. Quantitative comparison of segmentation results in Argentina dataset: (a) correctly segmented objects; (b), incorrectly segmented objects; and (c), non-segmented objects.

permits to capture less linear trajectories, eliminating the possibility to miss the detection of trajectory segments with high curvature, since the linking process could favor direction continuity. In terms of regularization, it is obvious to conclude that a non-linear technique improves the results. In terms of the distance function, we verify that metrics that consider point-to-point geometrical information and shape benefit the matching process. Comparing our method with two competitive algorithms based on micro and macro approaches of motion, we reinforce its versatility, accuracy and pertinence for several surveillance scenarios.

For qualitative validation, Fig. 22 presents the spatiotemporal comparison between the extracted trajectories and the manual ones. The density obtained clearly shows the robustness and efficiency of our system, especially when compared with *Coherent Filtering* and *STRUCK* methods. Is important to mention that the assignment process always finds a correspondence between a manual trajectory and an auto-generated one, which corroborates the density evaluation.

6.4. Motion segmentation

This section reports the usefulness of our system related to human activity tasks. In this case, we evaluate our motion segmentation performance against two state-of-the-art works [10,43]. We follow both the qualitative and quantitative analysis approach of [10] to present and compare the dynamic segmentations of different behavior states through a video sequence. For the qualitative analysis, we used the same datasets as in [10], namely Argentina and Boston.⁴ However, for the quantitative evaluation we do not report the results in the Boston dataset because it is a very time-consuming task to annotate all motion objects (14130 frames). We should highlight that we were unable to reproduce the results in [10], even using the author's source code and the same parameterisation.⁵

For the experiments, we use the following parameters for each approach: (i) pathlines, *integration time* = 15; (ii) streaklines, *streak length* = 40; (iii) our, *minibatch* = 3, *memory* = 5. In order for the comparison to be as fair as possible we use the same optical flow result per pair of frames for each approach, namely the Classic + NL from [44].

Frames 115 and 213 illustrate two behavioral phases in which traffic lights change and north/south flow of pedestrians emerges, and west/east and north/south bound vehicles flow develop. Remaining frames refer to Boston dataset and also consider different behaviors when traffic light changes an east/west flow of pedestrians emerges simultaneously with a north/south bound vehicle flow. Fig. 23 demonstrates that streaklines are spatially and temporally pronounced and more accurate on capturing dynamic objects than the pathlines, which show fragmented segments of movement. However, our approach clearly shows longer motions and robustly segments different flows, even on cluttered conditions. For instance, our approach is able to detect flow of pedestrians on each sidewalks and distinguish them from north and south bound (frame 115), and detect and separate standalone motions (frames 40, 433 and 2042). None of the other approaches are able to do this, even inspecting the results reported in [10]. We clarify that our segmentation approach is computed per pixel, considering the similarity of the cosine difference in a 8-connected neighborhood, i.e. it follows the same process of the streak flow similarity of [10]. In our case, the flow is derived from the long-range trajectories, where each one is resampled and their segments are considered as the flow vectors.

For the quantitative comparison we also follow the same criterion stated in [10]. Our approach outperforms by a large margin, more than twice, both state-of-the-art approaches in the number of correctly segmented objects, even considering the results reported in [10]. The number of non-segmented objects is less than three times of the reported by the remaining approaches, which corroborates the qualitative analysis. However, it also presents a higher number of incorrectly segmented objects, probably because an over-segmentation on clutter regions (see Fig. 24).

7. Conclusion

We present a complete and novel system based on global dense flow and local motion information that extracts meaningful long-range trajectories. Its main advantages are its efficiency on different scenarios, and the correlation between the extracted trajectories and the individual pedestrians movements. The system combines a macro and micro analysis of motion, that to the best of our knowledge was not explored before. This work presents important contributions on different stages, namely on the system formulation, motion extraction and propagation scheme, a novel technique for removal of flow vector outliers, a fine-to-coarse flow representation, and integration of local motion information into a global re-correlation algorithm at the tracklet-level to form long-range trajectories. We also show that the system provides robust spatiotemporal motion information that can be used for human activity tasks such as segmentation, where it outperforms state-of-the-art algorithms.

An evaluation framework is proposed to deal with the matching problem between global and individual trajectories. We demonstrate very good results on different datasets, outperforming micro and macro state-of-the-art approaches. However, two aspects should be improved: the false positive rate, since sometimes they are wrongly detected; and the assignment process, since in some cases the system produces most similar trajectories than the ones that were assigned by the matching process.

More sequences should be used to optimize system's parameters for each scene context. In that way, we believe that the system could get even better results. For future work, we will conduct a deeper system component analysis, explore the fine-to-coarse representation to model different level of human-activity-related knowledge, and will use the trajectories extracted from other scene contexts, such as duo interaction and individual tracking, to verify their impact on action classification. We believe that detection and classification of human activity could benefit from a system like ours.

Acknowledgment

The first author would like to thank FCT - Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) for the financial support for the PhD grant with reference SFRH/BD/51430/2011.

References

- [1] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, *Image Vision Comput.* 14 (8) (1996) 609–615, [http://dx.doi.org/10.1016/0262-8856\(96\)01101-8](http://dx.doi.org/10.1016/0262-8856(96)01101-8).
- [2] X. Wang, K. Tieu, E. Grimson, Learning semantic scene models by trajectory analysis, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), *ECCV* (3), *Lecture Notes in Computer Science*, vol. 3953, Springer, 2006, pp. 110–123.
- [3] X. Wang, K.T. Ma, G.-W. Ng, W.E. Grimson, Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models, *Int. J. Comput. Vision* 95 (3) (2011) 287–312, <http://dx.doi.org/10.1007/s11263-011-0459-6>.
- [4] J. Sun, Y. Mu, S. Yan, L.F. Cheong, Activity recognition using dense long-duration trajectories, in: *ICME, IEEE*, 2010, pp. 322–327.

⁴ http://www.cs.ucf.edu/ramin/?page_id=99

⁵ We acknowledge and thank the efforts of the first author of [10] to try to overcome such discrepancy, although without success.

- [5] M. Raptis, S. Soatto, Tracklet descriptors for action modeling and video analysis, in: *Proceedings of the 11th European Conference on Computer Vision: Part I, ECCV'10*, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 577–590.
- [6] H. Wang, A. Kläser, C. Schmid, L. Cheng-Lin, Action recognition by dense trajectories, in: *IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, United States, 2011, pp. 3169–3176. <<http://hal.inria.fr/inria-00583818>>.
- [7] P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: action recognition through the motion analysis of tracked features, in: *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009, IEEE, 2009, pp. 514–521.
- [8] S. Ali, M. Shah, A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, CVPR '07, 2007, pp. 1–6, <http://dx.doi.org/10.1109/cvpr.2007.382977>.
- [9] M. Hu, S. Ali, M. Shah, Learning motion patterns in crowded scenes using motion flow field, in: *ICPR, IEEE*, 2008, pp. 1–5.
- [10] R. Mehran, B.E. Moore, M. Shah, A streakline representation of flow in crowded scenes, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), *ECCV (3)*, Lecture Notes in Computer Science, vol. 6313, Springer, 2010, pp. 439–452.
- [11] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, 2013 IEEE Conference on Computer Vision and Pattern Recognition 0 (2009) 1446–1453, <http://dx.doi.org/10.1109/CVPRW.2009.5206771>.
- [12] X. Zhao, D. Gong, G. Medioni, Tracking using motion patterns for very crowded scenes, in: *Proceedings of the 12th European Conference on Computer Vision - Volume Part II, ECCV'12*, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 315–328, http://dx.doi.org/10.1007/978-3-642-33709-3_23.
- [13] J. Kim, K. Grauman, Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates, in: *CVPR*, 2009, pp. 2921–2928.
- [14] S. Ali, M. Shah, Floor fields for tracking in high density crowd scenes, in: *ECCV*, 2008, pp. 1–14.
- [15] M. Hu, S. Ali, M. Shah, Detecting global motion patterns in complex videos, in: *ICPR*, 2008, pp. 1–5.
- [16] O. Ozturk, T. Yamasaki, K. Aizawa, Detecting dominant motion flows in unstructured/structured crowd scenes, in: *Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10*, IEEE Computer Society, Washington, DC, USA, 2010, pp. 3533–3536, <http://dx.doi.org/10.1109/ICPR.2010.862>.
- [17] B. Zhou, X. Tang, X. Wang, Coherent filtering: detecting coherent motions from crowd clutters, in: *Computer Vision–ECCV 2012*, Springer, 2012, pp. 857–871.
- [18] B. Zhou, X. Wang, X. Tang, Understanding collective crowd behaviors: learning a mixture model of dynamic pedestrian-agents, in: *CVPR*, 2012, pp. 2871–2878.
- [19] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 935–942, <http://dx.doi.org/10.1109/cvprw.2009.5206641>.
- [20] B. Jobard, G. Erlebacher, M.Y. Hussaini, Lagrangian-Eulerian advection for unsteady flow visualization, in: *Proceedings of the Conference on Visualization'01*, IEEE Computer Society, 2001, pp. 53–60.
- [21] K. Polthier, E. Preuß, Variational Approach to Vector Field Decomposition, Springer, 2000.
- [22] T. Weinkauff, H. Theisel, Streak lines as tangent curves of a derived vector field, *IEEE Trans. Visual. Comp. Graph.* 16 (6) (2010) 1225–1234, <http://dx.doi.org/10.1109/TVCG.2010.198>.
- [23] V. Verma, D. Kao, A. Pang, A flow-guided streamline seeding strategy, in: *Proceedings of the Conference on Visualization '00, VIS '00*, IEEE Computer Society Press, Los Alamitos, CA, USA, 2000, pp. 163–170.
- [24] B. Jobard, W. Lefer, Creating evenly-spaced streamlines of arbitrary density, in: *Eurographics Workshop*, Springer Verlag, 1997, pp. 43–56.
- [25] K. Wu, Z. Liu, S. Zhang, R.J. Moorhead II, Topology-aware evenly spaced streamline placement, *IEEE Trans. Visual. Comp. Graph.* 16 (5) (2010) 791–801, <http://dx.doi.org/10.1109/TVCG.2009.206>.
- [26] J. Lezama, K. Alahari, J. Sivic, I. Laptev, Track to the future: spatio-temporal video segmentation with long-range motion cues, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [27] G. Farneback, Two-frame motion estimation based on polynomial expansion, in: *Proceedings of the 13th Scandinavian Conference on Image Analysis, SCIA'03*, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 363–370.
- [28] T. Brox, J. Malik, Large displacement optical flow: descriptor matching in variational motion estimation, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (3) (2011) 500–513, <http://dx.doi.org/10.1109/TPAMI.2010.143>.
- [29] E.M. Pereira, J.S. Cardoso, R. Morla, Motion flow tracking in unconstrained videos for retail scenario, in: J.A. Sanches, L. Micó, J.S. Cardoso (Eds.), *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*, vol. 7887, Springer, Berlin, Heidelberg, 2013, pp. 340–349.
- [30] N. Sundaram, T. Brox, K. Keutzer, Dense point trajectories by GPU-accelerated large displacement optical flow, in: *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, Springer, 2010.
- [31] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1027–1035.
- [32] T. Weinkauff, H. Theisel, Streak lines as tangent curves of a derived vector field, *IEEE Trans. Visual. Comp. Graph.* 16 (6) (2010) 1225–1234, <http://dx.doi.org/10.1109/TVCG.2010.198>.
- [33] A. Mebarki, P. Alliez, O. Devillers, Farthest point seeding for efficient placement of streamlines, in: *Visualization*, 2005. VIS 05. IEEE, IEEE, 2005, pp. 479–486.
- [34] M. Rubinstein, C. Liu, W. T. Freeman, Towards longer long-range motion trajectories (2012) 53.1–53.11, doi:<http://dx.doi.org/10.5244/C.26.53>.
- [35] R. Thompson, A note on restricted maximum likelihood estimation with an alternative outlier model, *J. R. Statist. Soc., Series B (Methodological)* 47 (1) (1985) 53–55.
- [36] M. Hubert, E. Vandervieren, An adjusted boxplot for skewed distributions, *Comput. Stat. Data Anal.* 52 (12) (2008) 5186–5201, <http://dx.doi.org/10.1016/j.csda.2007.11.008>.
- [37] A. Kimber, Tests for many outliers in an exponential sample, *J. R. Statist. Soc. Series C (Appl. Statist.)* 31 (3) (1982) 263–271.
- [38] G.B. Frank E. Grubbs, Extension of sample sizes and percentage points for significance tests of outlying observations, *Technometrics*, doi:<http://dx.doi.org/10.2307/1267134>.
- [39] S. Hare, A. Saffari, P.H. Torr, Struck: Structured output tracking with kernels, in: *IEEE International Conference on Computer Vision (ICCV)*, 2011, IEEE, 2011, pp. 263–270.
- [40] P. Ochs, J. Malik, T. Brox, Segmentation of moving objects by long term video analysis, *IEEE Trans. Patt. Anal. Mach. Intell.* 36 (6) (2014) 1187–1200.
- [41] H.W. Kuhn, The hungarian method for the assignment problem, *Naval Res. Logist. Quart.* 2 (1955) 83–97.
- [42] L. Zelnik-manor, P. Perona, Self-tuning spectral clustering, *Advances in Neural Information Processing Systems*, vol. 17, MIT Press, 2004, pp. 1601–1608.
- [43] S. Ali, M. Shah, A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis, in: *CVPR*, IEEE Computer Society, 2007.
- [44] D. Sun, S. Roth, M. Black, Secrets of optical flow estimation and their principles, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, 2010, pp. 2432–2439, <http://dx.doi.org/10.1109/CVPR.2010.5539939>.