Detecting Bipedal Motion from Correlated Probabilistic Trajectories

Atsuto Maki[a,*], Frank Perbet[a], Björn Stenger[a], Roberto Cipolla[a,b]

[a] *Cambridge Research Laboratory, Toshiba Research Europe*
208 Science Park, Cambridge CB4 0GZ, United Kingdom
Phone:+44 1223 436900, Fax:+44 1223 436909
[b] *Department of Engineering, University of Cambridge, CB2 1PZ, United Kingdom*

Abstract

This paper is about detecting bipedal motion in video sequences by using point trajectories in a framework of classification. Given a number of point trajectories, we find a subset of points which are arising from feet in bipedal motion by analysing their spatio-temporal correlation in a pairwise fashion. To this end, we introduce *probabilistic trajectories* as our new features which associate each point over a sufficiently long time period in the presence of noise. They are extracted from directed acyclic graphs whose edges represent temporal point correspondences and are weighted with their matching probability in terms of appearance and location. The benefit of the new representation is that it practically tolerates inherent ambiguity for example due to occlusions. We then learn the correlation between the motion of two feet using the probabilistic trajectories in a decision forest classifier. The effectiveness of the algorithm is demonstrated in experiments on image sequences captured with a static camera, and extensions to deal with a moving camera are discussed.

Keywords: motion, trajectories, spatio-temporal features, decision forest

1. Introduction

Point motion in an image sequence not only gives strong cues about the underlying geometry in 3D space, but may also be characteristic for an ob-

*Corresponding author
Email address: atsuto.maki@crl.toshiba.co.uk (Atsuto Maki)

ject class to which the point belong. Further, given a multiple of 2D point patterns, we can infer far more information than we would obtain from a single point trajectory through the correlation between the motion patterns. Hence, trajectories of points in image sequences provide a strong visual cue, often allowing the human brain to infer the scene behind those points. For example, when points close to the joints of a walking person are tracked, the psychological effect of *kinetic depth* allows us to perceive walking motion solely from the 2D point motion pattern. This has first been studied by Johansson using moving light displays (MLDs) (Johansson, 1973). The goal in this paper is to achieve this recognition ability for detecting pedestrian motion from tracked points on a pair of feet whose trajectories are characteristic and spatio-temporally correlated.

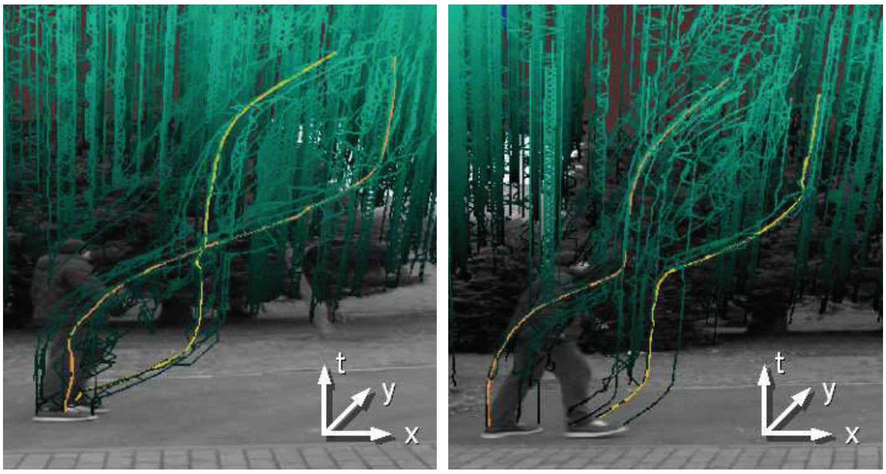


Figure 1: **Space-time volume with point trajectories.** *See the supplementary video for further explanations of space-time representation. Brighter green indicates corner positions more recent in time. Left: Two sample trajectories of corners on the feet are highlighted in yellow. Right: Another case where features are swapped during the short occlusion. Our method is able to correctly classify both cases as walking motion.*

The work presented in this paper can be categorized as motion-based recognition, but is unique in the sense that we do not assume clean point tracks. In order to obtain a discriminative trajectory, in this application, a point should ideally be tracked during a complete walk cycle (about one second). However, point trajectories of typical outdoor scenes are rarely reliable over such a long period of time. Therefore, we retain the concept of temporal connectedness by introducing the notion of *probabilistic trajectories*. See Figure 1 for an example of such trajectories. These are sampled from a directed acyclic graph whose edges represent temporal point correspondences weighted by their matching probability in terms of appearance and location. We choose to use standard corner features (Harris and Stephens, 1988) for the points to track rather than space-time interest points (Laptev and Lindeberg, 2003) in order to continue detecting points even when they are stationary during the walk cycle. Temporal correspondence is thus hypothesized while sacrificing matching accuracy in order to obtain longer and more discriminative trajectories. The advantage of this representation is that it permits inherent trajectory ambiguity, for example due to occlusions, and practically gain richer representations of the scene which facilitate the tasks of recognition using motion.

Physical models of bipedal motion have recently been used in tracking a walking person (Brubaker et al., 2010). Intuitively, the motion of a point on a single foot is composed of two periods of dynamic and static phases (Bissacco, 2005) and motion of points from a pair of feet are alternating in a cyclic manner. We aim to directly learn to detect this type of foot motion in a discriminative manner. The key idea for our bipedal motion detection from trajectories is thus to detect *correlated spatio-temporal features*. That is, we detect pedestrian motion by observing the correlation between the motion of two feet of the same person. It is in contrast to (Brostow and Cipolla, 2006) whose premise is that a pair of points that appear to move together are part of the same individual. The motivation behind our strategy is the fact that bipedal motion is essential to any walking person in terms of physical dynamics of walking motion whereas the motion of other body parts such as the arms typically exhibits significantly more variation. To the best of our knowledge this is the first attempt[1] to recognize motion by way of investigating correlation of point trajectories.

In this work we opt for a learning based approach and develop a classifier for bipedal motion of a pair of feet among a number of point trajectories. We employ a decision forest classifier (Breiman, 2001; Geurts et al., 2006; Ho, 1998) which has been successfully applied to different classification tasks (Brostow et al., 2008; Lepetit et al., 2005; Rogez et al., 2008; Shotton et al.,

[1]An early description of this work has appeared in (Perbet et al., 2009).

2008). The reason of using it is in the ease of training, the ability to deal with a large amount of data, and good generalisation performance. It is also well suited to our probabilistic input. That is, we use sampled subgraphs of probabilistic trajectories as input data. We build a two-stage decision forest classifier. The first stage identifies candidate foot trajectories and the second stage associates candidate trajectories as pairs, effectively exploiting the correlated spatio-temporal features.

1.1. Related Work

A lot of work in the area of motion-based recognition including human gait analysis have been inspired by the biological phenomenon (Cédras and Shah, 1995; Gavrila, 1999). These methods typically require a robust method for feature extraction. Thus, trajectories of interest points were either obtained by using markers to simplify image analysis (Campbell and Bobick, 1995) or acquired from motion captured data (Meng et al., 2006) in place of MLDs; automatic acquisition of accurate point tracks is difficult in many cases due to effects such as occlusions, lighting changes and image noise (BenAbdelkader et al., 2002). Although relatively little work on recognition were performed purely from low-level features extracted from natural image sequences (Polana and Nelson, 1994), increasing challenges in computing and applying point trajectories have been recently presented (Sand and Teller, 2008; Perbet et al., 2009; Messing et al., 2009; Matikainen et al., 2009; Sun et al., 2010; Sundaram et al., 2010; Wu et al., 2011; Wang et al., 2011) especially in the context of action recognition.

Two of the common critical factors for computing reliable trajectories are, however, to select good repeatable features that are relevant to motion recognition, and to maintain them continuous throughout a required part of the given image sequence. In this respect, Kanade-Lucas-Tomasi (KLT) tracker (Shi and Tomasi, 1994) is a popular option and utilized in (Messing et al., 2009; Matikainen et al., 2009; Sun et al., 2010) although obtained trajectories inevitably suffer from discontinuous to an extent according to the noise and clutters. One solution to deal with discontinuities could be to generate shorter but reliable 'tracklets' (Ge and Collins, 2008) and to link them in an additional step (Huang et al., 2008). Other extentions have been for computing trajectories in a dense manner, typically with incorporation of optical flow; particle video (Sand and Teller, 2008) is one of the early such representations. For the same goal, the work in (Sun et al., 2010) combines KLT with SIFT-trajectories (Sun et al., 2009), and particle trajectories (Wu

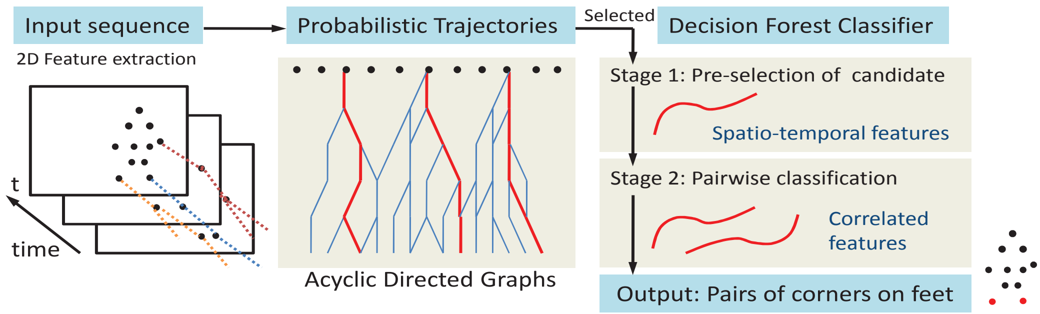


Figure 2: **Schematic of the algorithm.** *Given a video sequence and 2D corners detected in each frame, we first sample probabilistic trajectories of corners in the graph, and then classify the trajectories by a two-stage decision forest. We design correlated spatio-temporal features for classification.*

et al., 2010) have been used in (Wu et al., 2011). Also, a real-time dense point tracker (Sundaram et al., 2010) has been developed based on on large displacement optical flow (Brox and Malik, 2011). As mentioned earlier, nevertheless, there are intrinsic difficulties in maintaining correct and consistent point correspondence in generating long trajectories, most notably due to occlusions. In order to have a continuum of their 2D coordinate in the form of a trajectory, it will be indispensable to somehow reinforce correspondence. This is especially important when the correlation need be analysed between trajectories of points which can often occlude with each other. The probabilistic trajectories introduced in this paper are designed by prioritising the concept of temporal connectedness, and to utilize what we perceive from 2D motion of points in natural images for recognition.

1.2. Contributions and Assumptions

Figure 2 shows a schematic of our algorithm. The contributions of the paper are four-fold: (i) the introduction of *probabilistic trajectories* which temporally associate each point over a sufficiently long time period under both image noise and occlusion, (ii) the pairwise analysis of trajectories for detecting characteristic correlation between the two feet in bipedal motion, (iii) the design of efficient features which are computed in the two-staged decision forest classifier, and (iv) a discussion on potential extensions to deal with camera motion.

We do not assume clean point tracks but instead assume that the camera captures dynamics of motion at sufficiently high rate (we use 60 fps) and that people walk with approximately constant speed and direction during a gait cycle.

2. Probabilistic Trajectories

In this section, we describe the three successive steps that generate *probabilistic trajectory*. The basic idea is to hypothesize trajectories by enforcing temporal correspondences between consecutive frames since repeated detection of the same point over a long time interval is not always possible. In practice, we generate an acyclic graph for each point of interest using it as the root node and grow a graph such that each edge represents a possible temporal correspondence. Corners of two consecutive frames are connected probabilistically using their spatial distance and their appearance. Over T frames, those connections form a graph of possible trajectories. A walk in this graph describes a possible trajectory of a given corner over time, also including many incorrect trajectories. Our assumption is that most of these will still be discriminative, see Figure 1, right. Different paths from the root node toward the leaf nodes give different trajectories, allowing for an inherent ambiguity in the matching process. For example, although some trajectories may reflect apparent motion rather than real motion, this is encoded in a probabilistic manner. This matching process ensures that at each time step trajectories of equal length are available for all points in the frame.

2.1. Matching Between Two Consecutive Frames

In every frame we extract Harris corners (Harris and Stephens, 1988) and find potential ancestors for each point among the feature set from the previous frame. Let $p_i(t), i = 1, ..., n$ be the i^{th} corner detected at a 2D location $\mathbf{x}_i(t) \in \mathbf{R}^2$ at time t and let $p_j(t-1), j = 1, ..., m$ be the j^{th} corner found at $\mathbf{x}_j(t-1)$ among m corners which were within a certain range from $\mathbf{x}_i(t)$ in frame $t-1$. We then define the temporal matching score, $P_{ij}(t)$, that $p_i(t)$ matches $p_j(t-1)$ in terms of their appearance similarity S_{ij}, and the spatial distance D_{ij}, by

$$P_{ij}(p_i(t), p_j(t-1)) \propto \exp(-\alpha S_{ij}) \ \exp(-\beta D_{ij}) \ , \tag{1}$$

where α and β are positive weighting coefficients.

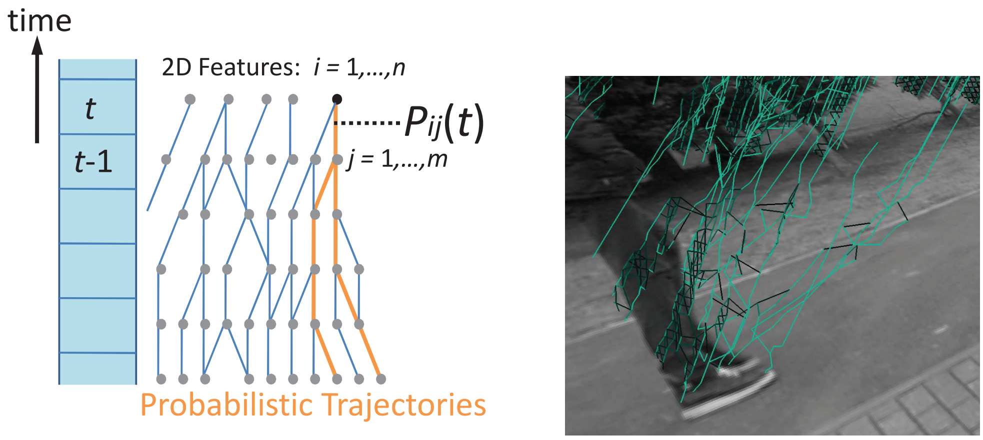


Figure 3: **Probabilistic Trajectories.** Left: *A sketch of a graph and selected probabilistic trajectories as our motion descriptor.* Right: An example of partial graph with varying color representing different probabilities, i.e. brighter indicates higher values.

The appearance similarity S_{ij} is computed from the local image regions around $p_i(t)$ and $p_j(t-1)$, respectively, as the SAD score between them (after subtracting the mean intensity of each image patch) and D_{ij} by their spatial distance $D_{ij} = \|\mathbf{x}_i(t) - \mathbf{x}_j(t-1)\|$. The assumption that the camera captures dynamics of motion at sufficiently high rate, 60 fps rather than usual 30 fps, is to ensure that for each corner point detected in frame t we can find the corresponding corner in frame $t-1$ within a reasonable range so that relatively smooth probabilistic trajectories can be generated.

We represent the existence of a potential match between $p_i(t)$ and $p_j(t-1)$ as a binary value, $E_{ij}(t) \in \{0, 1\}$, based on $P_{ij}(t)$ and define the match as active, $E_{ij}(t) = 1$, with the condition:

$$P_{ij} > \max_j P_{ij} - e \ , \tag{2}$$

where the threshold value e is dynamically adjusted so that the number of pairs is constant which is set to $4n$. Note that this may result in no active matches for some corners with low values of $\max_j P_{ij}$. We also add temporal matches for the same set of consecutive frames in the forward direction by repeating the process in a reverse manner so that more potential matches are ensured.

2.2. Acyclic Graph with Matching Probabilities

For each time step t we have determined temporal matches $E_{ij}(t)$ between corners across previous adjacent frames. We retain these for the last T frames (the choice of T will be discussed later). Defining each point $p_i(t)$ as a root node, we generate an acyclic graph, $\mathcal{G}_i(N, E)$, of depth T by tracing active temporal matches along the time axis backward for T frames, see Figure 3. The graph $\mathcal{G}_i(N, E)$ consists of nodes, N, which represent matched corners in the preceding T frames, and edges, E, connecting these nodes. Namely, $E = (E_{ij}(\tau), \tau = t, ..., t - T + 1)$. An edge representing an active match, $E_{ij}(t)$, has $P_{ij}(t)$ as its associated weight.

Note that the number of frames, d, for which each corner can be traced back (until encountering an inactive edge or a 'dead end') is available for each node. For example, $d[E_{ij}(t)] = 1$ if $p_j(t-1)$ has no ancestor and $E_{jk}(t-1) = 0$ for all k (where k is an index to features in frame $t-2$). We assign d to each node as its attribute while ideally $d > T$ where at least one path can be found containing nodes from the T previous frames.

2.3. Sampling Probabilistic Trajectories

In each time step the graph is updated and trajectories are sampled from it that are then classified. Intuitively, the sampled trajectories need to be long and physically plausible. Now, we define the *probabilistic trajectories*, $X_i(t) \in \mathbf{R}^{2T}$ of $p_i(t)$, as the paths connecting the root node to different leaf nodes of $\mathcal{G}_i(N, E)$. In practice, a graph traversal of $\mathcal{G}_i$ guided by a probabilistic selection of edges at each node results in plausible trajectories. In particular, we use the sampling probability, $\widehat{P}_{ij}$, in which we also take into consideration the traceable depth d and the velocity conservation factor, V_{ij}:

$$\widehat{P}_{ij}(p_i(t), p_j(t-1)) \propto P_{ij} \exp\left(-\frac{\gamma}{d[E_{ij}]+1}\right) \exp(-\delta V_{ij}) \ , \tag{3}$$

where γ and δ are positive weighting coefficients, and the last factor

$$V_{ij}(\tau) = \|(\mathbf{x}_h(\tau+1) - \mathbf{x}_i(\tau)) - (\mathbf{x}_i(\tau) - \mathbf{x}_j(\tau-1))\| \tag{4}$$

is valid when $\tau < t$ (so that the coordinate of the previous node in the path, $\mathbf{x}_h(\tau+1)$, is available). We set $\gamma = 10$ and $\delta = 1$ in our experiments.

A sophisticated selection of paths such as in (Torresani et al., 2008) would help if spatial coherence of matched points could be taken into account. In our case, however, points on different feet have diverse spatial path and we remain to find the path individually.

2.4. Computational complexity

We detect n corner points in each frame and use m corners in the previous frame for computing the correlation; the complexity of matching between two consecutive frames is $O(nm)$. However, m is set to be significantly smaller than n; in our experiments we set $n = 300$ and $m = 10$. Also, the total number of pairs that are considered to be parts of trajectories is adjusted to $4n$. The complexity for the computation between consecutive frame can be seen as $O(n)$ given $n >> m$. For generating an acyclic graph and thereby sampling trajectories, we need to compute the velocity conservation factor as well as the traceable depth for the length of the trajectories T; the complexity in this respect is $O(T)$, making the total computational complexity for computing the probabilistic trajectories $O(nT)$.

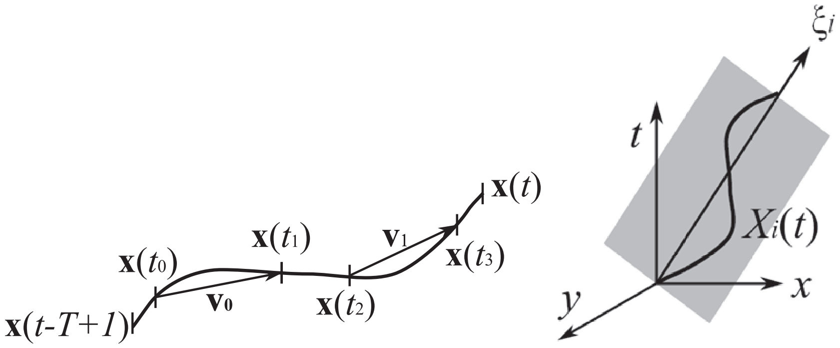
3. Classification of Trajectories

Given a corner, $p_i(t)$, and its probabilistic trajectory, $X_i(t)$, our task is now to determine whether or not $X_i(t)$ is the trajectory of a foot during walking motion. In order for a trajectory to contain discriminative features, we consider its length T as approximately covering one walk cycle. As mentioned above, the key idea is to observe point trajectories in pairs. That is, we also consider $p_u(t)(u \neq i)$ that are located in the neighborhood of $p_i(t)$ and examine the spatio-temporal correlation between the probabilistic trajectories, $X_i(t)$ and $X_u(t)$.

In order to avoid examining the large number of possible pairs we also use the fact that some trajectories can be rejected immediately as candidates, such as those from stationary background points or those that are too noisy due to incorrect temporal association. Thus, we employ a two-stage classification process:

1° Selection of candidate trajectories.
2° Pairwise classification of pertinent trajectories.

It should be noted that we benefit from the selection of the candidates in reducing the complexity in terms of the number of possible pairs to be considered in the second stage, and therefore overall computational cost. In each stage, we need a classification tool and invariant features that allow us to distinguish trajectories of walking motion from others. We perform classification using decision forests in both stages. Decision forests are well



(a) (b)

Figure 4: **Feature Vectors from Trajectories.** (a) In order to compute features we sample many pairs of velocity vectors from a trajectory. (b) The principal direction ξ of the trajectory $X_i(t)$ is used for the directional feature computation.

suited to our task because each of our input instances is probabilistic, i.e. any trajectories sampled as subgraphs are probabilistic in both stages, 1° and 2°.

3.1. Selection of Candidate Trajectories

The goal in the first stage is to select candidate trajectories prior to pairwise classification. We thus carry out the selection of candidate trajectories by individual $X_i(t)$. The feature design for this stage is based on the observation that $X_i(t)$ originating from a foot is characterized by dynamic and static phases, being distinguishable from simple trajectories coming from background.

3.1.1. Feature Vectors

Let a trajectory, $X_i(t)$, be represented by a vector $X_i(t) = [\mathbf{x}(t), \mathbf{x}(t-1), ..., \mathbf{x}(t-T+1)]^\top$. We first remove its linear component, $\bar{X}_i(t)$, and convert $X_i(t)$ to its canonical form, $\tilde{X}_i(t) = [\tilde{\mathbf{x}}(t), \tilde{\mathbf{x}}(t-1), ..., \tilde{\mathbf{x}}(t-T+1)]^\top$ (see Appendix). The merit of using the canonical form, $\tilde{X}_i(t)$, is that it represents the motion characteristics independent of its location.

We generate two feature vectors from $\tilde{X}_i(t)$, $\mathbf{v}_0$ and $\mathbf{v}_1$ as the velocity term. By randomly choosing four time instances as cutting points, $t_c(c = 0, ..., 3; t_c < t_{c+1})$, we extract

$$\mathbf{v}_0 \quad = \quad \bar{\mathbf{x}}(t_1) - \bar{\mathbf{x}}(t_0), \tag{5}$$

$$\mathbf{v}_1 = \bar{\mathbf{x}}(t_3) - \bar{\mathbf{x}}(t_2). \tag{6}$$

Namely, we sample two random velocities, $\mathbf{v}_0$ and $\mathbf{v}_1$, along a trajectory by choosing two points per velocity, see Figure 4 (a). This operation of cutting a trajectory at four points is motivated by the observation that four dynamical models per gait cycle is a reasonable choice in a probabilistic decomposition human gait (Bregler, 1997) where coherent motion is used as low-level primitives. Note that $t \geq t_c \geq t - T$. We then define our features, f_s and f_d, by the distance and the inner product of scaled versions of the two vectors:

$$f_s = \|a_0\mathbf{v}_0 - a_1\mathbf{v}_1\|, \tag{7}$$
$$f_d = \langle b_0\mathbf{v}_0, b_1\mathbf{v}_1\rangle, \tag{8}$$

where a_i and $b_i, i = 0, 1$ are random coefficients in $(0, 1)$. Different features f_s and f_d are generated by sampling values for the coefficients a_i and b_i, as well as the cutting points, $t_c(c = 0, ..., 3)$, of the trajectory.

3.1.2. Learning Using Random Samples

We obtain training data by manually annotating points corresponding to foot regions in a video, and then extracting probabilistic trajectories, $X_i(t)$ of length T, as random subgraphs which stem from the annotated corners. Extracting trajectories for each corner in each frame, we obtain a large number of training data. Training is performed separately for each tree using a random subset of the training data.

We recursively split the training data at each node, using the standard method involving information gain (Breiman, 2001). Namely, we plot the responses of randomly selected features in a histogram and learn a threshold, θ, which gives the maximum information gain. A node becomes a leaf node where the information gain is below a threshold value. At each leaf node, the class distribution of foot/non-foot is computed from the number of instances that reach the node. See Figure 5 for an example of our decision forest.

We annotate the ground truth data of feet with a tag of left/right foot so that they can be directly used for training the decision forest in the second stage. It should be noted that those points that can be associated with both feet are also annotated with equal probabilities of being on left/right foot. See Figure 6 for an example of ground truth labels.

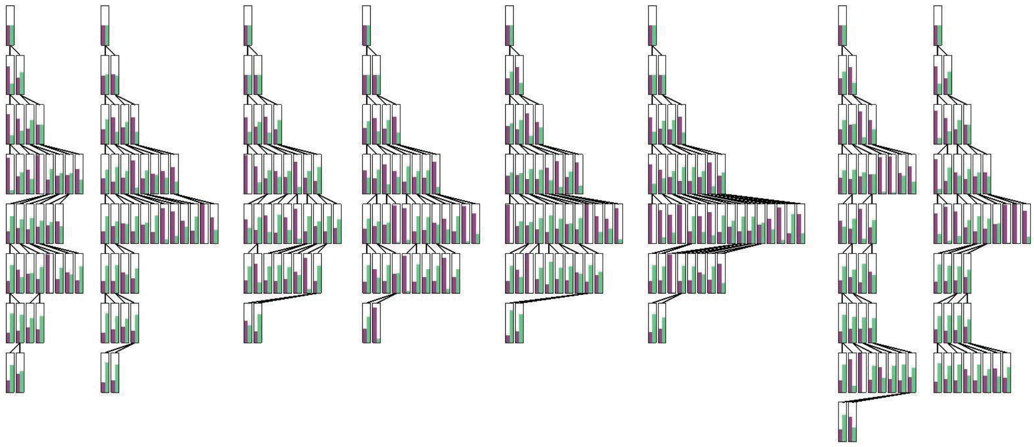


Figure 5: **Example overview of the decision forest** (of the secocnd stage, $F = 8$). The learned class distributions of foot/non-foot are displayed for each node as a histogram.

3.1.3. Selection of Trajectories

We classify candidate trajectories with a decision forest (Breiman, 2001; Geurts et al., 2006) which is an ensemble of F decision trees. Each tree examines all input trajectories, $\tilde{X}_i(t)$. Given an input trajectory at the root node, each decision tree recursively branches left or right down to the leaf nodes according to the feature response, f_s and f_d in (8), of a learned function at each non-leaf node. At the leaf nodes, we obtain the class distributions of foot/non-foot. The output from F decision trees is averaged to select candidate trajectories.

3.2. Pairwise Classification of Walking Motion

Given that a corner $p_i(t)$ is selected as a candidate point in the first stage, we pick those $p_u(t)(u \neq i)$ which are located in the neighborhood of $p_i(t)$ and examine how their probabilistic trajectories, $\tilde{X}_i(t)$ and $\tilde{X}_u(t)$, are spatio-temporally correlated.

3.2.1. Features for Directional Correlation

Although the trajectory, $X_i(t)$, is three-dimensional, when walking in a straight line, the trajectory lies approximately in a 2D plane. If a set of two candidate trajectories, $X_i(t)$ and $X_u(t)$, arises from walking motion of two feet, the orientations of their 2D planes in 3D space should be close to each other because of the consistency of a pair of step motions (Hoffman and Flinchbaugh, 1982). Based on this observation, we compute the covariance matrices, C_i, of $\tilde{\mathbf{x}}(\tau), \tau = t, ..., t-T+1$, and the eigenvector, $\xi_i \in \mathbf{R}^2$, corresponding to the greatest eigenvalue so that ξ_i represents the principal direction of $\tilde{X}_i(t)$ along its 2D plane, see Figure 4 (b). Analogously ξ_u is computed for $\tilde{X}_u(t)$.

We expect ξ_i and ξ_u to be approximately parallel, their directions should be both close to the walking direction. For most of the gait cycle the vector connecting the two front points on the trajectory $\mathbf{x}_{iu}(t) = \mathbf{x}_i(t) - \mathbf{x}_u(t)$ can be used as an approximation for this direction. We compute a feature vector containing inner products, $\mathbf{c} \in \mathbf{R}^3$,

$$\mathbf{c} = \begin{bmatrix} \|\langle \xi_i, \xi_u \rangle\| \\ \|\langle \xi_i, \mathbf{x}_{iu}(t) \rangle\| \\ \|\langle \xi_u, \mathbf{x}_{iu}(t) \rangle\| \end{bmatrix} \tag{9}$$

and a random vector $\phi \in \mathbf{R}^3$, $\|\phi\| = 1$, so that

$$f_o = \langle \phi, \mathbf{c} \rangle. \tag{10}$$

3.2.2. Features for Walking Phase Correlation

Importantly, we design a feature based on the fact that trajectories from a pair of feet are out of phase with each other, alternating in a cyclic manner with dynamic and static phases. This means that one foot is mainly in the dynamic phase while the other is in the static phase. Since one has nearly zero velocity during most of the cycle, we can expect the dot product of their velocity vectors, after proper rectification, to be also close to zero. For this purpose we consider the trajectory, $X_i(t)$, in terms of velocity by generating a vector

$$Y_i(t) = [\mathbf{y}(t), \mathbf{y}(t-1), ..., \mathbf{y}(t-T+2)]^\top \in \mathbf{R}^{2(T-1)} \tag{11}$$

where $\mathbf{y}(\tau) = \mathbf{x}(\tau) - \mathbf{x}(\tau-1), \tau = t, ..., t-T+2$. We convert each $\mathbf{y}(\tau)$ to $\check{\mathbf{y}}(\tau)$ by projecting it to the axis of ξ_i. Thus, the rectified velocity vector is

$$\check{Y}_i(t) = [\check{\mathbf{y}}(t), \check{\mathbf{y}}(t-1), ..., \check{\mathbf{y}}(t-T+2)]^\top. \tag{12}$$

Rather than simply taking the inner product of the entire $\check{Y}_i(t)$ and $\check{Y}_u(t)$, which would result in a scalar, we compute their piecewise dot products. By

Figure 6: **Ground truth labels:** *Corners detected inside the circles are annotated as being on a foot. We use an in house annotation tool which accelerates the process by allowing probabilistic labelling.*

cutting each of $\breve{Y}_i(t)$ and $\breve{Y}_u(t)$ into l pieces at common fixed cutting points, $t_c(c = 0, ..., l-2; t_c > t_{c+1})$, we acquire a vector

$$\mathbf{q} = [\langle \breve{Y}_i'(t), \breve{Y}_u'(t) \rangle, ..., \langle \breve{Y}_i'(t_{l-2}), \breve{Y}_u'(t_{l-2}) \rangle]^\top \in \mathbf{R}^l, \tag{13}$$

where $\breve{Y}_i'(t_c)$ represents a portion of $\breve{Y}_i(t)$ starting at t_c. We then define a phase feature f_p as the inner product of $\mathbf{q}$ with a random vector, $\psi \in \mathbf{R}^l$ where $\|\psi\| = 1$:

$$f_p \quad = \quad \langle \psi, \mathbf{q} \rangle. \tag{14}$$

We choose to use $l = 5$, again assuming that the four dynamical models per gait are well covered in the trajectories.

3.2.3. Final Detector Output

The output from the second decision forest consists of a set of hundreds of feature pairs along with their probabilities of being a pair of feet (see bottom-right in Figure 7 for an example). In order to extract a major pair from this set, we run mean-shift clustering and take the average of the most probable cluster as the final estimate.

Note that the cost for the algorithm including the classification and this clustering step could vary depending on the number of candidate trajectories that are selected in the first stage.

Figure 7: **Results on Sequence I** *Detection results superimposed on the input frames (from left to right, 150 frames between each view). Top: Extracted corner points and candidate points after the first stage are shown in green, the rejected points in purple, and points with trajectories of insufficient length in black. Middle: Pairs of corners extracted as feet in the second stage shown as green line segments. Bottom: Final detection after running mean-shift. Right: Example of all possible pairs between pre-selected corners at stage* 1. *Purple line segments are those rejected in stage* 2.

4. Experiments

In order to obtain training data we made manual annotations in real training images. Figure 6 shows how we acquire them in a sequence. The two circles indicate areas where corners detected inside are annotated as being on foot in the current frame. Corners on the right foot and the left foot are annotated separately by brown and yellow circles, respectively. The smaller circles indicate the annotated locations of the feet in other frames of the image sequence. As inputs, we captured video sequences (resolution 1280×720 pixels at 60 fps) in which a person walks in seven different directions as well as other sequences including different persons walking at different speed.

Figure 7 illustrates the performance of the proposed classification on a sequence of 350 frames. The selected candidates of the first stage and the classified pairs in the second stage are shown in green, in the top and the middle rows, respectively. Although there are some connections between corners for example on arms as their motion is similar to feet (see the bottom-right picture in a larger scale), the connections between feet are generally dominant, and the final detection results are shown in red in the bottom row. The algorithm currently runs at 2-5 frames per second.

Figure 8 shows detection in a 400-frame sequence of two pedestrians cross-

Figure 8: **Results on Sequence II.** *Detection of bipedal motion of two people walking across the scene from opposite directions (from top to bottom). Left: Candidate points after the first stage are shown in green, rejected points in red, points with short trajectories in black. Middle: Candidate pairs after the second stage shown as green line segments. Right: Final detection result of pairs of feet after running mean-shift shown as red line segments.*

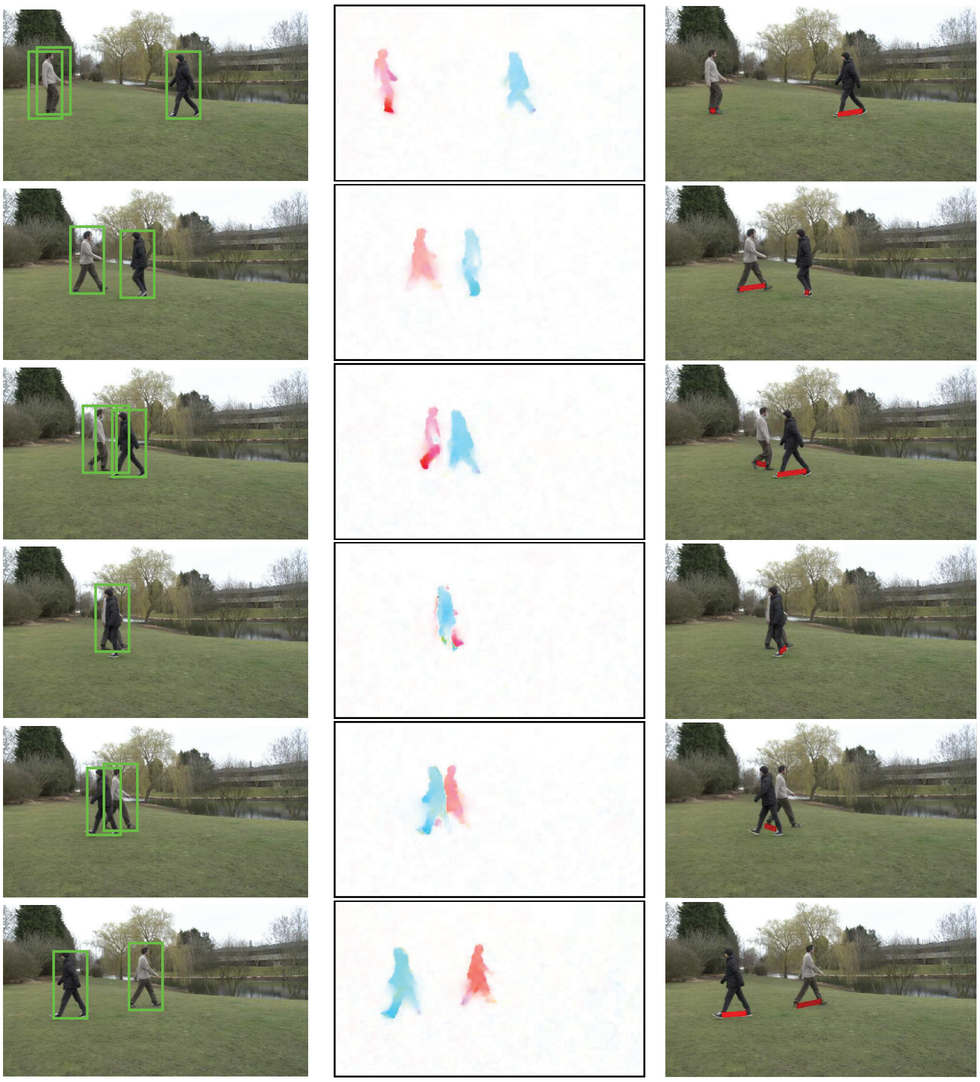


Figure 9: **Pedestrian detection on Sequence II by motion-based technique.** *Detection of two pedestrians walking across the scene from opposite directions by a sliding window search with histograms of flow (HOF) and HOG descriptors. See texts for details. Left: Final detection result shown with green bounding boxes. Middle: Color coded optic flow used for computing histograms of flow. Right: Pairs of feet detected by the proposed method: the same results that are in Figure 8 (for side-by-side comparison).*

Table 1: **The detection rates** (%). *Top: The detection rates of pairs of feet for the sequence including two pedestrians. See text for the definition of* r_{simple} *and* r_{pure}*. Middle: The detection rates of pedestrians by a sliding window search with the HOF and HOG descriptors (Walk et al., 2010). Bottom: The detection rates of pairs of feet using the annotations as the ground truth. (The threshold distance is set to 30 pixels.)*

Proposed	r_{simple}	r_{pure}
Person A (front)	90.6	83.4
Person B (back)	78.4	69.1

HOF+HOG	r_{simple}	r_{pure}
Person A (front)	92.4	83.6
Person B (back)	83.4	67.2

Proposed	Example 1	Example 2
Single person	68.9	66.0

ing the scene. Candidate of a set of feature (trajectory) pairs as the output from the second decision forest are shown as green line segments in the second column. Note that we can observe two obvious modes corresponding to two pedestrians for which we run mean-shift clustering and take the average of the most probable cluster(s). This example shows that we can select more than one major mode at this stage to determine the final estimate when dealing with multiple targets.

Table 1 (top) further shows the detection rates for each of the two pedestrians; r_{simple} is based on a simple count of successful cases where a mode connecting the two feet is detected whereas r_{pure} indicates a similar rate but excluding the cases when an extra mode is also detected by mistake for instance between an arm and the background. The overall detection rates are lower for Person B that is walking in behind and occluded by Person A in part of the sequences. During this crossing phase only one pair of feet is detected, but subsequently both are detected again correctly. The supplementary video demonstrates the performance of detections.

For a comparison, we have also applied a state-of-the-art motion-based pedestrian detector with histograms of flow (HOF) descriptor[2] (Dalal et al.,

[2]The exact descriptor we implemented is called IMHd2 (Walk et al. CVPR10) which is

2006; Walk et al., 2010) to Sequence II in a framework of sliding window search; Figure 9 shows some examples of the results with green bounding boxes as well as the color coded optic flow (Werlberger et al., 2009) which was the basis for computing HOF features. The results demonstrate that the motion-based pedestrian detection also performs well generally, missing the target only during the crossing phase. By the nature of sliding window search, however, multiple detections could appear for each target (e.g. in the top row) although a non-maximum suppression has been performed. Also, a spurious detection is observed when two targets are close to each other (see the third row). Note that in the same frame their feet are detected properly by the proposed approach. Nevertheless, when two people are half overlapping (the fifth row), the clustering process after the feet detection is confused (see also the corresponding row of Figure 8 for the raw detections) while the pedestrian detection is performing stably.

Another difference that should be addressed is the location of detection; some ambiguity is inherent in detections with a sliding window search, which is not considered as a problem when evaluated by the intersection-over-union measure with annotations, whereas the proposed method directly indicates quite precise positions of feet in the given image. We will discuss the importance of this issue in terms of an application in Section 6.

Table 1 (middle) shows the detection rates for each of the two pedestrians computed for Sequence II by the sliding window search with HOF and HOG descriptors (Walk et al., 2010). Analogously to the case with our feet detection, r_{simple} refers to a simple count of successful pedestrian detections in terms of intersection-over-union measure whereas r_{pure} indicates a similar rate but excluding the cases when wrong detection(s) also occurred by mistake. As was the case with feet detection, the overall detection rates are lower for Person B than Person A mainly due to the crossing phase. The performance is somewhat comparable to the proposed feet detection although it is not possible to deduce superiority of one against the other; the IMHd2 is computed for the entire pedestrian regions at a time but uses optic flow just based on two frames whereas the feet detection uses only local information as few as two feature points but for 60 frames. If we make a simple com-

combined with HOG features. We used the TUD-MotionPairs dataset (Wojek et al., 2009) for training the model as suggested in (Walk et al., 2010), and the histogram intersection kernel SVM (Maji et al., 2008) for the classifier.

Figure 10: **Results on Sequence III.** *Feet are correctly detected in the first frames (left, middle), but in the last frame (right) a detection in the arm region occurs due to very similar motion. Top: Candidate points after the first stage are shown in green, rejected points in purple, points with short trajectories in black. Middle: Candidate pairs after the second stage are green line segments. Bottom: Final detection after running mean-shift.*

parison for Person A, r_{simple} with IMHd2 is a little higher than that with the proposed feet detection (see Table 1 (top)) while r_{pure} being at the same level, implying an equivalent false positive ratio. r_{simple} with IMHd2 for Person B is also higher than that with the feet detection, but the score of r_{pure} drops significantly to the level that is lower than the proposed feet detection, reflecting more cases of spurious detections involved in the window search.

In order to evaluate the detection rate in a more strict sense, for an outdoor sequence with one pedestrian of 595 frames, we compute the error as distance between the detected pairs and the annotated pairs. For this the correspondences between the two pairs is found and the error defined as the average distance between corresponding points. The average error is less than a threshold distance[3] from the ground truth for 410 frames. For another sequence, the distance was below the same threshold for 310 frames out of 470 input frames. Approximately half of the error cases were due to incorrect detection of arm motion. See Table 1 (bottom) for the summary of the detection rates considering the distances to the annotations.

5. Discussion

5.1. Failure Case

Figure 10 shows an example sequence of 500 frames where outliers become more dominant and the final detection is no longer at the foot location. Although pairs are detected on the feet (left), a number of pairs remain candidates as the result of classification stages. Outliers include pairs of points on the arms as well as pairs between the body and the background. Through a detailed analysis typical cases have been found where a trajectory is first on body but taken over the background, or vice versa. Those pairs tend to cluster and take over the final signal as being from feet after a few seconds. However, one possible approach to avoid those cases will be to perform further careful training by using such instances as negative examples.

5.2. Occlusions and Crowd

One of the benefits of our new representation is that it tolerates inherent ambiguity due to occlusion which occurs internally to a target. However, external occluding object/target would be a cause of error, just as in many

[3]We set it to be 30 pixels in this evaluation.

existing algorithms, as shown in Figure 8 for the case with two people; depending on the phase of the overlap the output of the final clustering process could be a pair of feet of the frontal person, or a pair of feet of two nearby people whose motion happens to be correlated in the similar way as that of a real pair of feet does. The stability against more precise classification is therefore in working progress.

The current approach will be confused with further crowded people, being not suitable to such a situation. For dealing with complicated crowded sequence, other representations such as particle trajectories (Wu et al., 2010) have been proposed which are designed for anomaly detection while producing some representative trajectories of a crowd. On the other hand, our approach will help when the position of feet in given images need be detected explicitly.

5.3. Moving Camera

Moving cameras generally pose significant challenges for recognising motion due to the changes in the field of view. Although we have assumed a static camera to capture input sequences, future work will be directed to the case of a moving camera. In order to tackle this problem, it will be necessary to separate the global camera motion and the local object motion somehow in the acquired frames. Figure 11 shows trajectories for the cases with both stationary and moving camera. The space-time volume is displayed so that the time-axis is along the vertical direction. In the case of a stationary camera, trajectories connecting background corners are vertically aligned. On the other hand, trajectories viewed by a moving camera exhibit more variation. However, for sufficiently smooth camera motion the trajectories of points on the feet are still recognizable, and therefore we believe that it is possible to eliminate the dominant camera motion.

One way to deal with such variations is to generate a rectified velocity vector in (12) so as to cancel the possible camera motion; given a velocity vector, $Y_i(t)$, as in (11) we can compute the rectification by

$$\breve{\mathbf{y}}(\tau) = \hat{\mathbf{y}}(\tau) - \min_{\tau} \hat{\mathbf{y}}(\tau) \tag{15}$$

where $\hat{\mathbf{y}}(\tau)$ is obtained by taking the absolute value of each element of resulting vector after the projection.

Other possibilties are to employ a global motion compensation step similar to (Mikolajczyk and Uemura, 2008) or the motion features computed

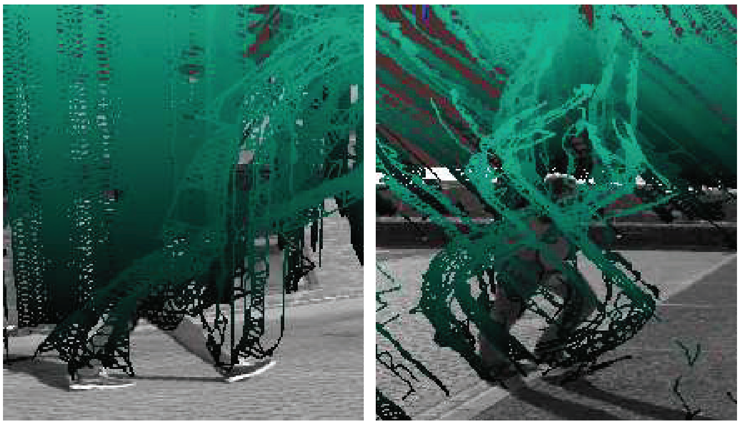


Figure 11: **Static and moving camera.** *Point trajectories in the case of a stationary camera (left) and a moving camera (right).*

from 3D trajectories introduced in (Brostow et al., 2008). Also, a promising approach to handle the moving camera is to decompose the trajectories into their camera-induced and object-induced componens based on low rank optimization as recently suggested in (Wu et al., 2011).

6. Conclusion

We have introduced a new algorithm for detecting bipedal motion from point trajectories. In particular, we proposed to use the fact that trajectories from two feet are spatio-temporally correlated. To this end, we have introduced (i) the notion of *probabilistic trajectories*, (ii) the pairwise analysis of trajectories for detecting their correlation, (iii) the design of efficient features for a two-stage decision forest classifier, and (iv) potential extensions to deal with camera motion. To the best of our knowledge this is the first attempt to recognize walking motion by way of investigating correlation of point trajectories.

The strategy in this paper was to retain the uncertainty in the track associations and let the classifier handle this uncertainty. However, other advanced methods for association could result in less ambiguous trajectories and thus allow the features to be more discriminative. Although our method currently assumes little variation in walking speed, it will be also useful in the future work to model the period of a walk cycle (Cutler and Davis, 2000;

Laptev et al., 2005) and to identify each phase of walking motion.

The original goal of the work is to replicate the ability to recognize motion by the effect of *kinetic depth* using tracked points on a pair of feet. In terms of applications, we have designed the method to be used as a module of pedestrian detection system, where feet detection helps to measure the distance to the target since the 2D location in the image can be directly mapped to 3D distance given a calibrated camera. This function would be especially useful for an automotive system to measure the time-to-contact in the area of monocular surveillance given the driving speed (Enzweiler et al., 2008). State-of-the-art methods for pedestrian detection based on a window search return a bounding box as the detected result, but the bottom end of the bounding box does not necessarily coincide with the precise 2D location of feet. Thus, we point out that the proposed motion-based technique may well complement appearance-based methods such as in (Dalal and Triggs, 2005) in the context of pedestrian detection.

Appendix A. The Canonical Form of Trajectories

The canonical form, $\tilde{X}_i(t)$, of a trajectory $X_i(t)$, is computed as

$$\tilde{X}_i(t) = X_i(t) - \bar{X}_i(t) \tag{A.1}$$

where $\bar{X}_i(t) = [\bar{\mathbf{x}}(t), ..., \bar{\mathbf{x}}(t-T+1)]^\top$ and

$$\bar{\mathbf{x}}(\tau) = \frac{1}{T-1}[(t-\tau)\,\mathbf{x}(t-T+1) + (\tau-t+T-1)\,\mathbf{x}(t)]. \tag{A.2}$$

References

BenAbdelkader, C., Davis, L. S., Cutler, R., 2002. Motion-based recognition of people in eigengait space. In: Proc. Int. Workshop on Automatic Face- and Gesture-Recognition. pp. 267–274.

Bissacco, A., 2005. Modeling and learning contact dynamics in human motion. In: IEEE CVPR (1). pp. 421–428.

Bregler, C., 1997. Learning and recognizing human dynamics in video sequences. In: IEEE CVPR. pp. 568–575.

Breiman, L., 2001. Random forests. Machine Learning 45 (1), 5–32.

Brostow, G. J., Cipolla, R., 2006. Unsupervised bayesian detection of independent motion in crowds. In: IEEE CVPR (1). pp. 594–601.

Brostow, G. J., Shotton, J., Fauqueur, J., Cipolla, R., 2008. Segmentation and recognition using structure from motion point clouds. In: ECCV. Vol. I. pp. 44–57.

Brox, T., Malik, J., 2011. Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (3), 500–513.

Brubaker, M. A., Fleet, D. J., Hertzmann, A., 2010. Physics-based person tracking using the anthropomorphic walker. International Journal of Computer Vision 87 (1-2), 140–155.

Campbell, L., Bobick, A., 1995. Recognition of human body motion using phase space constraints. In: IEEE ICCV. pp. 624–630.

Cédras, C., Shah, M. A., 1995. Motion based recognition: A survey. Image and Vision Computing 13 (2), 129–155.

Cutler, R., Davis, L. S., 2000. Robust real-time periodic motion detection, analysis, and applications. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 781–796.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: IEEE CVPR (1). pp. 886–893.

Dalal, N., Triggs, B., Schmid, C., 2006. Human detection using oriented histograms of flow and appearance. In: ECCV (2). pp. 428–441.

Enzweiler, M., Kanter, P., Gavrila, D., 2008. Monocular pedestrian recognition using motion parallax. In: Intelligent Vehicles Symposium. pp. 792–797.

Gavrila, D., 1999. The visual analysis of human movement: A survey. Computer Vision and Image Understanding 73 (1), 82–98.

Ge, W., Collins, R. T., 2008. Multi-target data association by tracklets with unsupervised parameter estimation. In: BMVC.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Machine Learning 36 (1), 3–42.

Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conference. pp. 147–151.

Ho, T.-K., 1998. The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20 (8), 832–844.

Hoffman, D. D., Flinchbaugh, B. E., 1982. The interpretation of biological motion. Biol. Cyb. 42:3, 195–204.

Huang, C., Wu, B., Nevatia, R., October 2008. Robust object tracking by hierarchical association of detection responses. In: ECCV. Vol. II. pp. 788–801.

Johansson, G., 1973. Visual perception of biological motion and a model for its analysis. Perception & Psychophysics 14 (2), 201–211.

Laptev, I., Belongie, S. J., Pérez, P., Wills, J., 2005. Periodic motion detection and segmentation via approximate sequence alignment. In: IEEE ICCV. pp. 816–823.

Laptev, I., Lindeberg, T., 2003. Space-time interest points. In: IEEE ICCV. pp. 432–439.

Lepetit, V., Lagger, P., Fua, P., 2005. Randomized trees for real-time keypoint recognition. In: IEEE CVPR (2). pp. 775–781.

Maji, S., Berg, A. C., Malik, J., 2008. Classification using intersection kernel support vector machines is efficient. In: CVPR. pp. 1–8.

Matikainen, P., Hebert, M., Sukthankar, R., September 2009. Trajectons: Action recognition through the motion analysis of tracked features. In: Workshop on Video-Oriented Object and Event Classification, ICCV 2009.

Meng, Q., Li, B., Holstein, H., 2006. Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach. Image Vision Comput. 24 (8), 795–809.

Messing, R., Pal, C., Kautz, H. A., 2009. Activity recognition using the velocity histories of tracked keypoints. In: IEEE ICCV. pp. 104–111.

Mikolajczyk, K., Uemura, H., June 2008. Action recognition with motion-appearance vocabulary forest. In: IEEE CVPR. Anchorage, pp. 1–8.

Perbet, F., Maki, A., Stenger, B., 2009. Correlated probabilistic trajectories for pedestrian motion detection. In: IEEE ICCV. pp. 1647–1654.

Polana, R., Nelson, A., 1994. Low level recognition of human motion (or how to get your man without finding his body parts). In: Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects. pp. 77–82.

Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., Torr, P. H. S., 2008. Randomized trees for human pose detection. In: IEEE CVPR.

Sand, P., Teller, S. J., 2008. Particle video: Long-range motion estimation using point trajectories. International Journal of Computer Vision 80 (1), 72–91.

Shi, J., Tomasi, C., 1994. Good features to track. In: IEEE CVPR. pp. 593–600.

Shotton, J., Johnson, M., Cipolla, R., 2008. Semantic texton forests for image categorization and segmentation. In: IEEE CVPR.

Sun, J., Mu, Y., Yan, S., Cheong, L. F., 2010. Activity recognition using dense long-duration trajectories. In: IEEE International Conference on Multimedia and Expo. pp. 322–327.

Sun, J., Wu, X., Yan, S., Cheong, L. F., Chua, T.-S., Li, J., 2009. Hierarchical spatio-temporal context modeling for action recognition. In: IEEE CVPR. pp. 2004–2011.

Sundaram, N., Brox, T., Keutzer, K., 2010. Dense point trajectories by gpu-accelerated large displacement optical flow. In: ECCV. pp. 438–451.

Torresani, L., Kolmogorov, V., Rother, C., 2008. Feature correspondence via graph matching: Models and global optimization. In: ECCV. Vol. II. pp. 596–609.

Walk, S., Majer, N., Schindler, K., Schiele, B., 2010. New features and insights for pedestrian detection. In: CVPR. pp. 1030–1037.

Wang, H., Kläser, A., Schmid, C., Liu, C.-L., 2011. Action recognition by dense trajectories. In: IEEE CVPR. pp. 3169–3176.

Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H., 2009. Anisotropic Huber-L1 optical flow. In: BMVC.

Wojek, C., Walk, S., Schiele, B., 2009. Multi-cue onboard pedestrian detection. In: CVPR. pp. 1–8.

Wu, S., Moore, B. E., Shah, M., 2010. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In: IEEE CVPR. pp. 2054–2060.

Wu, S., Oreifej, O., Shah, M., 2011. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In: IEEE ICCV. pp. 1419–1426.