Human Pose and Path Estimation from Aerial Video using Dynamic Classifier Selection

Asanka G Perera · Yee Wei Law · Javaan Chahl

Abstract Background / introduction— We consider the problem of estimating human pose and trajectory by an aerial robot with a monocular camera in near real time. We present a preliminary solution whose distinguishing feature is a *dynamic classifier selection* architecture.

Methods— In our solution, each video frame is corrected for perspective using projective transformation. Then, two alternative feature sets are used: (i) Histogram of Oriented Gradients (HOG) of the silhouette, (ii) Convolutional Neural Network (CNN) features of the RGB image. The features (HOG or CNN) are classified using a *dynamic classifier*. A class is defined as a pose-viewpoint pair, and a total of 64 classes are defined to represent a forward walking and turning gait sequence.

Results— Our solution provides three main advantages: (i) Classification is efficient due to dynamic selection (4-class vs. 64-class classification). (ii) Classification errors are confined to neighbors of the true viewpoints. (iii) The robust temporal relationship between poses is used to resolve the left-right ambiguities of human silhouettes.

Conclusions— Experiments conducted on both fronto-parallel videos and aerial videos confirm our solution can achieve accurate pose and trajectory estimation for both scenarios. We found using HOG features provides higher accuracy than using CNN features. For example, applying the HOG-based variant of our scheme to the "walking on a figure 8-shaped path" dataset (1652 frames) achieved estimation accuracies of 99.6% for viewpoints and 96.2% for number of poses.

Keywords Pose estimation · Gait estimation · Trajectory estimation · Dynamic classifier selection · UAV · Drone

A. G. Perera · Y. W. Law · J. Chahl
School of Engineering, University of South Australia, Mawson Lakes, SA 5095, Australia
E-mail: asanka.perera@mymail.unisa.edu.au

Y. W. Law
E-mail: yeewei.law@unisa.edu.au

J. Chahl
Joint and Operations Analysis Division, Defence Science and Technology Group, Melbourne, Victoria 3207, Australia
E-mail: javaan.chahl@unisa.edu.au

1 Introduction

The research reported in this article is motivated by the application scenario — for example, in disaster response, where an unmanned aerial vehicle (UAV) is required to recognize the actions of a human subject, then take responsive actions. The application scenario invites the following challenges: (i) Before a UAV can even begin recognizing human actions, the UAV will first have to compute how to orientate itself towards the human subject. (ii) Many UAVs are equipped with only one monocular camera; hence, additional data provided by stereoscopic, infrared and more advanced cameras is unavailable. (iii) Recognizing human actions from videos captured from a stationary platform is already a challenging task, owing to the articulated structure and range of poses of the human body. (iv) The difficulty of recognition is compounded by the quality of videos which include perspective distortion, occlusion and motion blur.

We assume the UAV is in hovering flight, having a human subject within its field of view[1]. We estimate the gait sequence and movement trajectory of a person

---

[1] https://asankagp.github.io/aerialgaitdataset/

from a video captured by a UAV. Our solution consists of the following modules:

The *perspective correction module* compensates for perspective distortion in aerial videos using a projective transformation technique similar to [1,2,3]. Instead of one homography matrix, pre-annotated homography matrices are used for different levels of distortion caused by different camera elevation angles.

The *segmentation and feature extraction module* performs segmentation, using Histograms of Oriented Gradients (HOG) [4] or alternatively, Convolutional Neural Network (CNN) [5] features as shape descriptors. The model-free approach is combined with silhouette-based shape matching (in the case of HOG) or RGB-based shape matching (in the case of CNN) for efficient processing.

The *pose estimation module* uses a dynamic classifier selection architecture inspired by [6,7]. A total of 64 classes are defined for the combinations of 8 poses/gaits and 8 viewpoints in a human gait cycle. These views include front, back, side and diagonal views. Instead of performing 64-class classification, our dynamic classifier leverages the temporal relationships between poses and viewpoints, and performs significantly more reliable 4-class classifications instead.

The *trajectory estimation module* estimates the trajectory of the human subject by reconstructing the pose sequence using 3-D skeletons and localizing them with respect to the initial pose and viewpoint. The reconstructed poses provide the approximate shape of the ground-truth walking path.

The contribution of the paper is twofold.

- **A classifier architecture for efficiently and robustly estimating human gaits in monocular aerial videos.** By exploiting the temporal relationships between poses and viewpoints, our method can limit the wrongly estimated viewpoints to the adjacent viewpoints of the ground truth. The loss of limb and joint details in images usually leads to a left-right ambiguity issue, especially in front and back views [8]. However, our pose estimation solves this problem by taking into account possible temporal transitions between states. The dynamic classifier architecture presented in this work does not execute all the classifiers in the pool to make a decision. Instead, only the relevant classifier is selected based on the state transition graphs. This is a significant difference compared to similar architectures in the literature [6,9,10]. Experimental results confirm the proposed dynamic classifier is suitable for gait estimation in both ground and aerial videos.
- **The creation of a training dataset from an aerial platform for gait recognition.** This dataset

accounts for the natural twists and self-occlusions of a turning human body and minimizes the false positives caused by minor variations in heading.

The rest of this article is organized as follows. Section 2 discusses closely related work on perspective correction, pose estimation and their applications to UAV-based scenarios. Section 3 describes our solution. Section 4 reports experimental results. Discussion of issues and potential improvements is presented in Section 5. Section 6 concludes.

2 Related Work

This work is as an extension to the approach described in [11]. Compared to [11], here, we propose perspective correction for reducing errors of the dynamic classifier. Compensation for the perspective distortion is analyzed for videos captured at different heights/angles. Further, we combine transfer learning and dynamic classifiers to perform CNN-based classification. The dynamic classifiers are evaluated for HOG and CNN features. The performance measures are calculated with respect to the ground truth of test videos. The dataset used for this work will be publicly available. The study reported in [11] has been performed only with HOG features of silhouettes, and the ground truth pose information has not been considered for performance analysis.

The problem of recognizing human pose in statically captured videos has been studied extensively in recent literature [12]. Here, we discuss some closely related work.

2.1 Perspective correction

Perspective distortion needs to be corrected for processing that is robust to distance changes. Projective transformation, or homography, is an established approach for correcting perspective distortion [13], but this traditional approach requires the *vanishing point* to be manually specified. Rogez et al. [14,3] used *vertical scene lines* to estimate the vanishing point and localize the reconstructed poses based on the vanishing point, but their approach still requires manual determination of which lines are vertical. Our homography step is similar to Rogez et al.'s, except we determine the vanishing point based on the altitude and angle of the camera. Moreover, Rogez et al. conducted their study on statically captured videos, while we use video captured from a UAV.

2.2 Pose estimation by classification

Dynamic classifier selection (DCS) was originally proposed by Woods et al. [6], and is based on the *local accuracy estimation* of each individual classifier. Their approach selects an individual classifier which is most likely correct for a given sample. The final decision is made only by the selected classifier. Kuncheva [15] proposed a classifier selection and fusion method, but the experimental results show that DCS is the best performer while Kuncheva's is the second best, provided the classifiers have the same structure and training protocol. Ko et al. [7] developed a relatively similar classifier by integrating a majority voting system. Our classifier follows the DCS principles, but the best individual classifier is selected without executing the entire ensemble of classifiers.

Gaits estimated using direct classification [12] do not respect any temporal order [16,17]. A better approach is to consider the temporal relationship between poses, using techniques such as the ratio of the number of pixels in the intersection to the number of pixels in the union of two silhouette frames [18], dynamic time warping [19], lower limb joint angles [20] and frequency analysis of spatio-temporal gait signatures [21], to mention a few. Furthermore, general approaches to spatio-temporal action recognition can be found in [22,23,24, 25]. The temporal order of our reconstructed poses is based on the state transition model of poses and viewpoints.

Recent human pose estimation research has shown significant performance improvements by incorporating deep learning techniques [26]. The state of the art in human pose estimation has adopted convolutional neural networks as the main building block [27,28,29,30]. Deep learning models have also been adopted for human trajectory estimation in a range of settings. Some notable extensions are Recurrent Neural Networks (RNN) [31], Behaviour-CNN [32], unsupervised feature learning for classification and regression [33], and deep recurrent Long Short Term Memories (LSTMs) [34], to mention a few.

Left-right ambiguity is an inherent problem in silhouette-based pose estimation. Some strategies have been proposed in relation to depth information. Shotton et al. [35] trained classifiers to learn subtle visual cues from silhouettes to resolve the left-right ambiguity. In [36], the depth of each pixel in the silhouette was used to expand the 2D shape context into 3D space. Approaches have also been proposed for silhouettes in RGB images. Sigal et al. [37] proposed switching the left-right limbs of their graphical model to fit the silhouette with the smallest error. Some notable studies incorporated temporal information to infer the correct views of silhouettes. Both Discrete Cosine Transform (DCT) temporal prior in [38] and sequential clonal selection algorithm (CSA) in [39] handled this issue using the temporal continuity in images. Similarly, Lan et al. [40] selected the left vs. right configuration that is most consistent with the previous frame. Our proposed algorithm also uses temporal information between images but unlike the above methods, the transitions are determined based-on simple state transition graphs.

In recent literature, transfer learning approaches have been used effectively for human motion analysis problems. In a transfer learning setting, the first selected layers of a base network are used to develop the target network [41]. Notable work was done by Chaturvedi et al. [42], that employed deep transfer learning (DTL) to analyse the trajectories of basketball players using time-delayed Gaussian networks. Martín-Félez et al. [43] developed a system that learned gait features independently of the identity of people by applying transfer learning on a bipartite ranking model. A transfer learning approach similar to the one described in this paper can be found in [44], which introduced a framework for pose and gait estimation of elderly people. Their transfer learning was based on the Alexnet model [45], followed by a Siamese network to compare faces and upper/full bodies. Some related work focused on recognizing human actions across changes in the observer viewpoint [46,47]. Rahmani et al. [46] proposed a Robust Non-Linear Knowledge Transfer Model (R-NKTM) for action recognition from novel viewpoints. A similar problem was addressed by Farhadi et al. in [47] by training a discriminative appearance model. The authors used Maximum Margin Clustering to construct split-based features in the source view, then trained a classifier by transferring the splits in the source view to the target view.

Using UAV imagery for human pose or gait estimation is challenging due to platform mobility and susceptibility to wind gusts. UAVs are deployed in situations where it would be beneficial to interpret human movement, particularly in search and rescue applications [48], human-machine interface systems [49] and surveillance systems [50,51]. When employed in surveillance or search and rescue the movement of human subjects and their trajectory are vital information. Trajectory can be used for semantic analysis of human activities and prediction of future locations from video sequences [52]. Our vision-based trajectory estimation is relatively similar to [50] in terms of visual sensing.

2.3 UAV-based applications

Utilizing UAVs in human tracking and action recognition missions is a relatively new topic. Human detection methods from aerial videos have been suggested in relation to search and rescue missions [53,48]. The primary focus of these studies was to identify humans lying or sitting on the ground. Al-Naji et al. [54] used a hovering UAV to detect the vital signs of a human subject from the head and neck areas. Some studies focused on human identity recognition in low-resolution aerial videos. Oreifej et al. [55] presented an algorithm relying on a weighted voter-candidate formulation. The algorithm detects targets by analyzing the "blobs" of candidates against voters and addresses the need for human blob detection and tracking. Yeh et al. [56] proposed a relatively similar blob matching approach using an adaptive reference set of previously identified people. A system developed for UAV onboard gesture recognition was proposed by Monajjemi et al. in [57]. The system identified periodic movements of waving hands from other periodic movements like walking and running in an outdoor environment. Our experimental set-up is most similar to Monajjemi et al.'s. A crowd detection and localization approach using one UAV and a number of unmanned guided vehicles (UGVs) was presented in [58]. In contrast, our study uses a simpler configuration, but performs robustly on aerial video.

Table 1 Nomenclature

$\mathbf{H}$	Homography matrix.
ϕ, θ	Elevation angle, azimuth angle.
$\mathbb{R}$, $\mathbb{V}$, $\mathbb{P}$	Real number space, vector space, projective space.
V_i	The ith viewpoint, $i = 1, \ldots, 8$.
P_j	The jth pose, $j = 1, \ldots, 8$.
S	Training sample set.
K	Number of classes in a training set.
K'	Predicted class.
$\mathbf{M}$	An ECOC coding matrix.
C_{64}	The 64-class classifier invoked in the initialization stage.
$C_4(P, V)$	The 4-class classifier associated with pose P and viewpoint V.

3 Methodology

This section provides details of the perspective correction, segmentation and feature extraction, pose estimation and trajectory estimation modules. The block diagram of the entire process is given in Fig. 1. See Table 1 for the nomenclature used in this article.

3.1 Perspective correction

The relative orientation between the human subject on the ground and the camera in the sky is captured in a *horizontal coordinate system*, with coordinates ϕ and θ (see Fig. 2(a)). The camera viewpoint can take any (ϕ, θ) pair depending on the UAV position, where $\phi \in [0, \pi/2]$ and $\theta \in [0, 2\pi]$. A major problem with aerial photography is vertical perspective distortion, which occurs when $\phi > 0$, and worsens as ϕ gets larger. At low altitude the distorted human shape tends to have a large head and shoulders and small feet (see Fig. 2(b)). When $\phi = 90°$, perspective distortion cannot be corrected. For $60° \leq \phi < 90°$, the captured images have a severely distorted perspective that is difficult to accurately compensate. Therefore in this study, we limit the maximum ϕ to $60°$.

Perspective correction is done by mapping the distorted image plane (see Fig. 2(b)) to the undistorted vertical plane through *homography*. Segments on the undistorted vertical plane then enable the matching of test and training images.

A homography is a mapping from a projective space to itself. A *projective space* is an extension of Euclidean space in which two lines always meet at a point, and a point in the projective space is called a *homogeneous point*. Given an image, for every homogeneous point on the image plane, $\boldsymbol{x}$, there exists a homography matrix $\mathbf{H}$ [59, Section 3.1] that maps it to a homogeneous point, $\boldsymbol{x}'$, on the undistorted vertical plane, i.e.,

$$\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}.$$

The matrix $\mathbf{H}$ depends on the elevation angle ϕ. Instead of calculating $\mathbf{H}$ for each frame, we calculate it offline for each of the following values of ϕ:

- $\arctan(10/30) = 18.4°$,
- $\arctan(20/30) = 33.7°$,
- $\arctan(30/30) = 45.0°$,
- $\arctan(40/30) = 53.1°$;

and manually pre-annotate videos of the same elevation angle with the corresponding $\mathbf{H}$. To calculate $\mathbf{H}$, we manually select four points in a sample video frame to (i) delineate the area of interest and (ii) generate the vertical scene lines, as shown in Fig. 2(b). The vertical scene lines define the homography matrix $\mathbf{H}$.

3.2 Segmentation and feature extraction

After perspective correction, the human silhouette is segmented. The size of the silhouette in the image plane varies depending on the direct distance between the
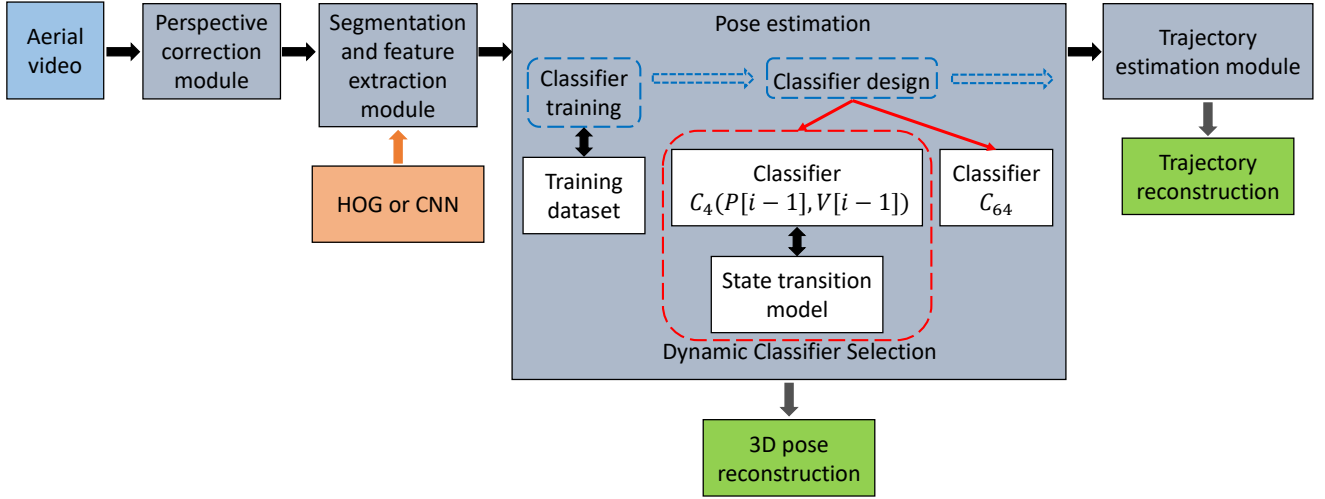
Fig. 1 The block diagram of the entire process. The perspective correction, segmentation and feature extraction, pose estimation: classifier training, pose estimation: classifier design, and trajectory estimation modules of the block diagram are explained from Sects. 3.1 to 3.5 in order.
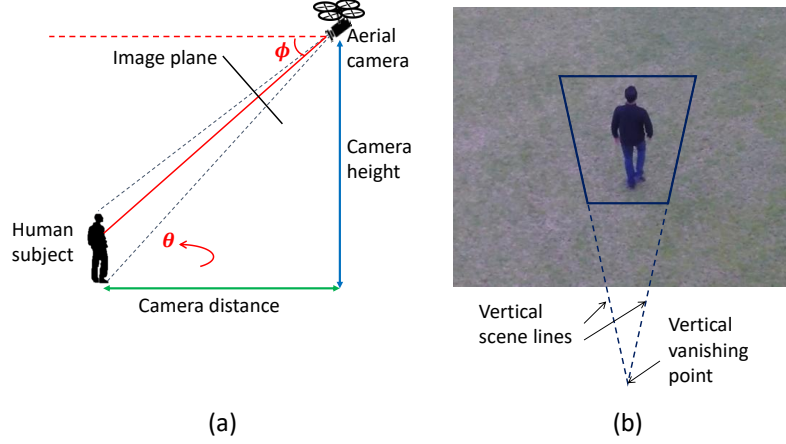

Fig. 2 (a) The UAV hovers at a known camera height and angle. In the horizontal coordinate system, $\phi \in [0, \pi/2]$ is the elevation/tilt angle, whereas $\theta \in [0, 2\pi)$ is the azimuth/pan angle. The azimuth angle is calculated in the radial direction between the heading direction of the human subject and the camera center axis on the horizontal plane. The observable space relative to the human observer is called the *viewing hemisphere*. (b) An input image with vertical scene lines. The scene lines are manually constructed according to the elevation angle. The blue box in the middle is the area of interest for homography in the vertical plane. The vanishing point is the point where parallel scene lines would meet each other on the image plane.

camera and the human subject. Perspective correction alone cannot address this scaling issue. Thus, the test silhouette is scaled up or down to match the scale of the training images. Prior to feature extraction, we use the online video annotation tool VATIC [60] to annotate the test videos. Two types of features are used, resulting in two variants of our scheme: HOG-based and CNN-based.

3.2.1 Feature extraction using HOG

For each frame, the RGB image is converted into a binary image and its bounding box area is segmented. Noise is removed using a Gaussian filter and small ob-

jects containing fewer than a threshold number of pixels are also removed. The remaining blob or blobs are considered to represent the human silhouette. Currently, the denoising parameters and segmentation parameters are customized for each video clip to obtain the best possible silhouette, so they are subject to improvements.

For feature extraction, the image window is divided into small spatial regions called "HOG cells" [4]. The weighted gradients in a HOG cell form a 1-D histogram which represents the orientation of the edge lines. The feature vector is formed from the HOG blocks, each of which represents a group of HOG cells.
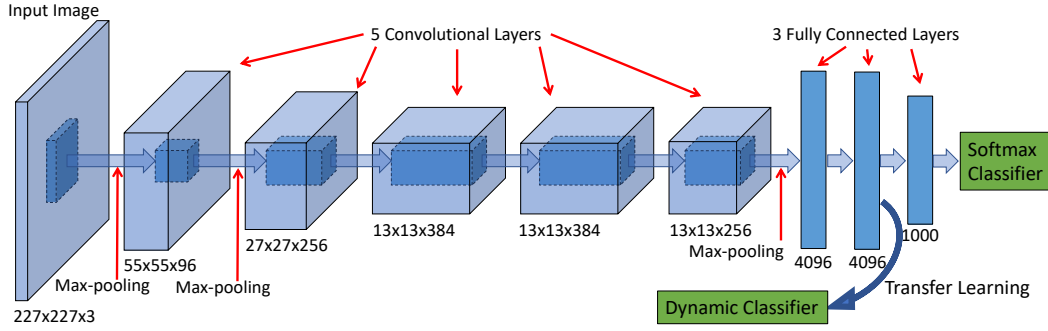
Fig. 3 The architecture of the pre-trained AlexNet deep convolutional neural network. The feature vector is extracted before the last fully connected layer and used to train the dynamic classifier.

3.2.2 Feature extraction using CNN

For each frame, the RGB image is cropped to meet the size requirement of the deep CNN AlexNet [45]. The AlexNet is a 11-layer network including 5 convolutional layers, 3 fully connected layers and 3 max pooling layers (see Fig. 3). The early convolutional layers have small receptive field sizes for learning low-level features, and later layers have larger field sizes for learning higher-level features. AlexNet has been pre-trained on 1.2 million ImageNet [61] images of 1000 classes, and showed the best performance in the ImageNet Large Scale Visual Recognition Challenge in 2012 [45]. Some classes of AlexNet are trained on images of humans in different settings; therefore, this pre-trained network was selected for our work.

In a pre-trained network, the weights for the deep layers are pre-determined. Instead of re-training AlexNet with our comparatively small dataset, we apply *transfer learning* in the standard way. We take the 4096-dimensional vector right before the last fully-connected layer of AlexNet as the feature vector (see Fig. 3). We then use the feature vectors to train an SVM classifier (as described in Section 3.4).

3.3 Pose estimation: classifier training

The training dataset is created to identify the eight sub-steps [62] of the human gait cycle (see Fig. 4). Each sub-step (or pose) has viewpoints from eight radial directions (azimuth angles that are 45° apart), giving rise to $8 \times 8 = 64$ pose-viewpoint pairs. The finite number of elevation-azimuth angle pairs are equivalent to the discretized viewing hemisphere described in [3,14,63].

We create two training datasets using silhouettes and color images. The silhouette dataset has 1017 images across 64 classes. The color image dataset contains images from our field videos, the MoBo Aligned
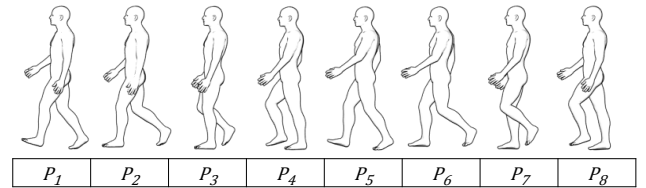

Fig. 4 A representation of the human gait cycle in 8 sub-steps: P_1 – both legs touch the ground; $(P_2 - P_4)$ – swinging the right leg; P_5 – both legs touch the ground; $(P_6 - P_8)$ – swinging the left leg.

dataset [64] and the HumanEva dataset [65]. To create this dataset, we collected images with varying perspective distortion for each class. The field images recorded from moving subjects have varying perspective distortions because the subjects walked in a circle (see the first three images in Fig. 7). The original MoBo images [66] were recorded on a treadmill using fixed cameras. We changed the backgrounds of MoBo Aligned images and manually added some perspective distortion to the images (see the second three images in Fig. 7) and no modifications were done to HumanEva images (the last two images in Fig. 7). We manually selected 4 points on the MoBo Aligned images and applied perspective transformation to get a birds eye view of them. We collected field images of 3 subjects, MoBo Aligned images of 15 subjects and HumanEva images of 2 subjects. The color dataset contains 8111 color images across 64 classes.

We used the entire training dataset (silhouettes or color images) to train the classifiers. The testing has been done using five selected videos (silhouettes or color images): a video each from CMU Motion of Body (MoBo) [66] and HumanEva2 [65] datasets (see Sect. 4.1 for more details) and three aerial videos (see Sect. 4.2 for more details). The annotated training and testing data will be publicly released in 2018.
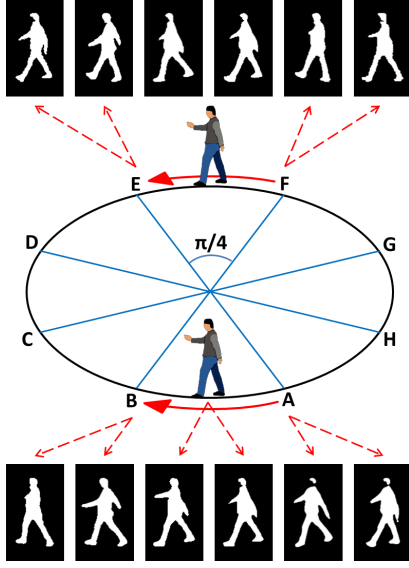
Fig. 5 The silhouettes in the top and bottom rows correspond to the first sub-step of human gait cycle shown in Fig. 4. For that particular pose there are eight possible viewpoints. This figure shows how the training images are selected for the shown pose and viewpoint. The broken arrows point to the selected silhouettes within the allowed orientation error margin. The starting point of the arrow roughly indicates the silhouette location on the circle. The silhouettes are independently selected from different image sequences captured under different lighting and noise conditions.

Figures 5–6 show only the silhouette images, but the same technique is followed to create the color dataset. An example of training data collection is shown in Fig. 5. The silhouettes in the figure correspond to the first substep of the human gait cycle shown in Fig. 4, namely P_1. The training data are collected at a camera distance of 30m and camera height of 10m (i.e., $\phi = 18.4°$), while the human subject walks on a marked circle of radius 5m in clockwise and anticlockwise directions. When walking from A to B on the circle, the orientation gradually changes up to 45° with respect to the orientation at A. In the training dataset, the images corresponding to the walk from A to B are considered as walking a straight line from A to B. This assumption can introduce a maximum of 22.5° orientation error. This error can be reduced by selecting more viewpoints (in other words selecting a viewpoint separation angle of less than 45°). However, we limit this study to eight viewpoints for simplicity and efficiency. The images captured at locations A, B, E and F on the circle have the maximum orientation error. Only the images captured at the mid-points of chords AB and EF represent the true orientation. The training images are selected in order to cover all of the possible heading directions within the accepted error margin ($\pm22.5°$).

The same procedure is followed to create the other 63 pose-viewpoint pairs as well. When the training dataset is used to estimate the poses in a test video of a person walking in a circle, the reconstructed path will not be a perfect circle (even assuming zero estimation errors) but a polygon. The reason for this polygon shape is the orientation angle error associated with each viewpoint.

One advantage of the training data collection method above is it accounts for the natural twists and self-occlusions of walking better than collecting data only from walking straight. For example, the images captured at points A and E in Fig. 5 have the same orientation error. However, the silhouettes of the same pose can hold differences as the person at A turns to his right and the person at E turns to his left on the circle. Another significant advantage is it reduces the false positives of the classifier arising from slight variations of heading. By including slightly oriented silhouettes (with respect to walking straight) in the dataset, we approximate all of the small variations in orientation to be within the range $[0°, 22.5°]$. This is a useful approximation when analyzing real walking patterns of human subjects because most of the time, people walk in straight lines and do not change their orientation frequently.

The collected training data consists of 64 labels, representing eight sub-steps of the gait cycle and eight viewpoints (see Fig. 6). For each label, the training data consists of silhouettes (for HOG) and RGB images (for CNN) created under different illuminations and orientations. Some sample images of class P_5V_4 are shown in Fig. 7.

3.4 Pose estimation: classifier design

We denote a training dataset of n observations by

$$S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}, \tag{1}$$

where $\boldsymbol{x}_i$ is the ith feature vector, and y_i the ith label. Suppose $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^m$, where m is the dimension of a feature vector; and $y \in \mathcal{Y} = \{1, ..., K\}$, where K is the number of classes. We can formulate the pose estimation problem, like most classification problems in computer vision [67], as a K-class classification problem: finding $f : \mathcal{X} \to \mathcal{Y}$ such that the classification error is minimized.

However, many real world problems are multiclass problems, $K > 2$. A standard way to create multiclass classifiers such as multiclass SVM is to map a multiclass problem onto many, possibly simpler, twoclass problems [67]. A potential solution is a classifier combination method, the basic idea of which is to exe-
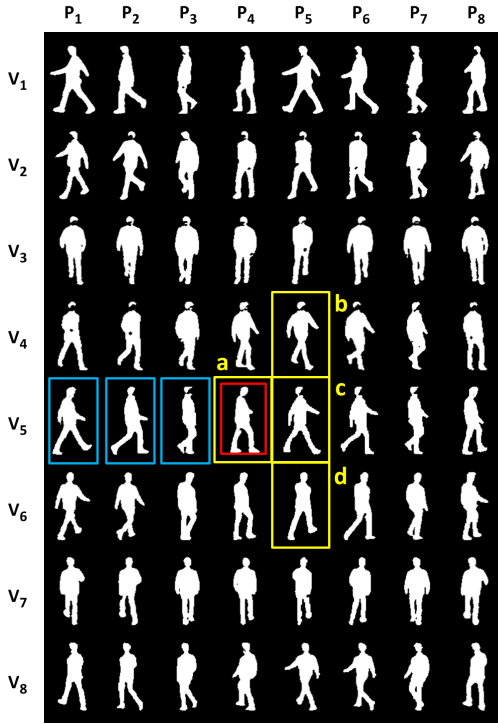
Fig. 6 The training dataset consists of 64 classes of pose-viewpoint pairs. The rows and columns represent the poses and the viewpoints respectively. Each silhouette in the figure is a random image from the pose-viewpoint subset it belongs. The initialization is performed using the classifier trained with the complete dataset (64 classes), namely C_{64}. Blue- and red-border windows show four consecutive frames (from left to right) initialized by this classifier. Once initialized, the pose and the viewpoint of the most recently initialized image (red-border window) are used to select the next classifier. In this example, the training image subsets of the next classifier are shown in yellow-border windows.

cute an ensemble of classifiers, and combine their outputs through a voting system, combination function, or weighting function [10]. In such a design, although each classifier is trained with a subset of the entire training set, each iteration involves the entire ensemble of classifiers. As a more efficient solution, we propose a *dynamic classifier selection* architecture, combining (i) a state transition model for the pose and viewpoint and (ii) an SVM-based error-correcting output codes (ECOC) framework [68] for our multiclass pose-viewpoint classification problem. Next, we will discuss this state transition model and ECOC framework in turn.

3.4.1 Viewpoint and pose transition

We model a pose-viewpoint pair as a state, and the transition of states using a state transition graph. This graph should not be confused with a Markov chain, because we do not assign probabilities to state transitions. Our state transition model is similar to Lan et al.'s model [69], with the differences being:

- Lan et al. used 8 viewpoints and different numbers of poses per viewpoint, whereas we use 8 viewpoints and 8 poses per viewpoint;
- each of their side views and 45° views is associated with 4 poses, and each of their front and back views is associated with 1 pose, resulting in a total of 26 states; whereas, our model consists of 64 states.

Our state transition graphs (see Fig. 8) are constructed based on the assumption that the human subject walks forward at a constant speed, does not take sharp turns and does not twist their body while turning. This assumption does not preclude left, right, or backward turns, as long as the turn is not abrupt, as exemplified by the yellow-border windows in Fig. 6. The state transition graphs establish a temporal relationship between the states, and dictate admissible state transitions, on which the classification outputs are based.

The admissible state transitions restrict the next classifier prediction to be one of the states the current state can transition to. Given the current pose and viewpoint, when a new image is available, the associated pose is predicted to be either the current pose (conceivably, the same pose appears in multiple consecutive frames when the video frame rate is high), or the pose in the next sub-step of the gait cycle (see Fig. 4). When the pose changes from the current state to the next state, the viewpoint of the next pose has to be one of the following: the same viewpoint (moving straight), 45° clockwise from the current viewpoint (turning left), or 45° anticlockwise from the current viewpoint (turning right).

3.4.2 Classification with error-correcting output codes (ECOC)

Considering the pose-viewpoint pairs as multiple classes in the problem domain, we select the ECOC framework [68] for multiclass classification. The ECOC is considered to be a powerful and popular multiclass classification technique [70]. Good results have been reported in [70,71,72,73] using ECOC for different multiclass classification problems.

The ECOC framework uses a set of binary classifiers to solve a multiclass classification problem. In a problem domain of K classes, the ECOC framework forms N binary problems (dichotomizers), where $N > K$. Dietterich et al. [68] represent the ECOC model of N binary problems using a coding matrix $\mathbf{M} = [m_{k,n}] = \{-1, +1\}^{K \times N}$, where each row encodes an N-dimensional binary vector (a codeword), and each column is used to

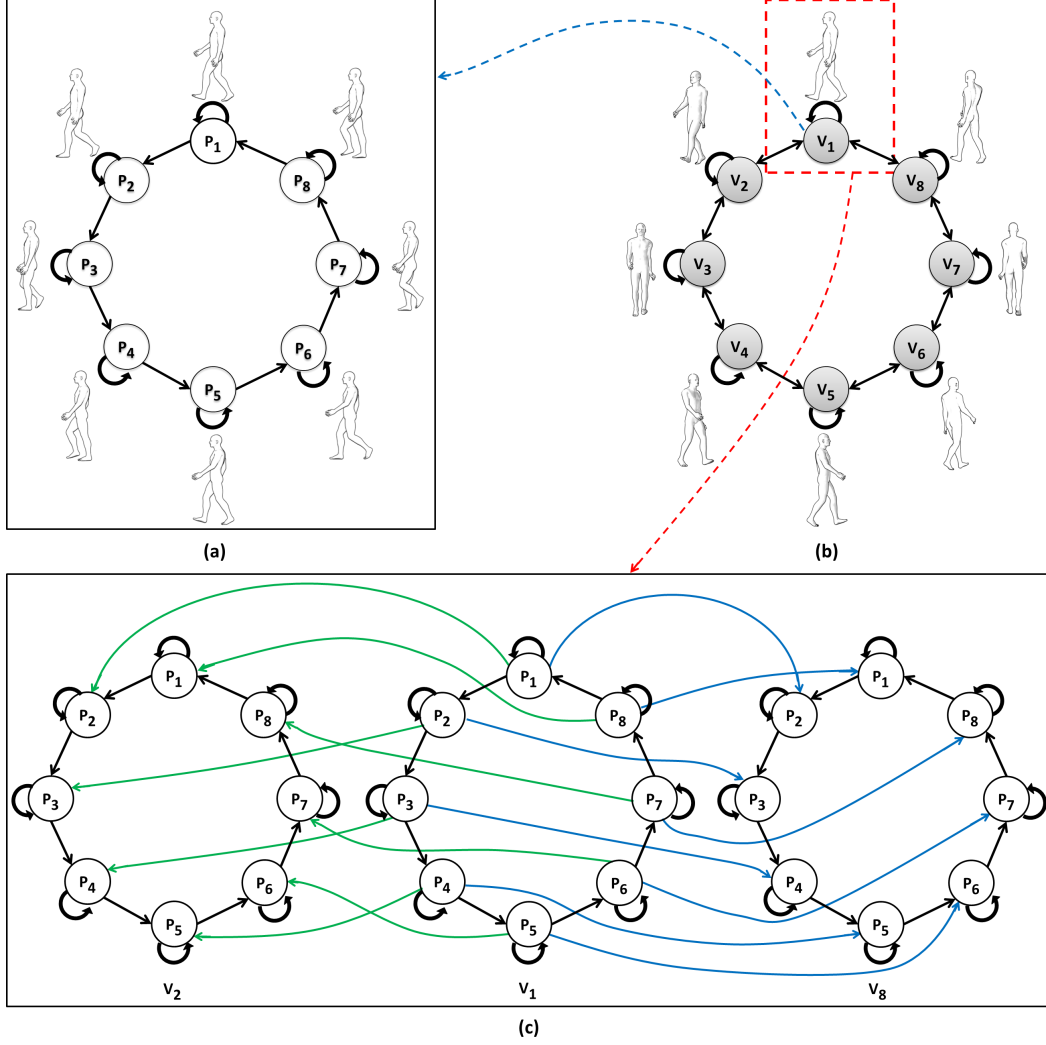**Fig. 7** Some sample images from the color training dataset.

Fig. 8 Our state transition model for the pose and viewpoint. Here, V_i represents "viewpoint i" and P_j represents "pose j". (a) The pose transition graph for viewpoint V_1. (b) The viewpoint transition graph, where each node corresponds to eight poses. Each viewpoint can transition to one of the immediately adjacent viewpoints, or can stay as itself. (c) The pose and viewpoint transition graphs for the selected viewpoint in (b), namely V_1.

train a binary learner. The coding design is such that $+1$ represents a positive example of a class, whereas -1 represents a negative example.

We use the ternary ECOC framework proposed by Allwein et al. [74] that follows the steps below:

– The coding matrix is defined as $\mathbf{M} = \{-1, 0, +1\}^{K \times N}$, where 0 tells the binary learner to ignore the corresponding class during training.

– We use one-versus-one [75] coding design which constructs $K(K-1)/2$ binary learners.
– The selected decoding scheme is loss-based decoding [74], and the binary learner is an SVM learner.
– In the classification stage, when an input x is available, the vector of predictions $\boldsymbol{f}(x) = [f_1(x) \; \cdots \; f_N(x)]$ is formed from the predicted outputs of the N classifiers.

– The predicted class is the class that minimizes some loss function L [74, Equation (5)]:

$$K' = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmin}} \sum_{n=1}^{N} L(m_{k,n}, f_n(x)). \qquad (2)$$

3.4.3 Classifier combination by dynamic classifier selection (DCS)

Our DCS architecture consists of a single 64-class SVM classifier denoted C_{64}, and 64 4-class SVM classifiers denoted $C_4(P_i, V_j)$, $i, j \in \{1, \ldots, 8\}$. The classifier $C_4(P_i, V_j)$ is associated with pose P_i and viewpoint V_j, and is trained to recognize the set of four classes:

$$\{(P_i, V_j), (P_{i \boxplus 1}, V_j), (P_{i \boxplus 1}, V_{j \boxminus 1}), (P_{i \boxplus 1}, V_{j \boxplus 1})\}, \qquad (3)$$

where $i, j \in \{1, \ldots, 8\}$ and the operators $\boxplus, \boxminus$ are defined as follows:

$$i \boxplus j = (i + j + 1) \mod 8 - 1, \qquad (4)$$
$$i \boxminus j = (i - j - 1) \mod 8 + 1. \qquad (5)$$

For example, the classifier $C_4(P_4, V_5)$ is trained to recognize the four classes labeled a, b, c and d in Fig. 6.

As depicted in Algorithm 1, our classification process works in two stages: (i) the initialization stage and (ii) the DCS stage. In the initialization stage, the first q video frames are classified using classifier C_{64}. The DCS stage starts with the $(q+1)$th video frame. In this stage, each frame is classified with a classifier chosen based on the class label predicted by the previous iteration.

To elaborate, consider the example in Fig. 6. Suppose $q = 4$, and the blue- and red-border windows are sample classes predicated by the classifier C_{64}. The red-border window highlights the class predicted for the qth frame. Since this class is (P_4, V_5), the classifier $C_4(P_4, V_5)$ is chosen to classify the $(q+1)$th frame. The training subsets for $C_4(P_4, V_5)$ are highlighted with the yellow-border windows a, b, c and d. A training subset refers to the images corresponding to a single class [7].

The most significant difference between the classifier architecture presented here and architectures in the recent literature [6, 9, 10] is that this architecture does not execute all of the classifiers to make a decision. Instead, only the relevant classifier is selected for every next image. The relevance of the classifier is determined by its training subsets, and the training subsets are selected based on the state transition graphs.

In Algorithm 1, the most resource-demanding component is ECOC SVM classification. The time and space complexities of this component are $\mathcal{O}(n_{\mathrm{sv}})$ and $\mathcal{O}(n_{\mathrm{sv}}m)$ respectively, where n_{sv} is the number of support vectors, and m is the number of features.

---

1   **input**: Labeled test images;
2 $i = 1$;
3   **while** *the ith video frame* **do**
4   **if** $i \leq q$ **then**
5   Initialization stage: apply classifier C_{64};
6   **else**
7   DCS stage: apply classifier $C_4(P[i-1], V[i-1])$;
8   **end**
9 $P[i] \leftarrow$ Predicted pose;
10 $V[i] \leftarrow$ Predicted view;
11 $i = i + 1$;
12   **end**

Algorithm 1: Algorithm for pose-viewpoint classification by dynamic classifier selection. Prior to the workflow, all classifiers should have been trained. Upon entering the workflow, classifier C_{64} is used to classify the first q video frames — a higher value of q implies higher conservativeness. Based on the output of C_{64} on the qth frame, a classifier $C_4(P[q], V[q])$ is selected and used to classify the $(q + 1)$th frame. Thereafter, $C_4(P[i-1], V[i-1])$ is used to classify the ith frame. Each classifier $C_4(\cdot, \cdot)$ recognizes only four classes.

When using HOG features, the SVM model was trained using a one-versus-one coding design, which involves $K(K-1)/2$ support vectors. Final cropped silhouettes were resized to 96×160 pixels. The HOG cell size was selected to be 4×4 resulting in a 32292-dimensional feature vector. In this case, for the 4-class dynamic classifier, n_{sv} is 6 and m is 32292.

When using CNN features, the SVM model was trained using a one-versus-all coding design, which involves K support vectors. In this case, for the 4-class dynamic classifier, n_{sv} is 4 and m is 4096.

3.5 Trajectory estimation

Trajectory estimation refers to estimation of the shape of path traversed by the human subject. Trajectory estimation is performed using the estimated viewpoints as inputs and thus, require the classifier to have minimal viewpoint estimation errors. The estimated trajectory is inevitably a polygonal approximation of the actual shape of the path. Various interpolation techniques could be applied to smoothen the estimated trajectory and thereby improve the approximation.

As shown in Algorithm 2, each estimated viewpoint serves as an estimation of the walker's orientation. For each estimated orientation, a 3-D pose is reconstructed from the estimated pose. The algorithm can be thought of as primarily handling two cases:

```
 1  input: Estimated pose and viewpoint;
 2  i = 0;
 3  Label 1: i = i + 1;
 4  Obtain the i^th estimated pose and viewpoint;
 5  if i = 1 then
 6  |   Initialization: calculate orientation, reconstruct
    |     3-D pose;
 7  |   go to Label 1;
 8  else
 9  |   if pose changed? then
10  |   |   if viewpoint changed? then
11  |   |   |   Calculate new orientation;
12  |   |   |   Reconstruct 3-D pose;
13  |   |   |   Rotate towards new orientation;
14  |   |   |   Translate along new orientation;
15  |   |   |   go to Label 1;
16  |   |   else
17  |   |   |   Reconstruct 3-D pose;
18  |   |   |   Translate along previous orientation;
19  |   |   |   go to Label 1;
20  |   |   end
21  |   else
22  |   |   go to Label 1;
23  |   end
24  end
```
Algorithm 2: Algorithm for trajectory estimation.

- Whenever an estimated pose is the same as the previous, the reason is assumed to be the camera's high frame rate and/or the subject's slow movement, and thus the subject is assumed to remain at the same location.
- Whenever an estimated pose differs from the previous, the subject is assumed to have moved a fixed distance from the location of the previous pose. When the orientation changes by x degrees, the next pose is positioned at a fixed distance from the location of the previous pose at an angle of $\pm x$ degrees ($+$ve for right turns, $-$ve for left turns). Due to the way the viewpoint angle is discretized, as explained in Section 3.3, x is a multiple of $360°/8 = 45°$.

The trajectory estimation algorithm uses the dynamic classifier's ability to resolve the left-right ambiguities of images. Without dynamic classifier selection, the classifier C_{64} can make errors between the front and back views (rows 3 and 7 in Fig. 6), as a result of self-occlusions, or loss of joint angle and limb length information after binary conversion. The time and space complexities of Algorithm 2 are both $\mathcal{O}(1)$.

4 Experimental results

We conducted three groups of experiments, which we discuss in the subsequent subsections. Across the experiments, the scenery and walking patterns vary sig-

nificantly. These experiments include view variations between front, diagonal, side and back views.

Pose/viewpoint estimation errors are expressed in terms of (i) classification errors and (ii) viewpoint and pose transitional errors (TE). We define *transitional errors* as follows:

Definition 1 *If the classifier prediction is different from the ground truth but is confined between the adjacent viewpoints (or poses), such predictions are considered as* viewpoint transition errors *(or* pose transition errors*).*

For example, when the ground-truth pose transitions from P_1 to P_2, given the similarity of the poses, it is likely for the classifier to still identify ground truth P_2 as P_1 for a few frames. Likewise, the classifier is likely to misclassify P_1 as P_2 before the ground-truth transition occurs. In other words, transitional errors can delay or advance true pose/viewpoint estimation. In this example, the performance measures without transitional errors are calculated as follows: when the ground truth is P_2, the predicted adjacent poses (P_1 and P_3) are considered to be true predictions, and all the other estimations are considered to be incorrect. Hereafter, we use the abbreviation TE in the equations, tables and figures to refer to transitional errors.

We now define pose/viewpoint estimation errors formally:

Definition 2 *The percent **pose estimation error**, including transitional errors, is*

$$e_{pose,\ with\ TE} \overset{\text{def}}{=} \frac{\left| \#frames\ with\ misclassified\ poses \right|}{\#frames} \times 100\%.$$
(6)

*The percent **pose estimation error**, excluding transitional errors, is*

$$e_{pose,\ no\ TE} \overset{\text{def}}{=} \frac{\left| \begin{array}{c} \#frames\ with\ misclassified\ poses \\ -\#frames\ with\ pose\ TE \end{array} \right|}{\#frames} \times 100\%.$$
(7)

*The percent **viewpoint estimation error**, including transitional errors, is*

$$e_{viewpoint,\ with\ TE} \overset{\text{def}}{=} \frac{\left| \begin{array}{c} \#frames\ with\ misclassified \\ viewpoints \end{array} \right|}{\#frames} \times 100\%.$$
(8)

*The percent **viewpoint estimation error**, excluding transitional errors, is*

$$e_{viewpoint,\ no\ TE} \overset{\text{def}}{=} \frac{\left| \begin{array}{c} \#frames\ with\ misclassified \\ viewpoints \\ -\#frames\ with\ viewpoint\ TE \end{array} \right|}{\#frames} \times 100\%.$$
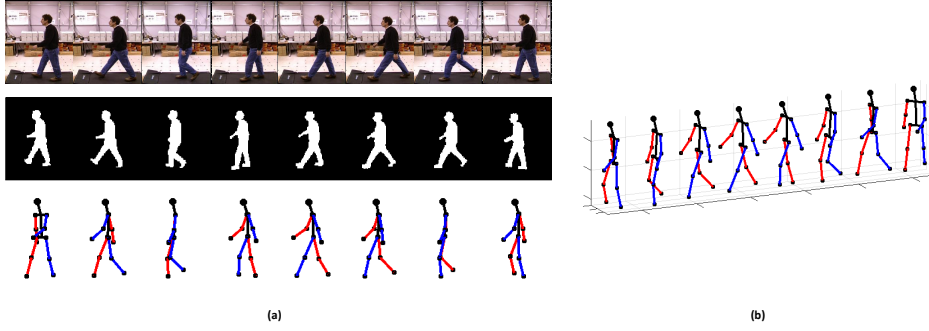
(a)　　　　　　　　　　　　　　　　　　　　　(b)

Fig. 9 Partial results for the CMU MoBo experiments: (a) The three rows represent the original images for subject 4071, the segmented silhouettes and the estimated poses respectively. (b) 3-D reconstruction of the estimated poses and trajectory of (a), where the rightmost skeleton corresponds to the leftmost image in (a). Red and blue colors mark the right and left sides of the body respectively.
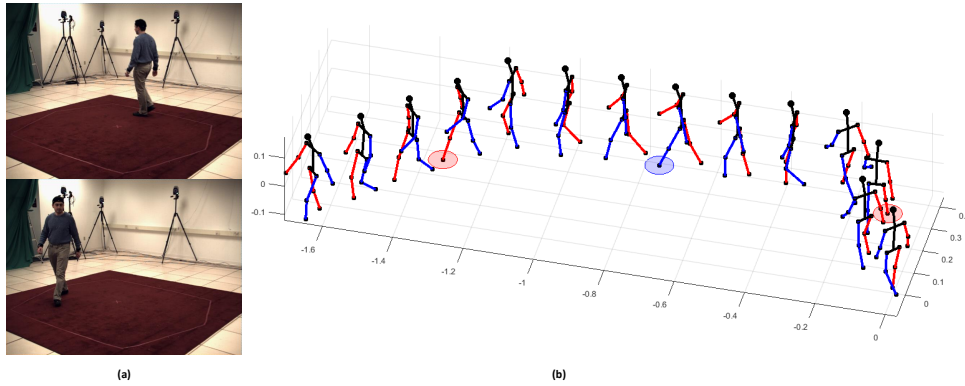

(a)　　　　　　　　　　　　　　　　　　　　　(b)

Fig. 10 Partial results for the HumanEva2 experiments: (a) The first and last images of the sequence for subject S2 combo C2. (b) 3-D reconstruction of the estimated poses and trajectory. Red and blue discs mark the sub-steps of the human gait cycle where both feet touched the ground — red when the right foot was in front and blue when the left foot was in front. The 3-D poses between two red or two blue markers represent a complete gait cycle (8 poses).

$$\tag{9}$$

For trajectory estimation, each estimated trajectory is plotted on a 2-D plane with *unitless* axes, and the starting location mapped to the origin. Along a trajectory, the estimated poses are reconstructed using Rogez et al.'s 3-D, 13-jointed skeletal models [76]. The proximity of the estimated trajectories to the actual trajectories was assessed.

4.1 Experiments with publicly available datasets

In this group of experiments, we used two publicly available human motion datasets: (i) CMU Motion of Body (MoBo) [66] and (ii) HumanEva2 [65]. Both CMU MoBo and HumanEva2 are recorded indoors by a ground-based camera with a static background. For these datasets, background subtraction is used for foreground/background segmentation, and tested only with HOG features.

From the CMU MoBo dataset, the image sequence for subject 4071 was selected, which shows the subject

walking on a treadmill at a constant speed. Fig. 9(a) shows the original images, the segmented silhouettes and the estimated poses. The reconstructed trajectory in Fig. 9(b) was a straight path, but the orientation was skewed by 45° due to the 45° error in the first estimated viewpoint. Table 2 shows that the dynamic classifier had significantly lower values for e_{pose} and $e_{\text{viewpoint}}$ than C_{64}.

From the HumanEva2 dataset, the image sequence for subject S2 combo C2 was selected, which shows the subject rounding a left turn. Figure 10(b) shows the 3-D reconstruction of the estimated poses and trajectory. The viewpoint of the third skeleton is incorrectly estimated as its adjacent viewpoint (+45° error). However, when moving from the third skeleton to the fourth one, the viewpoint was corrected. Table 2 shows that the dynamic classifier had significantly lower values for e_{pose} and $e_{\text{viewpoint}}$ than C_{64}.

Table 2 Estimation errors of C_{64} and the dynamic classifier using HOG features.

Experiment/dataset	#frames	$e_{\text{pose, with TE}}$		$e_{\text{viewpoint, with TE}}$	
		C_{64}	Dynamic classifier	C_{64}	Dynamic classifier
CMU MoBo	35	62.9%	**0**%	14.3%	**5.7**%
HumanEva2	130	73.8%	**59.2**%	31.5%	**19.2**%
Scenario 1 ($h = 2$m)	250	48.8%	**36.8**%	6.4%	**4.8**%
Scenario 2 ($h = 10$m)	784	30%	**23.5**%	11.9%	13%
Scenario 3 ($h = 10$m)	1652	27.5%	**23.5**%	**16.2**%	16.9%

Table 3 Estimation errors of the dynamic classifier using HOG/CNN features on UAV-captured videos.

Experiment	#frames	$e_{\text{viewpoint}}$		e_{pose}	
		CNN	HOG	CNN	HOG
Scenario 1 ($h = 2$m), with TE	250	22.8%	**4.8**%	**34**%	36.8%
Scenario 1 ($h = 2$m), no TE	250	**0**%	**0**%	3.2%	**1.2**%
Scenario 2 ($h = 10$m), with TE	787	44.5%	**13**%	52.7%	**23.5**%
Scenario 2 ($h = 10$m), no TE	787	15.7%	**0**%	30.4%	**3.2**%
Scenario 3 ($h = 10$m), with TE	1652	30.3%	**16.9**%	41.5%	**23.4**%
Scenario 3 ($h = 10$m), no TE	1652	16.9%	**0.4**%	17%	**3.8**%

4.2 Experiments with video captured from a UAV

In this group of experiments, three video datasets representing three different scenarios were captured from a rotorcraft UAV — specifically, a 3DR Solo — in a slow and low-altitude flight mode. For recording videos, we use a GoPro Hero 4 black camera with an anti-fish eye replacement lens (5.4mm, 10MP, IR CUT) and a 3-axis Solo gimbal. The images were sampled at a rate of 30fps. In order to ease the segmentation process, the videos were recorded with an uncluttered background and with the human subject wearing dark clothes. The UAV-captured videos are segmented as described in Section 3.2. These experiments were conducted using both HOG and CNN features.

Certain assumptions were made to ease the coordinate transformation between the camera and the human subject:

- The human subject stands upright on flat ground.
- The camera roll angle is zero.
- The roll, pitch and yaw angles of the UAV are zero during slow flight. Thus, the flight dynamics of the UAV has negligible effects on the camera elevation angle.
- The human subject is *approximately* centered in the video.

These are valid assumptions in the case of an aerial platform designed to track human motion in a large field of view (see Fig. 11). The camera elevation angle and height were directly recorded from the UAV control interface. The UAV was operated at a known ground distance (camera distance) from the human subject.
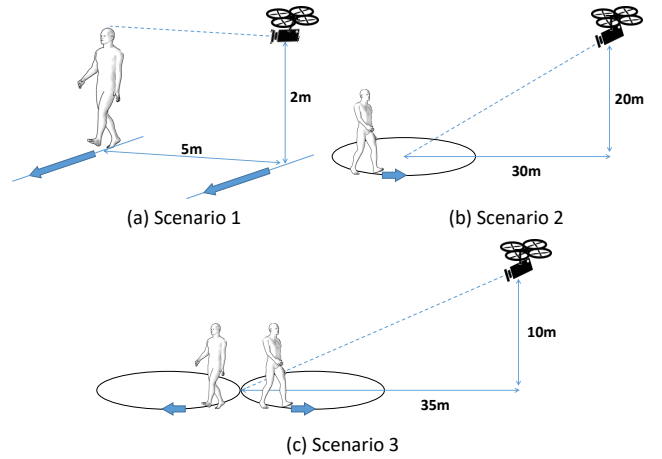

Fig. 11 How the videos for the three scenarios in Section 4.2 were recorded by a UAV: (a) A human subject was filmed from his left walking along a straight line. The UAV moved in synchrony with the subject. (b) A human subject was filmed walking in a circle, while the UAV stays pointed at the center of the circle. (c) A human subject was filmed walking on a figure 8 shape, while the UAV stays pointed at the center of the shape.

4.2.1 Scenario 1

As depicted in Fig. 11(a), a human subject was filmed from his left walking along a straight line, by a UAV moving in synchrony. To achieve synchrony, the UAV was manually operated to maintain roughly the same speed as the subject, at a constant ground distance of 5m from the subject. The camera was horizontal and 2m above ground. Here are the findings:

- Table 2 shows that the dynamic classifier had lower values for e_{pose} and $e_{\text{viewpoint}}$ than C_{64}.
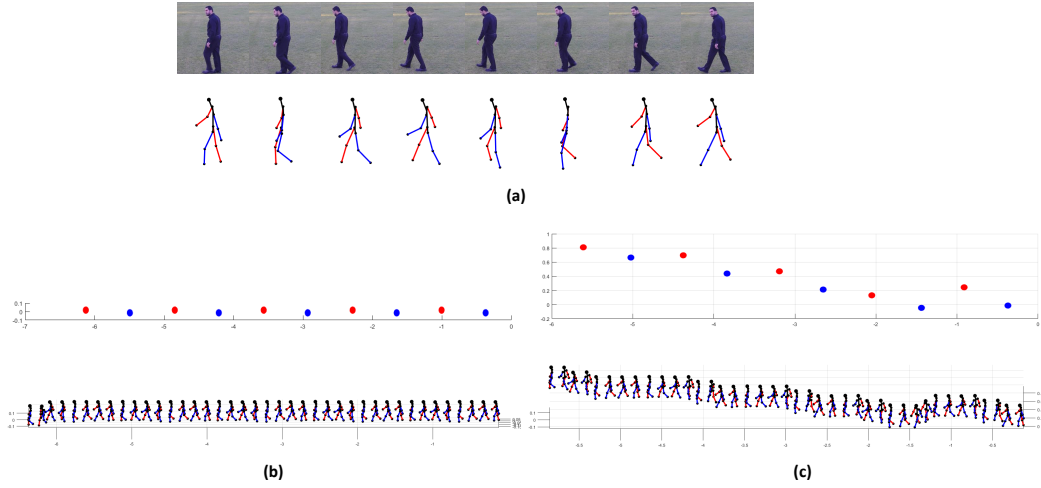
Fig. 12 Partial results for the Scenario 1 experiments, where a UAV follows a human subject on their left. (a) Some sample input images and their HOG-based estimated poses. (b) The estimated trajectory using HOG features. (c) The estimated trajectory using CNN features.
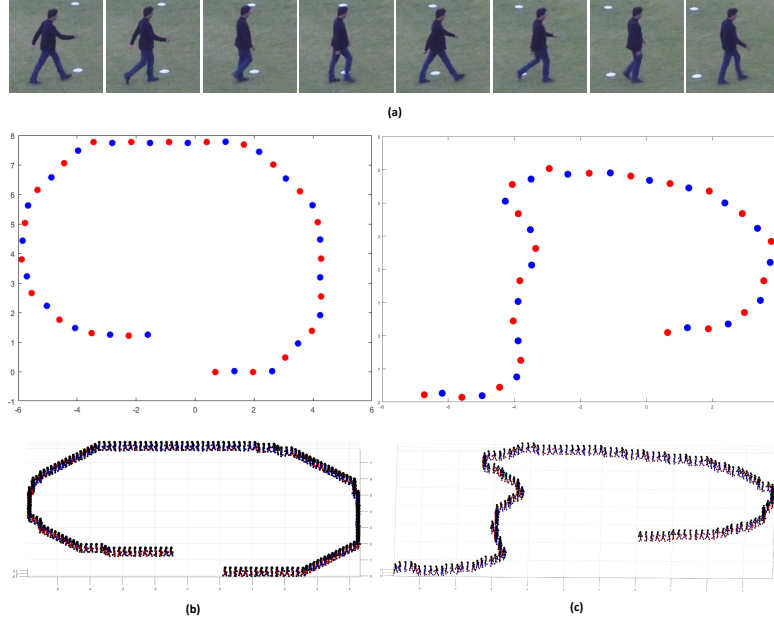

Fig. 13 Partial results for the Scenario 2 experiments: (a) The top row shows a series of cropped video frames. (b) The estimated trajectory using HOG features, where each dot marks where both feet touch the ground, with red representing right foot in front and blue representing left foot in front. 3-D reconstruction of the estimated poses and trajectory is also shown. (c) The estimated trajectory and 3-D reconstruction of the estimated poses using CNN features.

– When comparing HOG and CNN in Table 3 and Figs. 15–16, significantly lower values for $e_{\text{viewpoint}}$ and slightly higher values for e_{pose} can be seen for HOG. Zero to very low errors are observable once the transitional errors were removed.

– Figure 12 shows a successful 3-D reconstruction of the estimated poses and trajectory for a segment of the path using HOG and separately CNN features.

– The confusion matrix in Fig. 17(a) shows viewpoint confusion was rare and confined to a neighbor of the true viewpoint.

4.2.2 Scenario 2

As depicted in Fig. 11(b), a human subject was filmed walking on a marked circle by a UAV pointing at the center of the circle. The camera was 30m from the cen-
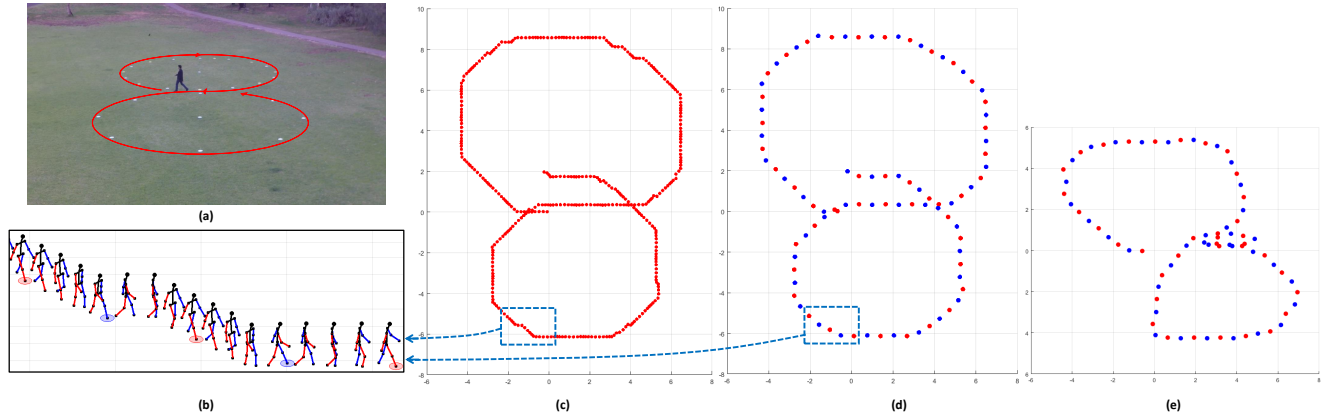
Fig. 14 Partial results for the Scenario 3 experiments ((b)-(d) correspond to HOG and (e) corresponds to CNN): (a) An original image of a human subject walking on a marked figure 8 path. (b) 3-D reconstruction of the estimated poses and trajectory, for a segment of the path shown in (c). (c) Tracking of the right foot of the human subject. Each point represents the estimated location of the right foot at each sub-step of the gait cycle. (d) Red and blue points represent where the right and left foot touches the ground respectively. (e) The estimated trajectory using CNN features.
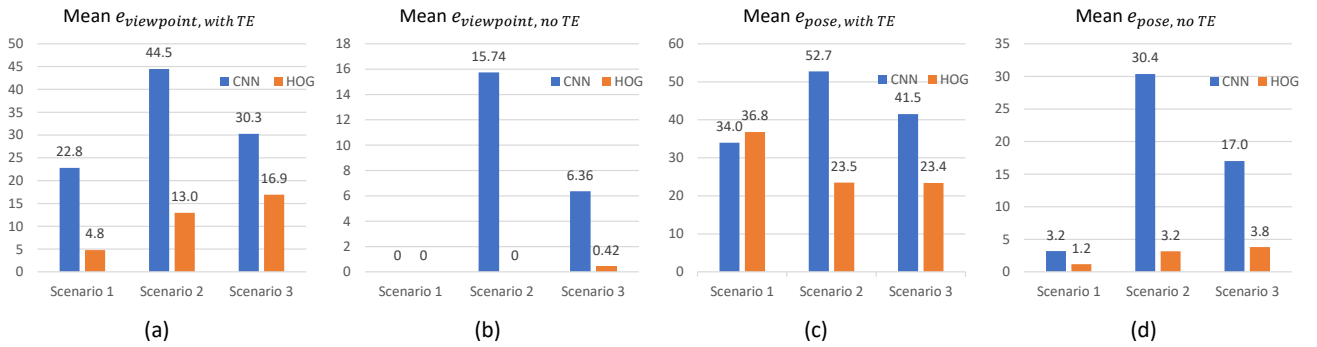

Fig. 15 Mean viewpoint and pose estimation error (%).

ter of the circle and 10m above ground. Here are the findings:

- Table 2 shows that the dynamic classifier had a significantly lower e_{pose} than C_{64} and a slightly higher $e_{viewpoint}$ than C_{64}.
- Table 3 shows that CNN gives higher estimation errors than HOG does. The errors for CNN dropped significantly upon removal of the transitional errors.
- Figure 13(b) shows the HOG-based estimated trajectory is approximately circular, as is the true trajectory. It also shows a 3-D reconstruction of the estimated poses and trajectory, with a small number of visibly wrong viewpoints. Figure 13(c) shows the trajectory estimated using CNN features failed to follow the ground truth in the second half of the circular path.
- The confusion matrix in Fig. 17(b) shows viewpoint confusions are confined to neighbors of the true viewpoints. The lowest confusion rates are associated

with the diagonal viewpoints V_2, V_4, V_6 and V_8; whereas, high classification accuracy has been recorded for V_1, V_3, V_5 and V_7. The reason is viewpoints V_1, V_3, V_5 and V_7 correspond to the front, back and side views. These four viewpoints suffer minimal self-occlusions in the silhouettes compared to the others, and hence, provide better image details.

4.2.3 Scenario 3

As depicted in Fig. 11(c), a human subject was filmed walking on a marked 8-shaped path by a UAV pointing at the center of the path, which was created by joining two circles of radius 5m. The walk starts and ends at the same point in the marked path. The camera distance from the middle of the path, where the two circles meet, was 35m and the camera height was 10m. Perspective distortion of the video frames was negligible due to the small elevation angle, and so, the video frames were
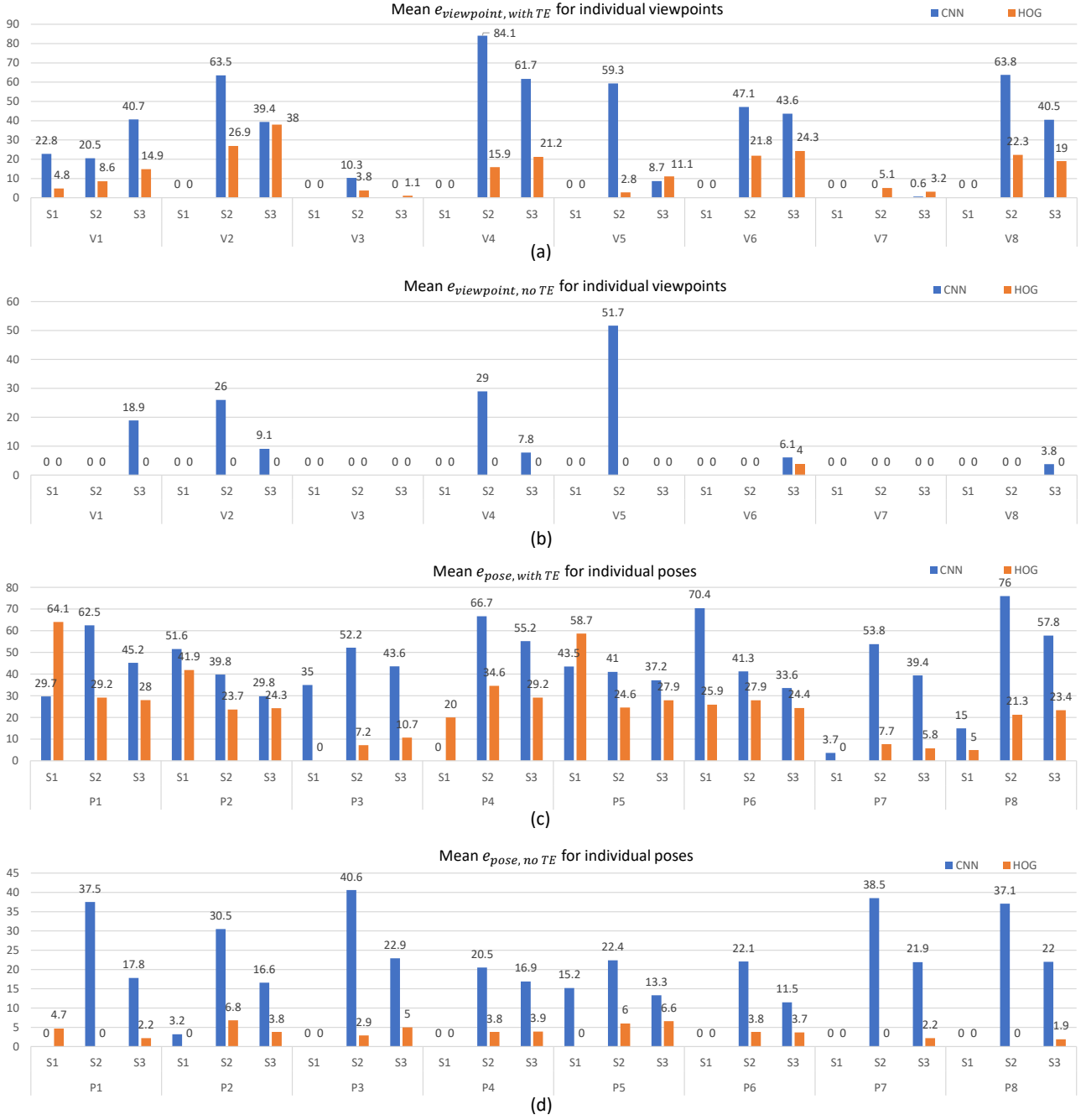
Fig. 16 Mean viewpoint and pose estimation error (%) for each viewpoint and pose. S_1-S_3 denote Scenarios 1–3 respectively.

segmented without perspective correction. Here are the findings:

– Table 2 shows that the dynamic classifier has a significantly lower e_{pose} than C_{64} and a comparable $e_{viewpoint}$ to C_{64}.

– Table 3 shows CNN gave higher estimation errors than HOG does. These results are consistent with those for Scenarios 1–2.

– Figure 14(b) shows a 3-D reconstruction of the estimated poses and trajectory using HOG features for a segment of the path, which contains some visibly wrong viewpoints. However, Figs. 14(c) and 14(d) show the estimated trajectory approximates the a

	V1	V2	V3	V4	V5	V6	V7	V8
V1	96.8	3.2	0	0	0	0	0	8.6
V2	0	0	0	0	0	0	0	0
V3	0	0	0	0	0	0	0	0
V4	0	0	0	0	0	0	0	0
V5	0	0	0	0	0	0	0	0
V6	0	0	0	0	0	0	0	0
V7	0	0	0	0	0	0	0	0
V8	0	0	0	0	0	0	0	0

(a) Scenario 1

	V1	V2	V3	V4	V5	V6	V7	V8
V1	91.4	0	0	0	0	0	0	8.6
V2	7.7	73	19.2	0	0	0	0	0
V3	0	3.8	96.1	0	0	0	0	0
V4	0	0	14.5	84	1.4	0	0	0
V5	0	0	0	2.1	97.2	0.7	0	0
V6	0	0	0	0	0	78.2	21.8	0
V7	0	0	0	0	0	0	94.9	5.1
V8	6.4	0	0	0	0	0	16	77.7

(b) Scenario 2

	V1	V2	V3	V4	V5	V6	V7	V8
V1	85	0	0	0	0	0	0	14.9
V2	16.8	62	21.2	0	0	0	0	0
V3	0	0	98.9	1.1	0	0	0	0
V4	0	0	1	78.8	20.2	0	0	0
V5	0	0	0	6.3	88.9	4.8	0	0
V6	0	0	0	0	16.6	75.7	3.9	3.9
V7	0	0	0	0	0	3.2	96.8	0
V8	3.8	0	0	0	0	0	15.2	80.9

(c) Scenario 3

Fig. 17 Viewpoint confusion matrices for Scenarios 1–3 using HOG features.

figure 8 well. In Fig. 14(e), results for CNN approximately reflect the shape of the path, but both types of errors e_{pose} and $e_{\text{viewpoint}}$ are significantly higher than those for HOG.

– The confusion matrix in Fig. 17(c) shows most of the viewpoint confusion was confined to neighbors of the true viewpoints. The worst confusion rate was associated with viewpoint V_2, whereas the highest classification accuracy was recorded for V_3. Generally, self-occlusions and loss of limb details are comparatively mild in V_1, V_3, V_5 and V_7, and hence they had the lowest confusion rates. Nevertheless, confusion rates depended largely on individual body dynamics.

4.3 Experiments with perspective distortion

This group of experiments was conducted using HOG features to analyze the effect of perspective distortion in detail. These experiments were extensions of the Scenario 2 experiments discussed in the previous subsection. In addition to 10m, the UAV was flown at heights of 20m, 30m and 40m (see Fig. 18). The lowest height of 10m caused negligible perspective distortion, but at $h = 40\text{m}$ ($\phi = 53.1°$), the video suffers from severe perspective distortion. The main observations are:

– In terms of pose estimation accuracy, perspective correction helped the dynamic classifier, but not C_{64}, which was significantly worse than the dynamic classifier.
– In terms of viewpoint estimation accuracy, perspective correction helps the dynamic classifier much more than it helped C_{64}.
– The advantage of perspective correction was more pronounced on more distorted videos.
– The advantage of perspective correction was more pronounced for the dynamic classifier than C_{64}.

Table 4 once again confirms the advantage of the dynamic classifier over C_{64}, which does not take into account the ordinal relationship between poses. The advantage was more pronounced for more distorted videos, provided perspective correction was applied.

5 Discussion

Our discussion pertains to the dynamic classifier, HOG features, CNN features, perspective correction, limitations of the approach and considerations for practical implementation.

Dynamic classifier. A drawback of the dynamic classifier is its dependence on accurate initial estimation. The solution given here is to use a multiclass classifier for the initialization, namely C_{64}, that recognizes all pose-viewpoint pairs. However, like all classifiers, C_{64} sometimes makes mistakes, throwing the $C_4(\cdot,\cdot)$ classifiers off-course. A potential improvement is to re-initialize the dynamic classifier (see Algorithm. 1) periodically.

HOG features. HOG features are traditionally considered to be handcrafted features, and in some domains, they have been replaced by CNN features. HOG cells in the literature do not capture additional information compared to CNN and they are significantly different features. HOG features are based on the weighted gradients in a HOG cell which represents the orientation of the edge lines. HOG are low-level features while CNN are high-level features with the ability to adapt to the task at hand during training.

However, given the robustness achieved in these multiclass classification experiments, HOG features outperform CNN features in pose and trajectory estimation. In our experiment, we extracted HOG features from a silhouette and CNN features from a color image. Our observation for the overall robustness of HOG features is that it is dependent on silhouettes and hence significantly on edges. However, segmentation of aerial images (for HOG) is very challenging due to the varying reso-

Table 4 Estimation errors of C_{64} and the dynamic classifier for perspective-distorted and perspective-corrected videos. Here, "PD" and "PC" refer to "perspective distorted" and "perspective corrected" respectively.

Scenario 2		#frames	$e_{\text{pose, with TE}}$		$e_{\text{viewpoint, with TE}}$	
			C_{64}	Dynamic classifier	C_{64}	Dynamic classifier
$h = 10$m	No distortion	787	**30**%	**23.5**%	**11.9**%	**13**%
$h = 20$m	PD	784	**37.2**%	**22.1**%	**18.9**%	**17.2**%
	PC		49.9%	39.9%	20.9%	20.4%
$h = 30$m	PD	810	57.9%	56.7%	28.9%	44.8%
	PC		**42.5**%	**40.6**%	**25.7**%	**37.2**%
$h = 40$m	PD	817	68.4%	74.4%	38%	42.6%
	PC		**53.5**%	**37.3**%	**30.5**%	**24.8**%

lution and background, and can benefit from the latest advances in semantic segmentation.

CNN features. The accuracy of CNN-based feature extraction depends on many factors such as the neural network model, nature of the original training dataset and complexity of the test image. The followings are the possible reasons why we achieved a lower accuracy for CNN compared to HOG:

– In the HOG approach, all the images are silhouettes and the features are formed from the edge details. In contrast, the CNN is sensitive to high-level features such as texture, background, face and gender, in addition to edges, and this can cause overfitting. An overfitting model learns the noise and random details in training data in addition to the targeted details. A similar observation of HOG features outperforming CNN features in classification due to overfitting by the latter has been reported in [77]. Techniques such as dropout [78] and DeCov regularizer [79] have been proposed to reduce overfitting and increase generalization. However, we did not apply these techniques, and the scope of our finding is limited to standard transfer learning.
– Transfer learning is a successful approach for many computer vision-related problems [42], but it has some constraints from the base network when copying the first n layers of the base network to the first n layers of the target network (left frozen feature layers). As a result, the feature layers do not change during the training of the new task. A possible alternative is to use deep transfer learning (DTL) which offers more flexibility when extracting high-level features [41]. DTL can perform layer-by-layer feature transference to solve a target problem in either a supervised or unsupervised setting [80].
– In a CNN, the features detected by earlier layers include low-level image details such as edges and colors. However, in the later layers the features progressively become more specific to the object categories of the original dataset.

– We used the original weights of AlexNet because our new dataset is very small compared to the original pre-trained dataset. This standard practice of not changing weights for a small dataset helps to reduce overfitting [41].
– In many computer vision applications, CNN features outperform low-level features when the neural network has been trained with a sufficiently large, application-specific dataset [81]. On the other hand, HOG does not need such a large dataset to achieve high accuracy.

Considering the factors above, it is not surprising CNN was outperformed by HOG in our experiments.

Left-right ambiguity. To illustrate how our algorithm handles left-right ambiguity (i.e., confusion between front and back views), we present an example in Fig. 19. Here, we consider a subject turning left from a side view to the back view. Once the subject turns $90°$ to the left, his shape should be identifiable as the back view rather than the front view. As demonstrated in the figure, the back view cannot be confused with the front view, although they are similar in shapes, because the 4 classes are confined to three adjacent viewpoints related to the left turn.

Perspective correction. The results presented in Section 4.3 confirm the intuition that perspective correction is imperative for severely perspective-distorted videos. Our solution has problems with purely frontal or rear views, because frontal and rear silhouettes do not provide sufficient details for differentiating pose. A potential solution is provided by the mobility of the aerial platform itself. The UAV can be programmed to seek a good elevation angle and azimuth angle, before it starts analyzing the human subject's action. This will require control algorithms and machine intelligence that go beyond the scope of this work.

Limitations of the approach. In this study, we tried to validate the suitability of dynamic classifiers for perspective distorted image sequences. We limited our work to gait estimation. However, the dynamic classi-

Reconstructed poses using C_4 have a temporal relationship, and most of the time an accurate viewpoint.

Reconstructed poses using C_{64} do not have a temporal relationship and the pose order is incorrect. However, most of the time their estimated viewpoint is correct.

Classifier C_4

Classifier C_{64}

(a) No perspective distortion, $h=10m$

(b) No perspective distortion, $h=10m$

h=10m

(c) Perspective distorted, $h=20m$

(d) Perspective corrected, $h=20m$

h=20m

(e) Perspective distorted, $h=30m$

(f) Perspective corrected, $h=30m$

h=30m

(g) Perspective distorted, $h=40m$
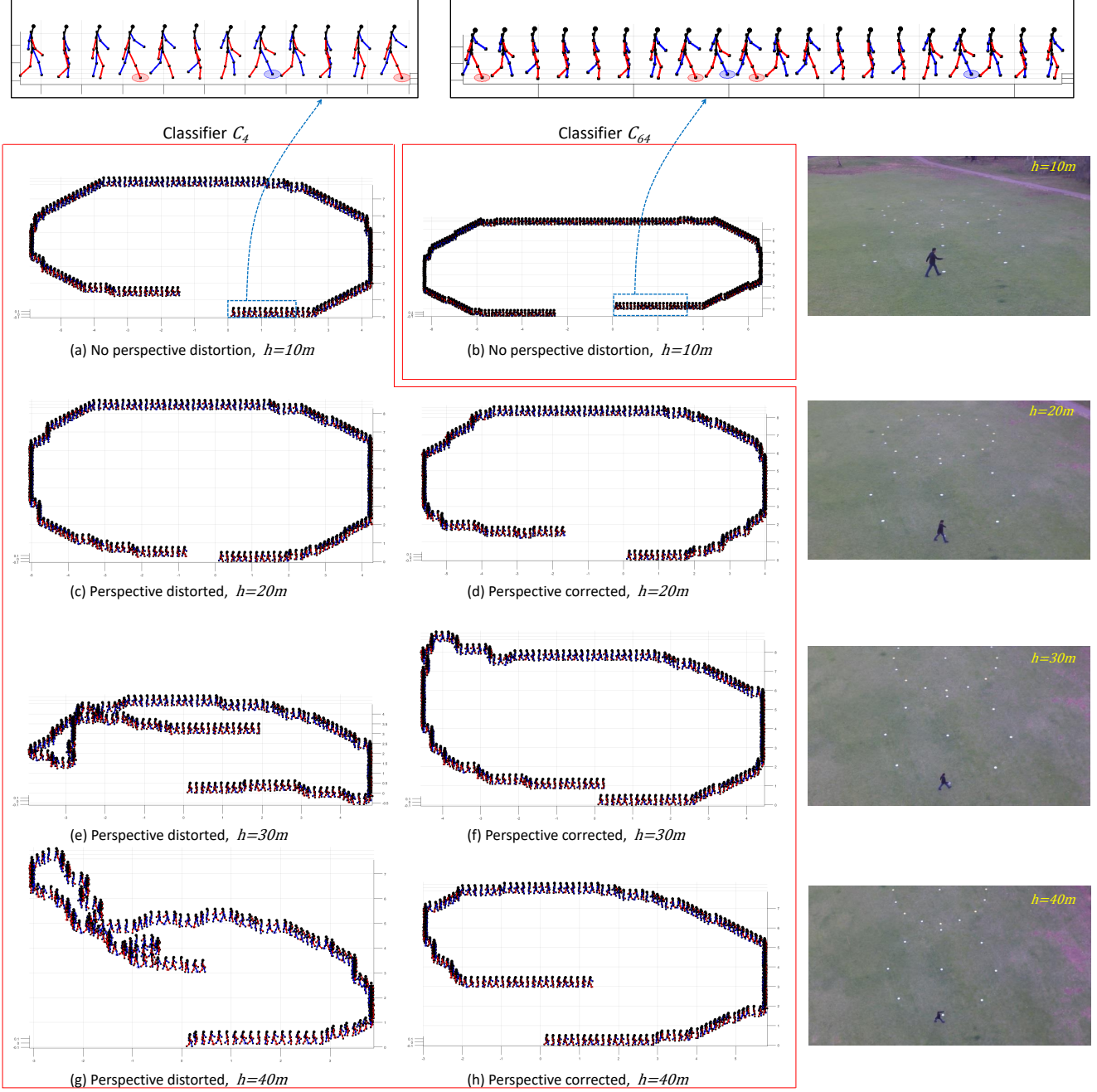
(h) Perspective corrected, $h=40m$

h=40m

Fig. 18 Reconstructed poses and their trajectories correspond to four different heights are shown (using HOG features). The zoomed sections from graph (a) and (b) show the 3-D reconstruction of poses for the first 50 frames using classifiers C_4 and C_{64} respectively. Graph (b) does not give reliable pose or trajectory estimation due to the independently estimated poses of C_{64}. However, its viewpoint estimation is approximately equal to C_4. That is the reason it gives a similar trajectory to C_4. Graphs (c) to (h) present the effect of perspective distortion compensation on trajectory estimation for three different heights using the C_4 classifier. This trajectory estimation comparison can not be done with C_{64} because it does not provide an ordinal relationship between poses. An original image captured at the respective height is included at the end of each row.

fiers can be extended to estimate complex human poses. Another limitation is our system cannot handle complex gait sequences like sharp turns, twists and walking

backwards. These are possible extensions to the current system, and can be addressed in future work. We used the standard transfer learning approach with Alexnet.
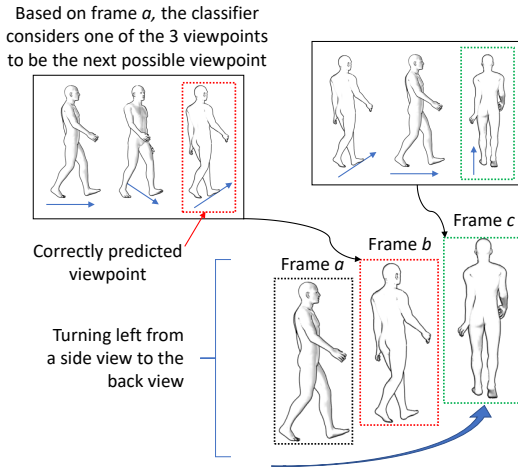
Fig. 19 An example of a subject turning left from a side view. In the classification stage, the first prediction is selected from the current and adjacent two viewpoints, each being 45° different from the current viewpoint. The prediction comes from 4 classes (current pose and viewpoint, and the next pose with 3 possible viewpoints). For a left turn, the most likely first prediction is the one which is 45° left to the current viewpoint. Again, in the next correct prediction, the prediction comes from 4 classes which includes the back view.

However, an application-specific deep transfer learning framework can offer more flexibility to fine tune the neural network model. Finally, our training dataset is relatively small. The accuracy and the robustness of the classifiers can be further improved by adding more diverse images to the training data.

Practical implementation. The original motivation for this work was to make UAVs intelligent enough to recognize human activities, so the question about whether the proposed solution can run on an embedded platform is relevant. The most computationally intensive components of the proposed solution include homography, human detection, HOG feature extraction and SVM classification. The most computationally intensive is SVM, but even this can be implemented on resource-constrained devices [82]. Further efficiency is ensured by the fact that a single 4-class classifier needs to run after initialization (recall Algorithm. 1). In conclusion, all the algorithmic components are practical for an embedded platform. Note that 3-D reconstruction of the estimated poses and trajectory is meant for visualizations, not embedded applications.

6 Conclusion and future work

As a first step toward solving the problem of estimating human pose and trajectory in monocular videos from an aerial platform, the paper presents a solution

that consists of perspective correction by homography, HOG/CNN feature extraction and dynamic classifier selection. The dynamic classifier is the defining feature of our solution, consisting of a 64-class classifier (namely C_{64}) and 64 4-class classifiers. The dynamic classifier works in conjunction with (i) a state transition model for the pose and viewpoint; and (ii) an SVM-based ECOC framework, which reduces multiclass classification to a set of efficiently solvable binary classification sub-problems (see Sects. 3.3–3.4). Trajectory estimation is for the estimation of the shape of the path traversed by the human subject, and is dependent on viewpoint estimation (see Section 3.5).

Experiments have been conducted with the CMU MoBo and HumanEva2 datasets and our own UAV-captured datasets, using e_{pose} and $e_{viewpoint}$ as defined in Equations. (6)–(9) as performance measures. The performance measures were calculated using two alternative feature sets (HOG and CNN), and the accuracies were compared before and after removing the transitional errors. Results show that

- The dynamic classifier outperforms C_{64}.
- Classification errors in the confusion matrix are evidently confined to neighbors of the true viewpoints. This property of the dynamic classifier enables fast recovery from incorrect estimations.
- HOG features, compared to CNN features, facilitate more accurate estimation.
- The more perspective-distorted a video is, the more necessary perspective correction is for reducing the estimation errors of the dynamic classifier.
- The proposed solution works well with both indoor and outdoor videos, and both ground videos and perspectively distorted aerial videos.
- The estimated trajectories approximate the actual trajectories well.

The solution proposed in this article is limited to estimating walking gaits. Our immediate plan is to extend the current work to the recognition of gestures performed during either walking or standing. Replacing the current HOG descriptors with Yang et al.'s flexible mixtures-of-parts model [83] should provide a promising start.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Informed Consent The data collection was conducted under the approval of University of South Australia's Human Research Ethics Committee (protocol no. 0000035185).

References

1. Orrite C, Herrero JE. Shape matching of partially occluded curves invariant under projective transformation. Computer Vision and Image Understanding. 2004;93(1):34 – 64.
2. Richter-Gebert J. Perspectives on projective geometry: a guided tour through real and complex geometry. Springer Science & Business Media; 2011.
3. Rogez G, Orrite C, Guerrero JJ, Torr PHS. Exploiting projective geometry for view-invariant monocular human motion analysis in man-made environments. Computer Vision and Image Understanding. 2014;120:126 – 140.
4. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1; 2005. p. 886–893 vol. 1.
5. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278–2324.
6. Woods K, Kegelmeyer WP, Bowyer K. Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997 Apr;19(4):405–410.
7. Ko AHR, Sabourin R, Britto AS, Jr. From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition. 2008;41(5):1718 – 1731.
8. Agarwal A, Triggs B. Recovering 3D human pose from monocular images. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2006 Jan;28(1):44–58.
9. Kuncheva LI, Bezdek JC, Duin RPW. Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition. 2001;34(2):299 – 314.
10. Tulyakov S, Jaeger S, Govindaraju V, Doermann D. In: Marinai S, Fujisawa H, editors. Review of Classifier Combination Methods. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. p. 361–386.
11. Perera AG, Law YW, Al-Naji A, Chahl J. Human Motion Analysis from UAV Video. International Journal of Intelligent Unmanned Systems. 2018;Accepted on 5 April 2018, in press.
12. Wang J, She M, Nahavandi S, Kouzani A. A Review of Vision-Based Gait Recognition Methods for Human Identification. In: 2010 International Conference on Digital Image Computing: Techniques and Applications; 2010. p. 320–327.
13. Hartley R, Zisserman A. Multiple view geometry in computer vision. Cambridge university press; 2003.
14. Rogez G, Guerrero JJ, Martínez J, Orrite-Urunuela C. Viewpoint Independent Human Motion Analysis in Man-made Environments. In: BMVC. vol. 6; 2006. p. 659.
15. Kuncheva LI. Switching between selection and fusion in combining classifiers: an experiment. IEEE Trans Syst, Man, Cybern B. 2002 April;32(2):146–156.
16. Xue Z, Ming D, Song W, Wan B, Jin S. Infrared gait recognition based on wavelet transform and support vector machine. Pattern Recognition. 2010;43(8):2904 – 2910.
17. Collins RT, Gross R, Shi J. Silhouette-based human identification from body shape and gait. In: Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition; 2002. p. 366–371.
18. Sarkar S, Phillips PJ, Liu Z, Vega IR, Grother P, Bowyer KW. The humanID gait challenge problem: data sets, performance, and analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005 Feb;27(2):162–177.
19. Veeraraghavan A, Roy-Chowdhury AK, Chellappa R. Matching shape sequences in video with applications in human movement analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005 December;27(12):1896–1909.
20. Zeng W, Wang C, Li Y. Model-Based Human Gait Recognition Via Deterministic Learning. Cognitive Computation. 2014 Jun;6(2):218–229.
21. Boulgouris NV, Hatzinakos D, Plataniotis KN. Gait recognition: a challenging signal processing technology for biometric identification. IEEE Signal Processing Magazine. 2005 November;22(6):78–90.
22. Sheikh Y, Sheikh M, Shah M. Exploring the space of a human action. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. vol. 1; 2005. p. 144–149 Vol. 1.
23. Rao C, Yilmaz A, Shah M. View-Invariant Representation and Recognition of Actions. International Journal of Computer Vision. 2002;50(2):203–226.
24. Rapantzikos K, Avrithis Y, Kollias S. Spatiotemporal Features for Action Recognition and Salient Event Detection. Cognitive Computation. 2011 Mar;3(1):167–184.
25. Chen SB, Xin Y, Luo B. Action-Based Pedestrian Identification via Hierarchical Matching Pursuit and Order Preserving Sparse Coding. Cognitive Computation. 2016 Oct;8(5):797–805.
26. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing. 2017;234:11 – 26.
27. Wei SE, Ramakrishna V, Kanade T, Sheikh Y. Convolutional Pose Machines. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 4724–4732.
28. Newell A, Yang K, Deng J. Stacked Hourglass Networks for Human Pose Estimation. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016. Cham: Springer International Publishing; 2016. p. 483–499.
29. Rogez G, Weinzaepfel P, Schmid C. LCR-Net: Localization-Classification-Regression for Human Pose. In: CVPR 2017 - IEEE Conference on Computer Vision & Pattern Recognition. Honolulu, United States; 2017. Available from: https://hal.inria.fr/hal-01505085.
30. Pishchulin L, Insafutdinov E, Tang S, Andres B, Andriluka M, Gehler PV, et al. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. .
31. Shah R, Romijnders R. Applying Deep Learning to Basketball Trajectories. CoRR. 2016;abs/1608.03793. Available from: http://arxiv.org/abs/1608.03793.
32. Yi S, Li H, Wang X. Pedestrian Behavior Understanding and Prediction with Deep Neural Networks. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016. Cham: Springer International Publishing; 2016. p. 263–279.
33. Fernando T, Denman S, Sridharan S, Fookes C. Soft + Hardwired Attention: An LSTM Framework for Human Trajectory Prediction and Abnormal Event Detection. CoRR. 2017;abs/1702.05552. Available from: http://arxiv.org/abs/1702.05552.

34. Labbaci H, Medjahed B, Aklouf Y. A Deep Learning Approach for Long Term QoS-Compliant Service Composition. In: Maximilien M, Vallecillo A, Wang J, Oriol M, editors. Service-Oriented Computing. Cham: Springer International Publishing; 2017. p. 287–294.

35. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, et al. Real-time human pose recognition in parts from single depth images. In: CVPR 2011; 2011. p. 1297–1304.

36. Zhao S, Chen L, Yao H, Zhang Y, Sun X. Strategy for dynamic 3D depth data matching towards robust action retrieval. Neurocomputing. 2015;151:533 – 543. Available from: http://www.sciencedirect.com/science/article/pii/S0925231214013940.

37. Sigal L, Black MJ. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2; 2006. p. 2041–2048.

38. Huang Y, Bogo F, Lassner C, Kanazawa A, Gehler PV, Romero J, et al. Towards Accurate Marker-less Human Shape and Pose Estimation over Time. In: International Conference on 3D Vision (3DV); 2017. .

39. Li Y, Sun Z. Generative tracking of 3D human motion in latent space by sequential clonal selection algorithm. Multimedia Tools and Applications. 2014 Mar;69(1):79–109.

40. Lan X, Huttenlocher DP. Beyond trees: common-factor models for 2D human pose recovery. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 1; 2005. p. 470–477 Vol. 1.

41. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in Neural Information Processing Systems 27. Curran Associates, Inc.; 2014. p. 3320–3328.

42. Chaturvedi I, Ong YS, Arumugam RV. Deep transfer learning for classification of time-delayed Gaussian networks. Signal Processing. 2015;110:250 – 262. Machine learning and signal processing for human pose recovery and behavior analysis.

43. Martín-Félez R, Xiang T. Gait Recognition by Ranking. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C, editors. Computer Vision – ECCV 2012. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. p. 328–341.

44. Farrajota M, Rodrigues JMF, du Buf JMH. A Deep Neural Network Video Framework for Monitoring Elderly Persons. In: Antona M, Stephanidis C, editors. Universal Access in Human-Computer Interaction. Interaction Techniques and Environments. Cham: Springer International Publishing; 2016. p. 370–381.

45. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–1105.

46. Rahmani H, Mian A, Shah M. Learning a Deep Model for Human Action Recognition from Novel Viewpoints. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;PP(99):1–1.

47. Farhadi A, Tabrizi MK. Learning to Recognize Activities from the Wrong View Point. In: Forsyth D, Torr P, Zisserman A, editors. Computer Vision – ECCV 2008. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. p. 154–166.

48. Andriluka M, Schnitzspan P, Meyer J, Kohlbrecher S, Petersen K, von Stryk O, et al. Vision based victim de-

tection from unmanned aerial vehicles. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on; 2010. p. 1740–1747.

49. Naseer T, Sturm J, Cremers D. FollowMe: Person following and gesture recognition with a quadrocopter. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2013. p. 624–630.

50. Lim H, Sinha SN. Monocular Localization of a moving person onboard a Quadrotor MAV. In: 2015 IEEE International Conference on Robotics and Automation (ICRA); 2015. p. 2182–2189.

51. Aguilar WG, Luna MA, Moya JF, Abad V, Parra H, Ruiz H. Pedestrian Detection for UAVs Using Cascade Classifiers with Meanshift. In: 2017 IEEE 11th International Conference on Semantic Computing (ICSC); 2017. p. 509–514.

52. Lao W, Han J, n De With PH. Automatic video-based human motion analyzer for consumer surveillance system. IEEE Transactions on Consumer Electronics. 2009 May;55(2):591–598.

53. Rudol P, Doherty P. Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery. In: Aerospace Conference, 2008 IEEE; 2008. p. 1–8.

54. Al-Naji A, Perera AG, Chahl J. Remote monitoring of cardiorespiratory signals from a hovering unmanned aerial vehicle. BioMedical Engineering OnLine. 2017 Aug;16(1):101.

55. Oreifej O, Mehran R, Shah M. Human identity recognition in aerial images. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on; 2010. p. 709–716.

56. Yeh MC, Chiu HK, Wang JS. Fast medium-scale multiperson identification in aerial videos. Multimedia Tools and Applications. 2016;75(23):16117–16133.

57. Monajjemi M, Bruce J, Sadat SA, Wawerla J, Vaughan R. UAV, do you see me? Establishing mutual attention between an uninstrumented human and an outdoor UAV in flight. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2015. p. 3614–3620.

58. Minaeian S, Liu J, Son YJ. Vision-Based Target Detection and Localization via a Team of Cooperative UAV and UGVs. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2016 July;46(7):1005–1016.

59. Smith KE, Kahanpää L, Kekäläinen P, Treves W, Chardin M. An invitation to algebraic geometry. The Mathematical Intelligencer. 2004;26(4):71–72.

60. Vondrick C, Patterson D, Ramanan D. Efficiently Scaling up Crowdsourced Video Annotation. International Journal of Computer Vision. 2013 Jan;101(1):184–204.

61. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 248–255.

62. Whittle MW. Gait analysis: an introduction. Butterworth-Heinemann; 2014.

63. Rosales R, Sclaroff S. Combining Generative and Discriminative Models in a Framework for Articulated Pose Estimation. International Journal of Computer Vision. 2006;67(3):251–276.

64. Rogez G, Rihan J, Orrite-Uruñuela C, Torr PHS. Fast Human Pose Detection Using Randomized Hierarchical Cascades of Rejectors. International Journal of Computer Vision. 2012 Aug;99(1):25–52.

65. Sigal L, Balan AO, Black MJ. HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated HumanMotion. International Journal of Computer Vision. 2009;87(1):4.

66. Gross R, Shi J. The CMU Motion of Body (MoBo) Database. Pittsburgh, PA: Robotics Institute; 2001. CMU-RI-TR-01-18.

67. Garcia-Pedrajas N, Ortiz-Boyer D. Improving multiclass pattern recognition by the combination of two strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006 June;28(6):1001–1006.

68. Dietterich TG, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of artificial intelligence research. 1995;2:263–286.

69. Lan X, Huttenlocher DP. A unified spatio-temporal articulated model for tracking. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004). vol. 1; 2004. p. I–722–I–729 Vol.1.

70. Fürnkranz J. Round robin classification. Journal of Machine Learning Research. 2002;2(Mar):721–747.

71. Masulli F, Valentini G. Effectiveness of error correctingoutput coding methods in ensemble and monolithic learningmachines. Formal Pattern Analysis &Applications. 2004;6(4):285–300.

72. Masulli F, Valentini G. In: Effectiveness of Error Correcting Output Codes in Multiclass Learning Problems. Berlin, Heidelberg: Springer Berlin Heidelberg; 2000. p. 107–116.

73. Ghani R. Using error-correcting codes for text classification. In: ICML; 2000. p. 303–310.

74. Allwein EL, Schapire RE, Singer Y. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of machine learning research. 2000;1(Dec):113–141.

75. Hastie T, Tibshirani R. Classification by pairwise coupling. Ann Statist. 1998 04;26(2):451–471.

76. Rogez G, Orrite-Uruuela C, del Rincn JM. A spatio-temporal 2D-models framework for human pose recovery in monocular sequences. Pattern Recognition. 2008;41(9):2926 – 2944.

77. Şentaş A, Tashiev İ, Küçükayvaz F, Kul S, Eken S, Sayar A, et al. Performance Evaluation of Support Vector Machine and Convolutional Neural Network Algorithms in Real-Time Vehicle Type Classification. In: Barolli L, Xhafa F, Javaid N, Spaho E, Kolici V, editors. Advances in Internet, Data & Web Technologies. Cham: Springer International Publishing; 2018. p. 934–943.

78. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014;15(1):1929–1958.

79. Cogswell M, Ahmed F, Girshick RB, Zitnick L, Batra D. Reducing Overfitting in Deep Networks by Decorrelating Representations. CoRR. 2015;abs/1511.06068. Available from: http://arxiv.org/abs/1511.06068.

80. Kandaswamy C, Monteiro JC, Silva LM, Cardoso JS. Multi-source deep transfer learning for cross-sensor biometrics. Neural Computing and Applications. 2017 Sep;28(9):2461–2475.

81. Jain A, Tompson J, LeCun Y, Bregler C. MoDeep: A Deep Learning Framework Using Motion Features for Human Pose Estimation. In: Cremers D, Reid I, Saito H, Yang MH, editors. Computer Vision – ACCV 2014. Cham: Springer International Publishing; 2015. p. 302–315.

82. Anguita D, Ghio A, Pischiutta S, Ridella S. A Hardware-friendly Support Vector Machine for Embedded Automotive Applications. In: 2007 International Joint Conference on Neural Networks; 2007. p. 1360–1364.

83. Yang Y, Ramanan D. Articulated pose estimation with flexible mixtures-of-parts. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE; 2011. p. 1385–1392.