Articulated Motion Discovery using Pairs of Trajectories

Luca Del Pero[1] ldelper@inf.ed.ac.uk
Susanna Ricco[2] ricco@google.com
Rahul Sukthankar[2] sukthankar@google.com
Vittorio Ferrari[1] ferrari@inf.ed.ac.uk
[1]University of Edinburgh [2]Google Research

Abstract

We propose an unsupervised approach for discovering characteristic motion patterns in videos of highly articulated objects performing natural, unscripted behaviors, such as tigers in the wild. We discover consistent patterns in a bottom-up manner by analyzing the relative displacements of large numbers of ordered trajectory pairs through time, such that each trajectory is attached to a different moving part on the object. The pairs of trajectories descriptor relies entirely on motion and is more discriminative than state-of-the-art features that employ single trajectories. Our method generates temporal video intervals, each automatically trimmed to one instance of the discovered behavior, and clusters them by type (e.g., running, turning head, drinking water). We present experiments on two datasets: dogs from YouTube-Objects and a new dataset of National Geographic tiger videos. Results confirm that our proposed descriptor outperforms existing appearance- and trajectory-based descriptors (e.g., HOG and DTFs) on both datasets and enables us to segment unconstrained animal video into intervals containing single behaviors.

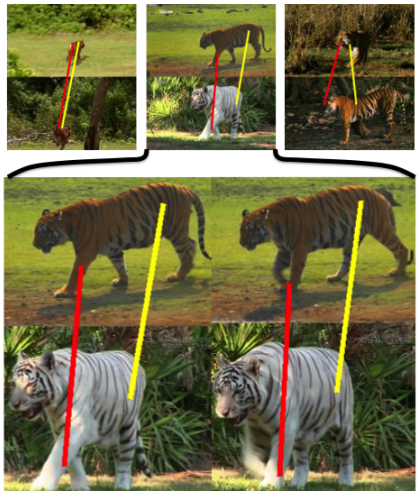

Figure 1. Examples of articulated motion pattern clusters discovered using pairs of trajectories (PoTs). These clusters capture tigers running, walking, and turning their heads, respectively. Inset shows detail of one PoT within the walking cluster. Yellow lines connect the first trajectories (on the tigers' body); red lines connect the second (on moving extremities).

1. Introduction

Internet videos provide a wealth of data that could be used to learn the appearance or expected behaviors of many object classes. However, traditional supervised learning techniques used on still images [1,2,4] do not easily transfer due to the prohibitive cost of generating ground-truth annotations in videos. In order to realize the full potential of this vast resource, we must instead rely on methods that require as little human supervision as possible.

We propose a bottom-up method for discovering the characteristic motion patterns of an articulated object class *in the wild*. Unlike the majority of action recognition datasets, in which human actors perform scripted actions [6, 31, 33, 44], and/or clips are trimmed to contain a single action [16, 34], our videos are unstructured, such as animals performing unscripted behaviors. The only assumption we make is that each video contain at least one instance of the object class. We leverage that the object is engaged in some (unknown) behaviors, and that such behaviors exhibit observable consistency, which we term characteristic *motion patterns*.

Our method does not require knowledge of the number or types of behaviors, nor that instances of different behaviors be temporally segmented within a video. The output of our method is a set of video intervals, clustered according to the observed characteristic motion patterns. Each interval contains one temporally segmented instance of the pattern. Fig. 1 shows some behaviors automatically discovered in tiger videos, such as walking, turning head, and running.

We identify consistency between observed motion patterns by analyzing the relative displacement of large numbers of ordered trajectory pairs (PoTs). The first trajectory in the pair defines a reference frame in which the motion of the second trajectory is measured. We preferentially sample trajectory pairs across joints, resulting in features particularly well-suited to representing fine-grained behaviors

of complex, articulated objects. This has greater discriminative power than state-of-the-art features defined using single trajectories in isolation [37, 38].

Although we often refer to PoTs using semantic labels for the location of their component trajectories (eye, shoulder, hip, etc.), these are used only for convenience. PoTs do not require semantic understanding or any part-based or skeletal model of the object, nor are they specific to an object class. Furthermore, the collection of PoTs is more expressive than a simple star-like model in which the motion of point trajectories are measured relative to the center of mass of the object. For example, we find the "walking" cluster (Fig. 5) based on PoTs formed by various combinations of head-paw (Fig. 2 III, a), hip-knee (c), knee-paw (b,d), or even paw-paw (e) trajectories.

In contrast to other popular descriptors [10, 37, 38], PoTs are appearance-free. They are defined solely by motion and so are robust to appearance variations within the object class. In cases where appearance proves beneficial for discriminating between behaviors of interest, it is easy to combine PoTs with standard appearance features.

In summary, our main contributions are: (1) a new feature based on ordered pairs of trajectories that captures the intricate motion of articulated objects (Sec. 3); (2) a method for unsupervised discovery of behaviors from unconstrained videos of an object class (Sec. 4); (3) a method for identifying periodic motion in video, which we use to segment videos into intervals containing single behaviors (Sec. 4.1); and (4) annotations for 80,000 frames from nature documentaries about tigers and 20,000 frames from YouTube videos of dogs (Sec. 5), available on our website [3].

2. Related work

Motion is a fundamental cue for many applications in video analysis and so has been widely studied, particularly within the context of action recognition [36, 41]. However, action recognition is traditionally formulated as a supervised classification problem [17, 34]. Work on unsupervised motion analysis has largely been restricted to the problem of dynamic scene analysis [7, 8, 18, 20, 40, 45]. These works typically consider a fixed scene observed at a distance from a static camera; the goal is to model the behavior of agents (typically pedestrians and vehicles) and to detect anomalous events. Features typically consist of optical flow at each pixel [7, 18, 40] or single trajectories corresponding to tracked objects [8, 45].

To our knowledge, only Yang *et al*. [43] considered the task of unsupervised motion pattern discovery, although from manually trimmed videos. Their method models human actions in terms of motion primitives discovered by clustering localized optical flow vectors, normalized with respect to the dominant translation of the object. In contrast, our pairwise features capture complex relationships between the motion of two different object parts. Furthermore, we describe motion at a more informative temporal scale by using multiframe trajectories instead of two-frame optical flow. We compare experimentally to [43] on the KTH dataset [33] in Sec. 5.2.

Although many approaches do not easily transfer from the supervised to the unsupervised domain, one major breakthrough from the action recognition literature that does is the concept of *dense trajectories*. The idea of generating trajectories for each object from large numbers of KLT interest points in order to model its articulation was simultaneously proposed by Matikainen *et al*. [21] and Messing *et al*. [23] for action recognition. These ideas were extended and refined in the work on tracklets [30] and dense trajectory features (DTFs) [37, 38]. DTFs currently provide state-of-the-art performance on video action recognition [12].

In contrast to our work, most trajectory-based methods treat each trajectory in isolation [21, 23, 30, 37, 38], with two notable exceptions [11, 24]. Jiang *et al*. [11] assign individual trajectories to a single codeword from a predefined codebook (as in DTF works [37, 38]). However, the codewords from a pair of trajectories are combined into a 'codeword pair' augmented by coarse information about the relative motion and average location of the two trajectories. Yet, this pairwise analysis is cursory: the selection of codewords is unchanged from the single-trajectory case, and the descriptor thus lacks the fine-grained information about the relative motion of the trajectories that our proposed PoTs provide. Narayan *et al*. [24] model Granger causality between trajectory codewords. Their global descriptor only captures pairwise statistics of codewords over a fixed-length temporal interval. In contrast, a PoT groups two trajectories into a single local feature, with a descriptor encoding their spatiotemporal arrangement. Hence, PoTs can be used to find point correspondences between videos (Fig. 5).

The few remaining methods that propose pairwise representations employ them in a very different context. Leordeanu *et al*. [19] learned object classes from still images by matching pairs of contour points from one image to pairs in another. Yang *et al*. [42] computed statistics between local feature pairs for food recognition in images. Matikainen *et al*. [22] used spatial and temporal features computed over pairs of sparse KLT trajectories to construct a two-level codebook for action classification. Dynamic-poselets [39] requires detailed manual annotations of human skeletal structure on training data to define a descriptor for pairs of connected joints. Raptis *et al*. [29] consider pairwise interactions between clusters of trajectories, but their method also requires detailed manual annotation for each action. None of these approaches is suitable for unsupervised articulated motion discovery.

A few recent works exploit video as a source of training data for object class detectors [26, 35]. They separate object

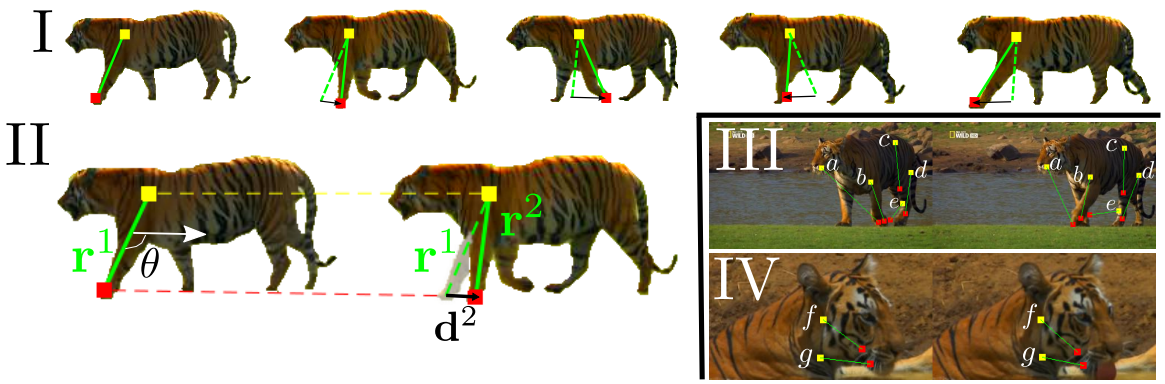

Figure 2. Modeling articulated motion with PoTs. Two trajectories in a PoT are ordered based on their deviation from the median velocity of the object: the anchor (yellow) deviates less than the swing (red). In I, the displacement of the swing relative to the anchor follows the swinging motion of the paw with respect to the shoulder. While both move forward as the tiger walks, the paw is actually moving backwards in a coordinate system centered at the shoulder. This back-and-forth motion is captured by the relative displacement vectors of the pair (in black) but missed when individual trajectories are used alone. The PoT descriptor is constructed from the angle θ and the black vectors $\mathbf{d^k}$, shown in II. The two trajectories in a PoT are selected such that they track object parts that move differently. A few selected PoTs are shown in III and IV. Legs move differently than the head (a), hip (c), knees (b,d), or other legs (e). In IV, the head rotates relative to the neck, resulting in different PoTs (f,g). Our method selects these PoTs without requiring prior knowledge of the object topology.

instances from their background based on motion, thus reducing the need for manual bounding-box annotation. However, their use of video stops at segmentation. They make no attempt at modeling articulated motion or finding common motion patterns across videos. Ramanan et al. [27] build a 2D part-based model of an animal from one video. The model is a pictorial structure based on a 2D kinematic chain of coarse rectangular segments. Their method operates strictly on individual videos and therefore cannot find motion patterns characteristic for a class. It is tested on just three simple videos containing only the animal from a single, unchanging viewpoint.

3. Pairs of Trajectories (PoTs)

We represent articulated object motion using a collection of *automatically selected* ordered pairs of trajectories (PoTs), tracked over n frames. Only two trajectories following parts of the object moving relatively to each other are selected as a PoT, as these are the pairs that move in a consistent and distinctive manner across different instances of a specific motion pattern. For example, the motion of a pair connecting a tiger's knee to its paw consistently recurs across videos of walking tigers (Figs. 1 and 5). By contrast, a pair connecting two points on the chest (a rather rigid structure) may be insufficiently distinctive, while one connecting the tip of the tail to the nose may lack consistency. Note also that a trajectory may simultaneously contribute to multiple PoTs (*e.g.*, a trajectory on the front paw may form pairs with trajectories from the shoulder, hip, and nose).

Fig. 2 (III-IV) shows a few examples of PoTs selected from two tiger videos. We define PoTs in Sec. 3.1, while we explain how to select PoTs from real videos in Sec. 3.2.

3.1. PoT definition

Anchors and swings. The first trajectory in each PoT (the *anchor*) defines a local coordinate frame, in which the motion of the second (*swing*) is measured. We select as anchor the trajectory whose velocity is closer to the median velocity of pixels detected to be part of the foreground (Sec. 3.2), aggregated over the length of the PoT (this approximates the median velocity of the whole object). This criterion generates a stable ordering, repeatable across the broad range of videos we examine. For example, the trajectories on the legs in Fig. 2 (I-II-III) are consistently chosen as swings while those on the torso are selected as anchors.

Displacement vectors. In each frame f_k, we compute the vector $\mathbf{r^k}$ from anchor to swing (green lines in Fig. 2). Starting from the second frame, a displacement vector $\mathbf{d^k}$ is computed by subtracting the vector $\mathbf{r^{k-1}}$ of the previous frame (dashed green) from the current $\mathbf{r^k}$ (solid green). $\mathbf{d^k}$ captures the motion of the swing relative to the anchor by canceling out the motion of the latter. Naively employing the green vectors $\mathbf{r^k}$ as raw features does not capture relative motion as effectively because the variation in $\mathbf{r^k}$ through time is dominated by the spatial arrangement of anchor and swing rather than by the change in relative position between frames. This can be intuitively appreciated by comparing the magnitudes of the green and black vectors in Fig. 2.

PoT descriptor. The PoT descriptor P consists of two parts: 1) the initial position of the swing relative to the anchor and 2) the sequence of normalized displacement vec-

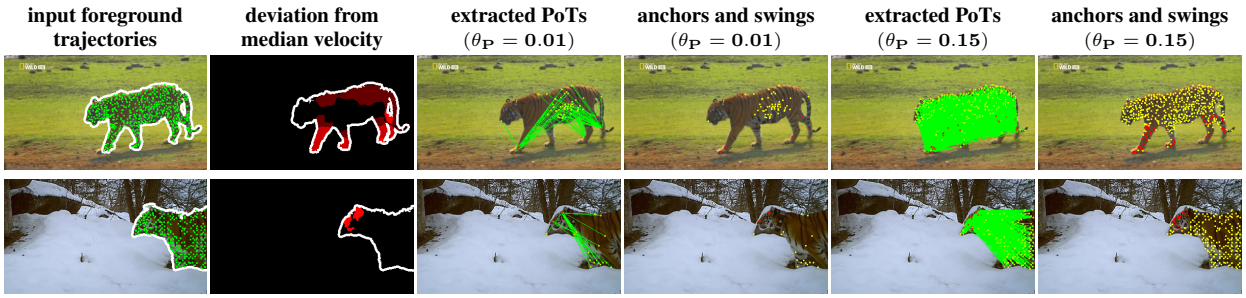

Figure 3. PoT selection on two different examples: a tiger walking (top) and one turning its head (bottom). We construct PoT candidates from the trajectories on the foreground mask (a), using all possible pairs. We prefer candidates where the anchor is closer to the median foreground velocity, denoted by dark areas in (b), while the swing follows a different motion (bright areas). We keep the highest $\theta_P\%$ ranking candidates according to this criterion. We show the selected PoTs for two different values of θ_P (c,e). Too strict a θ_P ignores many interesting PoTs, like those involving trajectories on the neck in the top row (c). We also show the trajectories used as anchors (yellow) and swings (red) without the lines connecting them (d,f). Imagine connecting any anchor with any swing: in most cases, the two follow different, independently moving parts of the object, which is the key requirement of a PoT. We use $\theta_P = 0.15$ in our experiments (e,f).

tors through time:

$$P = \left(\theta, \frac{\mathbf{d^2}}{D}, \ldots, \frac{\mathbf{d^n}}{D}\right), \tag{1}$$

where θ is the angle from anchor to swing in the first frame and the normalization factor is the total displacement $D = \sum_{k=2}^{n} ||\mathbf{d^k}||$. The DTF descriptor [37] employs a similar normalization. Note also that the first term in P records only the angle (and not the magnitude) between anchor and swing; this retains scale invariance and enables matching PoTs from objects of different size. The dimensionality of P is $2 \cdot (n-1)+1$; in all of our experiments, we set $n = 10$.

3.2. PoT selection

We explain here how to select PoTs from a set of input trajectories output by a dense point tracker [38]. We start with a summary of the process and give more details later.

First, we use a recent method for foreground segmentation [25] to remove trajectories on the background. Then, for each frame f we build the set $\mathcal{P}_f$ of PoTs starting at that frame. For computational efficiency, we directly set $\mathcal{P}_f = \emptyset$ for any frame unlikely to contain articulated motion. Otherwise, we form candidate PoTs from all pairs of foreground trajectories $\{t_i, t_j\}$ extending for at least n frames after f. Finally, we retain in $\mathcal{P}_f$ the candidates that are most likely to be on object parts moving relative to each other.

Foreground segmentation. State-of-the-art point trajectories already attempt to limit trajectories to foreground objects [38], but often fail on the wide range of videos we use. We instead use a recent method [25] for foreground segmentation in unconstrained video. The resulting *foreground mask* permits reliable detection of articulated objects even under significant motion and against unconstrained backgrounds. Our method is robust to errors in the foreground mask because they only affect a small fraction of the PoT collection (Sec. 5.3).

In addition to removing trajectories on the background, we also use this foreground mask to estimate the median velocity of the object, computed as the median optical flow displacement over all pixels in the mask.

Pruning frames without articulated motion. A frame is unlikely to contain articulated motion (hence PoTs) if the optical flow displacement of foreground pixels is uniform. This happens when the entire scene is static, or the object moves with respect to the camera but the motion is not articulated. We define $s(f) = \frac{1}{n}\sum_{i=f}^{f+n-1} \sigma_i$, where σ_i is the standard deviation in the optical flow displacement over the foreground pixels at frame i normalized by the mean, and n the length of the PoT. We set $\mathcal{P}_f = \emptyset$ for all frames where $s(f) < \theta_F$, pruning frames without promising candidate pairs. We set $\theta_F = 0.1$ using 16 cat videos in which we manually labeled frames without articulated motion. $\theta_F = 0.1$ achieves a precision of 0.95 and a recall of 0.75.

PoT candidates and selection. The candidate PoTs for an unpruned frame f are all ordered pairs of trajectories $\{t_i, t_j\}$ that exist in f and in the following $n-1$ frames and lie on the foreground mask. These trajectories are shown in Fig. 3(a). We score a candidate pair $\{t_i, t_j\}$ using

$$\mathrm{S}(\{t_i = a, t_j = s\}) = \sum_{k=f}^{f+n-1} ||v_s^k - v_m^k|| - ||v_a^k - v_m^k|| \ , \tag{2}$$

where v_m^k is the median velocity at frame k, and v_s^k and v_a^k the velocities of the swing and anchor, respectively. The first term favors pairs with a large deviation between swing and median velocity, while the second term favors pairs

where the velocity of the anchor is close to the median. As seen in Fig. 3, this generates a stable PoT ordering where anchors and swings fall on the core and extremities of the animal, respectively. However, note that the velocity of the anchors can vary; anchors along the tiger's back in the top row deviate significantly from the median velocity.

We rank all candidates using (2) and retain the top $\theta_P\%$ candidates as PoTs $\mathcal{P}_f$ for this frame. We found this approach to work quite well in practice. A few examples of the top ranking candidates are shown in Fig. 3. In practice, we use the PoTs shown in Fig. 3(e,f).

4. Motion pattern discovery

The input to our motion discovery system is a set of videos $\mathcal{V}$ containing objects of the same class, such as tigers. The desired output is a set of clusters $\mathcal{C} = (c_1, ..., c_k)$ corresponding to motion patterns. Each cluster should contain temporal intervals showing the same motion pattern (an interval is any subsequence of frames). For the "tiger" class, we would like a cluster with tigers walking, one with tigers turning their head, and so on. The videos we use (Sec 5.1) typically contain several instances of different motion patterns each. For our purposes, it is easier to cluster intervals that correspond to just one instance of a motion pattern, and ideally cover the whole duration of that instance. Hence, we first temporally partition videos into intervals corresponding to a single motion pattern (Sec. 4.1). Then we cluster these intervals to discover motion patterns (Sec. 4.2).

4.1. Temporal partitioning

We first partition videos into shots by thresholding color histogram differences in consecutive frames [15]. A shot will typically contain several different motion patterns. For example, a cat may walk for a while, then sit down and finally stretch. Here, we want to partition the shot into *single-pattern intervals*, *i.e.*, a "walking", a "sitting down" and a "stretching" interval. Unlike shots, boundaries between such intervals cannot be detected using simple color histogram differences. Instead we partition using two different motion cues: pauses and periodicity, which we discuss next.

Motion-based partitioning. We first note that the object often stays still for a brief moment between two different motion patterns. We detect such pauses as sequences of three or more frames without articulated object motion. However, some sequences lack pauses between different related behaviors (*e.g.*, a tiger walking begins to run). Thus, we also partition based on detected periodic motion.

Periodic motion detector. We use time-frequency analysis to detect periodic motion. We assume periodic motion patterns like walking, running, or licking generate peaks in the frequency domain (examples are available on our website [3]). Specifically, we model an input interval as a time sequence $s(t) = b^P_{f^t}$, where $b^P_{f^t}$ is the bag-of-words (BoW) of PoTs at frame f^t. We convert $s(t)$ to C one-dimensional sequences (one per codeword) and sum the FFTs of the individual sequences in the frequency domain. If the height of the highest peak is $\geq \theta_H$, we consider the interval as periodic. We ensure that the total energy in the frequency domain integrates to 1. Using the sum of the FFTs makes the approach more robust, since peaks arise only if several codewords recur with the same frequency.

Naively doing time-frequency analysis on an entire interval typically fails because it might contain both periodic and non-periodic motion (*e.g.*, a tiger walks for a while and then sits down). Hence, we consider all possible sub-intervals using a temporal sliding window and label the one with the highest peak as periodic, provided its height $\geq \theta_H$. The remaining segments are reprocessed to extract motion patterns with different periods (*e.g.*, walking versus running) until no significant peaks remain. For robustness, we only consider sub-intervals where the period is at least five frames and the frequency at least three (*i.e.*, the period repeats at least three times). We empirically set $\theta_H = 0.1$, which produces very few false positives.

4.2. Clustering intervals

Interval representation. We use k-means to form a codebook from a million PoT descriptors randomly sampled from all intervals. We run k-means eight times and choose the clustering with lowest energy to reduce the effects of random initialization [38]. We then represent an interval as a BoW histogram of the PoTs it contains (L1-normalized).

Hierarchical clustering. We cluster the intervals using hierarchical clustering with complete-linkage [13]. We found this to perform better than other clustering methods (*e.g.*, single-linkage, k-means) for all the descriptors tested. As an additional advantage, hierarchical clustering enables one to experiment with different numbers of clusters without re-running the algorithm.

Distance function. Hierarchical clustering requires computing the distance between pairs of input items. Given BoWs of PoTs b_u and b_v for intervals I_u and I_v, we use

$$\mathrm{d}(I_u, I_v) = -\exp\left(-(1 - \mathrm{HI}(b_u, b_v)) \right), \qquad (3)$$

where HI denotes histogram intersection. We found this to perform slightly better than the χ^2 distance for all descriptors tested. Note that this function can be also used on BoWs of descriptors other than PoTs. Additionally, it can be extended to handle different descriptors that use multiple feature channels, such as Improved DTFs [38], which we compare against in the experiments. In this case, the interval representation is a set of BoWs $(b^1_u, ..., b^C_u)$, one for each of the C channels. Following [38], we combine all

channels into a single distance function

$$\mathrm{d}(I_u, I_v) = -\exp\left(-\sum_{i=1}^{C} \frac{1-\mathrm{HI}(b_u^i, b_v^i)}{A_i}\right), \quad (4)$$

where A_i is the average value of $(1-\mathrm{HI})$ for channel i.

5. Experiments

In this section, we present our experimental results.

5.1. Evaluation protocol

Datasets. We experiment on two different datasets. First, we use a dataset of tiger videos collected from National Geographic documentaries. This dataset contains roughly two hours of high-resolution, professional footage divided into 500 shots, for a total of 80,000 frames. Throughout the experiments, we use various portions of this dataset:

- *Tiger_fg:* A set of 100 shots with accurate foreground masks [25], selected manually.
- *Tiger_val:* Another set of 100 shots where the segmentation algorithm works well with no overlap with Tiger_fg. We use Tiger_val to set the parameters of all the methods we test.
- *Tiger_all:* All the shots in the dataset.

Second, we use 100 shots of the dog class of the YouTube-Objects dataset [26], which mostly contains low-resolution footage filmed by amateurs.

Behavior labels. We annotated *each frame* in the dataset independently, choosing from the behavior labels listed in Table 1. When a frame shows multiple behaviors, we chose the one that happens at the larger scale (*e.g.*, we choose "walk" over "turn head" and "turn head" over "blink"). As animals move over time, a shot often contains more than one label. All the labels will be released on our website.

Evaluation criteria. We use two criteria commonly used for evaluating clustering methods: *purity* and *Adjusted Rand Index* (ARI) [28]. Purity is the number of items correctly clustered divided by the total number of items. An item is correctly clustered if its label coincides with the most frequent label in its cluster. While purity is easy to interpret, it only penalizes assigning two items with different labels to the same cluster. The ARI instead also penalizes putting two items with the same label in different clusters. Further, it is adjusted such that a random clustering will score close to 0. It is considered a better way to evaluate clustering methods by the statistics community [9, 32].

Baseline. We compare PoTs to the state-of-the-art Improved Dense Trajectory Features (IDTFs) [38]. IDTFs combine four different feature channels aligned with dense trajectories: Trajectory shape (TS), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and

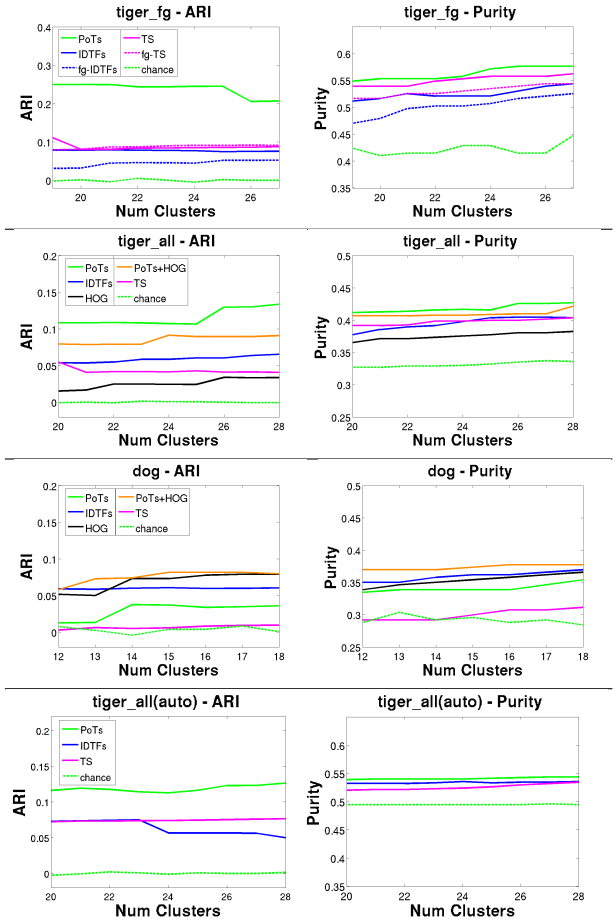

Figure 4. Results of clustering intervals using different descriptors, evaluated on Adjusted Rand Index (ARI) and purity (see text). PoTs result in better clusters than the full IDTFs [38] on tigers (top two rows). Restricting IDTFs to the foreground segmentation decreases the performance on tiger_fg, where we ensured the segmentation is accurate (top row). Adding appearance features (PoTs+HOG) is detrimental for tigers (second row), but improves performance on dogs (third row). IDTFs perform well for dogs, primarily due to the contribution of the HOG channel alone (compare the full descriptor, blue, with the HOG channel only, black, and trajectory shape (TS) channel only, magenta). For both tigers and dogs, PoTs+HOG performs better than IDTFs. PoTs also generate higher-quality clusters than the other methods when we cluster automatically partitioned intervals (bottom row).

Motion Boundary Histogram (MBH). TS is the channel most related to PoTs, as it encodes the displacement of an individual trajectory across consecutive frames. HOG is the only component based on appearance and not on motion. We also compare against a version of IDTFs where only trajectories on the foreground segmentation are used. We call this method fg-IDTFs. We use the same point tracker [38] to extract both IDTFs and PoTs. For PoTs, we do not remove trajectories that are static or are caused by the motion of the camera. Removing these trajectories improves the performances of IDTFs [38], but in our case they are useful as potential anchors.

Partitions	walk	turn head	sit down	tilt head	stand up	drag	wag tail	walk back	run	turn	jump	raise paw	open mouth	close mouth	blink	slide leg	drink	chew	lick	climb	roll	scratch	swim
whole shots	259	24	5	17	5	4	1	5	16	3	9	0	4	0	0	1	7	4	6	1	3	1	3
pause	272	76	11	30	9	4	2	6	16	2	11	2	11	3	10	2	7	5	7	1	5	1	3
pause+periods	**273**	**80**	**13**	**33**	**9**	**4**	**2**	**6**	**16**	**3**	**11**	**2**	**12**	**3**	**11**	**3**	**8**	**5**	**7**	**1**	**5**	**1**	**3**
ground truth	289	148	27	77	24	4	4	10	23	18	20	6	40	28	39	13	12	7	19	1	5	2	3

Partitions	walk	turn head	sit down	tilt head	stand up	walk back	run	turn	jump	open mouth	close mouth	blink	slide leg	lick	push skateboard
whole shots	25	4	0	1	0	0	10	0	4	0	0	0	0	0	13
pause	29	5	0	1	0	0	12	2	5	0	0	0	0	0	16
pause+periods	**29**	**9**	**0**	**3**	**0**	**1**	**13**	**5**	**5**	**0**	**0**	**0**	**0**	**0**	**16**
ground truth	39	25	1	12	1	2	20	14	8	2	1	2	1	1	19

Table 1. Number of intervals recovered per behavior on tigers (top) and dogs (bottom). Pause+periods consistently dominates others.

Calibration. We use Tiger_val to set the PoT selection threshold θ_P (Sec. 3.2) and the PoT codebook size K (sec. 4.2) using coarse grid search. As objective function, we used the ARI achieved by our method with the number of clusters equal to the true number of behaviors. The chosen parameters are $\theta_P = 0.15$ and $K = 800$. We tuned the IDTF codebook size similarly; the best size was 4000. Interestingly, this is the same value as chosen by Wang et al. [38] on completely different data.

5.2. Evaluating PoTs

We first evaluate PoTs in a simplified scenario where the correct single-pattern partitioning is given, *i.e.*, we partition shots using frames where the ground-truth label changes as boundaries. This allows us to evaluate the PoT representation separately from our method for automatic interval discovery (Sec. 4.1). We compare clustering using BoWs of PoTs to clustering using BoWs of IDTFs in Fig. 4. As the true number of clusters is usually not known a priori, each plot shows performance as a function of the number of clusters. The mid value on the horizontal axis corresponds to the true number of clusters (23 for tigers, 15 for dogs).

Evaluation on tigers. The clusters found using PoTs are better in both purity and ARI (Fig. 4). The gain over IDTFs is larger on Tiger_fg (top row), where PoTs benefit from the accurate estimate of the foreground. Here, PoTs also outperform fg-IDTFs. This shows that the power of our representation resides in the principled use of pairs, not just in exploiting the foreground segmentation to remove background trajectories. Results on Tiger_all (second row) show that PoTs can also cope with imperfect segmentation.

Consider now the individual IDTFs channels. HOG performs poorly and causes the complete IDTFs to perform worse than their TS channel alone, although both are inferior to PoTs. Similarly, adding the HOG channel to PoTs performs worse than pure motion PoTs but is still better than IDTFs. Appearance is in general not suitable for discovering fine-grained motion patterns. It is particularly misleading in a class like "tiger" where different instances have similar color and texture. The HOF and MBH channels of IDTF perform poorly on their own and are not shown here.

	whole shots	pauses	pauses+periods	ground truth
tiger # intervals	480	719	**885**	1026
tiger uniformity	0.78	0.85	**0.87**	1
dog # intervals	80	115	**219**	260
dog uniformity	0.72	0.80	**0.88**	1

Table 2. Interval uniformity for different partitioning methods. Pauses+periods consistently outperforms alternatives.

Evaluation on dogs. The complete IDTF descriptors perform better than PoTs on the dog dataset (Fig. 4, third row). However, the HOG channel is doing most of the work in this case. The dog shots come from only eight different videos, each showing one particular dog performing 1–2 behaviors in the same scene. Hence, HOG performs well by trivially clustering together intervals from the same video. If we equip PoTs with the HOG channel, they outperform the complete IDTFs. Similarly, when considering trajectory motion alone, PoTs outperform the IDTF TS channel. These experiments confirm that PoTs are a better representation for articulated objects than IDTF also on the dog data.

Comparison to motion primitives [43]. Last, we compare to the method of [43], which is based on motion primitives. We compare on the KTH dataset [33] in their setting. It contains 100 shots for each of six different human actions (*e.g.* walking, hand clapping). As before, we cluster all shots using the PoT representation: for the true number of clusters (6), we achieve 59% purity, compared to their 38% (Fig. 9 in [43]). For this experiment, we incorporated an R-CNN person detector [5] into the foreground segmentation algorithm [25] to better segment the actors.

5.3. Evaluating motion discovery

We now evaluate our method for partitioning into single-pattern intervals. Let the *interval uniformity* be the number of frames with the most frequent label in the interval, divided by the total number of frames. Our baseline is the average interval uniformity of the original shots without any partitioning. The combination of pauses and periodicity partitioning improves the average interval uniformity (Table 2). This is very promising, since the average inter-

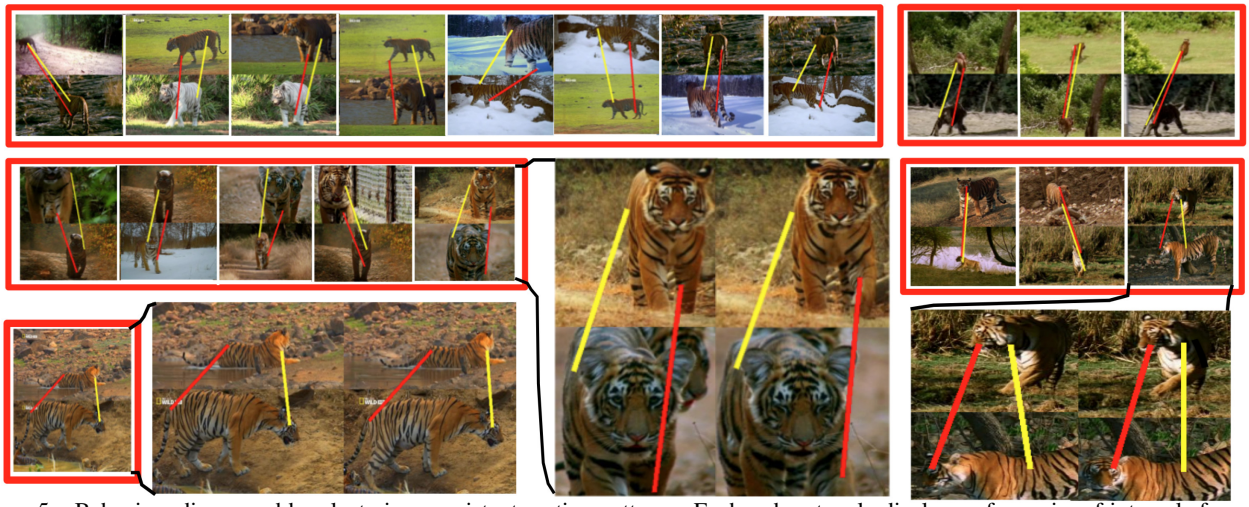

Figure 5. Behaviors discovered by clustering consistent motion patterns. Each red rectangle displays a few pairs of intervals from one cluster, on which we connect the anchors (yellow) and swings (red) of two individual PoTs that are close in descriptor space. The enlarged version show how the connected PoTs evolve through time and give a snapshot of the captured motion pattern in each cluster. The behaviors shown are: two different ways of walking (left, top and middle), sitting (bottom left), running (top right), and turning head (bottom right).

val uniformity is close to 90%, and the number of intervals found approaches the ground-truth number. In Table 1 we report the number of single-pattern intervals found by each method, grouped by motion pattern. Here, we only increase the count for intervals from distinct shots. Otherwise, we could approach ground truth by simply chopping one continuous behavior into smaller and smaller pieces. We chose this counting method because finding instances of the same pattern performed by different tigers is our goal. If we were to cluster whole shots, many patterns would be lost, and only a few dominant classes would emerge from the data. Instead, our method finds intervals for each label.

We report purity and ARI for the clusters of partitioned intervals. As the ground-truth label for a partitioned interval, which may not coincide exactly with a ground-truth interval, we use the label of the majority of the frames in the interval. To make this comparison fair, we evaluated the IDTF descriptors on the same single-pattern intervals. As before, PoTs outperform IDTFs (Fig. 4, bottom row). Finally, we show a few qualitative examples of the clusters found by our method in Fig. 5.

6. Discussion

We emphasize that the only supervision in the entire process is the initial video label (*i.e.*, we know the video contains a tiger or dog, respectively) and that the only cue used is motion, encoded by the PoT descriptor.

Appearance features have proved useful for traditional action recognition tasks [14, 34], since many activities are strongly characterized by the background and the apparel involved (*e.g.*, diving can be recognized from the appearance of swimsuits, or a diving board with a pool below). The dog dataset fits this paradigm: the appearance of the individual dog and the background was tightly correlated with the dog's behavior (*e.g.*, only one dog knew how to skateboard) and so adding appearance should be beneficial. Because PoTs and appearance features are complementary, we see the expected performance boost by adding the additional information. However, the tigers dataset shows that adding appearance features can be detrimental. Tigers varied in appearance (orange and white tigers, cubs and adults, etc.) but all tigers performed a variety of behaviors. On this dataset, the motion-only PoT descriptor outperforms all tested alternatives that included appearance information.

An essential feature of our method is that a collection of PoTs can encode detailed information about the relative motion between many different parts of an object. PoT anchors are scattered across the object; each may move with its own unique trajectory. Simplifying PoTs to a star-like model where all anchors coincide with the center of mass of the object (*i.e.*, normalizing by the dominant object motion) would result in a loss of expressive power and would be less robust for highly deformable objects.

PoTs are selected bottom-up and need not relate to the kinematic structure of an object class. This allows the extraction process to apply to any object and to leverage those PoTs that are discriminative for the particular class rather than being limited to pre-defined relationships We have shown that clustering built on top of PoTs finds motion patterns that are consistent across many shots. While many common behaviors (*e.g.*, walking) are cyclic, our method focuses instead on consistency across occurrences rather than periodicity within an occurrence. Periodic motion is exploited during partitioning, but the clustering procedure itself makes no such assumption, enabling us to discover behaviors such as a tiger turning its head.

Acknowledgments. We are very grateful to Anestis Papazoglou for help with the data collection, and to Shumeet Baluja for his helpful comments. This work was partly funded by a Google Faculty Research Award, and by ERC Starting Grant "Visual Culture for Image Understanding".

References

[1] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 1

[2] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998. 1

[3] L. Del Pero, S. Ricco, R. Sukthankar, and V. Ferrari. Dataset for articulated motion discovery using pairs of trajectories. http://groups.inf.ed.ac.uk/calvin/proj-pots/page/, 2015. 2, 5

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 1

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 7

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. on PAMI*, 29(12):2247–2253, December 2007. 1

[7] T. Hospedales, S. Gong, and T. Xiang. A Markov clustering topic model for mining behaviour in video. In *ICCV*, 2009. 2

[8] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. on PAMI*, 28(9):1450–1464, 2006. 2

[9] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. 6

[10] A. Jain, A. Gupta, M. Rodriguez, and L. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013. 2

[11] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, 2012. 2

[12] Y.-G. Jiang, J. Liu, G. Toderici, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS: ECCV workshop on action recognition with a large number of classes, 2014. 2

[13] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967. 5

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 8

[15] W.-H. Kim and J.-N. Kim. An adaptive shot change detection algorithm using an average of absolute difference histogram within extension sliding window. In *ISCE*, 2009. 5

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011. 1

[17] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 2

[18] D. Kuettel, M. Breitenstein, L. van Gool, and V. Ferrari. What's going on? Discovering spatio-temporal dependencies in dynamic scenes. In *CVPR*, 2010. 2

[19] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2007. 2

[20] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010. 2

[21] P. Matikainen, M. Hebert, and R. Sukthankar. Trajections: Action recognition through the motion analysis of tracked features. In *ICCV Workshop on Video-Oriented Object and Event Classification*, 2009. 2

[22] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, 2010. 2

[23] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009. 2

[24] S. Narayan and K. R. Ramakrishnan. A cause and effect analysis of motion trajectories for modeling actions. In *CVPR*, 2014. 2

[25] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, December 2013. 4, 6, 7

[26] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2, 6

[27] D. Ramanan, A. Forsyth, and K. Barnard. Building models of animals from video. *IEEE Trans. on PAMI*, 28(8):1319 – 1334, 2006. 3

[28] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971. 6

[29] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012. 2

[30] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *ECCV*, 2010. 2

[31] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009. 1

[32] J. M. Santos and M. Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *ICANN*, 2009. 6

[33] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proc. ICPR*, 2004. 1, 2, 7

[34] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. Technical Report CRCV-TR-12-01, University of Central Florida, 2012. 1, 2, 8

[35] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei. Discriminative segment annotation in weakly labeled video. In *CVPR*, 2013. 2

[36] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE T-CVST*, 18(11):1473–1488, 2008. 2

[37] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action Recognition by Dense Trajectories. In *CVPR*, 2011. 2, 4

[38] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 4, 5, 6, 7

[39] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*, 2014. 2

[40] X. Wang, X. Ma, and W. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models. *IEEE Trans. on PAMI*, 31(3):539–555, 2009. 2

[41] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 115(2):224–241, 2010. 2

[42] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *CVPR*, 2010. 2

[43] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Trans. on PAMI*, 35(7):1635–1648, 2013. 2, 7

[44] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009. 1

[45] X. Zhao and G. Medioni. Robust unsupervised motion pattern inference from video and applications. In *ICCV*, 2011. 2