Social behavior recognition in continuous video

Xavier P. Burgos-Artizzu*, Piotr Dollár†, Dayu Lin+, David J. Anderson*, Pietro Perona*
*California Institute of Technology, †Microsoft Research, Redmond, +NYU Medical Center
xpburgos@caltech.edu, pdollar@microsoft.com, dayu.lin@nyumc.org, {wuwei,perona}@caltech.edu

Abstract

We present a novel method for analyzing social behavior. Continuous videos are segmented into action 'bouts' by building a temporal context model that combines features from spatio-temporal energy and agent trajectories. The method is tested on an unprecedented dataset of videos of interacting pairs of mice, which was collected as part of a state-of-the-art neurophysiological study of behavior. The dataset comprises over 88 hours (8 million frames) of annotated videos. We find that our novel trajectory features, used in a discriminative framework, are more informative than widely used spatio-temporal features; furthermore, temporal context plays an important role for action recognition in continuous videos. Our approach may be seen as a baseline method on this dataset, reaching a mean recognition rate of 61.2% compared to the expert's agreement rate of about 70%.

1. Introduction

Detecting and classifying human and animal behavior from video is one of the most interesting challenges facing computer vision researchers [28, 2, 21, 6, 29]. Applications include: video surveillance, social robotics, scene understanding, ethology, neuroscience, medicine and genetics.

Automating the analysis of behavior is challenging. First, building a vision 'front end' that detects, segments, classifies and tracks bodies is technically difficult. Second, a crisp definition of 'behavior', including a distinction between short and simple 'movemes' [4], medium-scale 'actions' and more complex 'activities', still eludes us. A third obstacle is inadequate benchmark datasets to guide our thinking and to evaluate our algorithms (see Section 2).

Studying animal behavior [17, 2, 8, 37, 3, 7, 15] is perhaps the best strategy to make progress on these issues thanks to a number of practical advantages. First, laboratory animals, typically flies and mice, are easier to detect and track than humans, thus the vision front-end is easier to build. Second, fly and mouse behavior is simpler and perhaps more objectively studied than human behavior, which makes it more likely that we will arrive sooner at a satisfactory definition of behavior. Third it is more practical and ethical to set up reproducible experiments and collect abundant video of animals, rather than humans, especially when the most interesting behaviors are concerned (e.g. courtship, aggression) and when one wants to explore the role of nature and nurture with systematic manipulations. Furthermore, expert scientists are willing to thoroughly annotate animal behavior videos in the course of their studies in ethology, ecology, neuroscience, pharmacology and genetics. Studying behavior in animals thus presents an opportunity for making progress on modeling and classifying behavior, especially social behavior, which is difficult to study in humans. We believe that knowledge gathered from studying animal models will eventually lead to progress in modeling and automating the analysis of human behavior.

The main contributions of this study are:
1 — The largest and richest behavior dataset to date. The Caltech Resident-Intruder Mouse dataset (CRIM13) consists of 237x2 videos (recorded with synchronized top and side view) of pairs of mice engaging in social behavior, catalogued into thirteen different actions. Each video lasts ∼10min, for a total of over 88h of video and 8M frames. A team of behavior experts annotated each video frame-by-frame using a Matlab GUI which we developed ad-hoc [23]. Videos, annotations, mice tracks and annotation tool are all available from www.vision.caltech.edu/Video_Datasets/CRIM13/.
2 — An approach for the automatic segmentation and classification of *social* 'actions' in continuous video. Multi-agent behavior poses new challenges: (a) multiple animals have to be localized and tracked even when they touch and overlap; (b) each behavior may be described with respect to multiple frames of reference: the enclosure, the agent and the other animal; (c) social behaviors are highly variable both in duration and in visual appearance, see Figure 1. A video example with the output of our approach is available from the project website.
3 — Novel trajectory features for behavior recognition. We generate a large pool of weak features from the position of tracked objects. The weak trajectory features outperform widely used spatio-temporal features, see Table 3.
4 — Exploring temporal context in behavior analysis. After

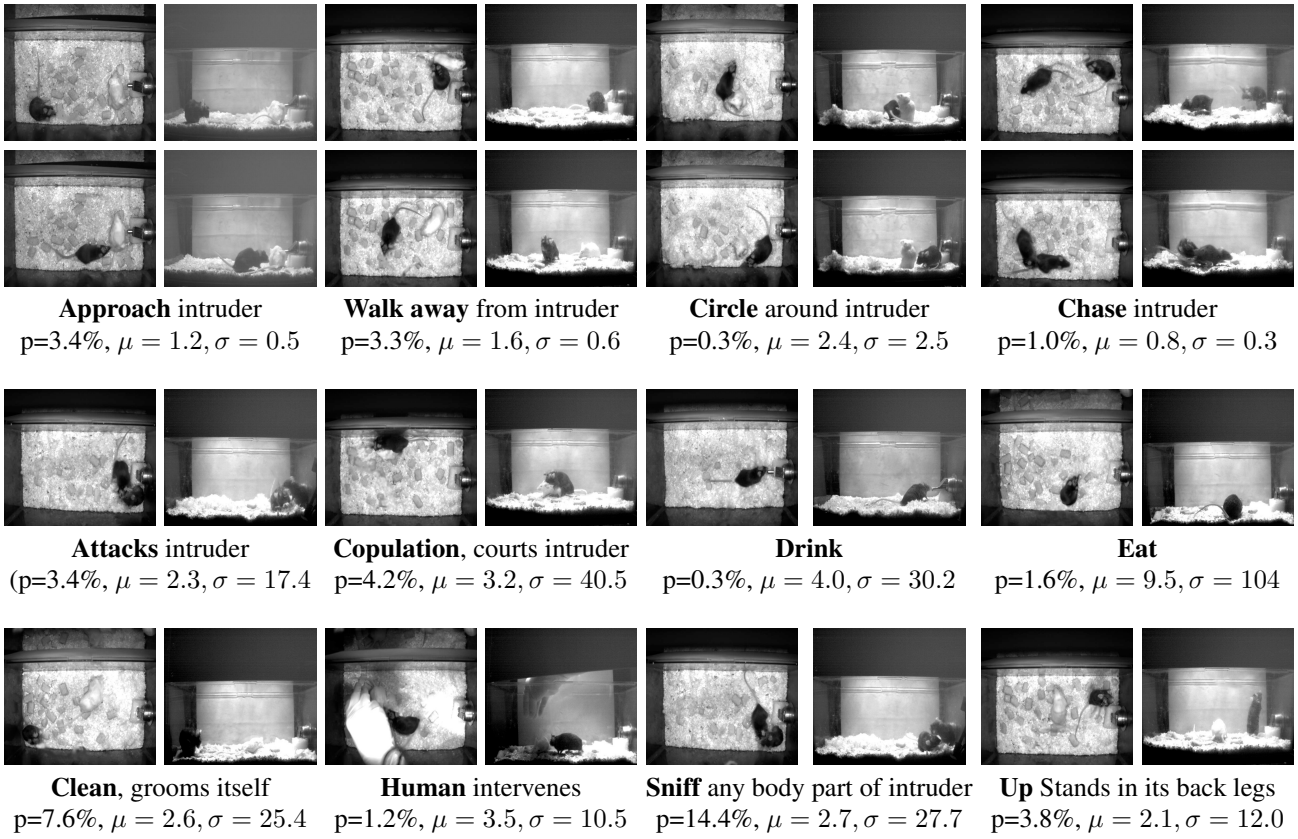


Figure 1. Behavior categories: frame examples, descriptions, frequency of occurrence (p), and duration mean and variance expressed in seconds (μ, σ). All behaviors refer to the cage resident, main mouse. The probability of *other* (not shown) is 55.3%.

first computing spatio-temporal and trajectory features from video, a second level analyzes temporal context using an extension of Auto-context [35] to video. The temporal context model helps segmenting the video into action 'bouts' improving results 8% across all behaviors.

Our method reaches a recognition rate of 61.2% on 13 categories, tested on 133 videos, see Table 3. Figure 2 shows a comparison of our approach with experts agreement on the 12 videos for which we have multiple expert annotations. On this smaller dataset, expert agreement is around 70%, while our method's performance is 62.6%.

However, CRIM13 still represents a serious challenge. Disagreement between human annotators lies almost entirely on the labeling of *other* behavior (less important), while our approach still makes mistakes between real behaviors. In fact, removing *other* from the confusion matrices results in a human performance of 91%, and only 66% for our approach. When counting *other*, our approach outperforms human agreement in 5 of the 12 behaviors (*approach, chase, circle, human, walk away*).

2. Related work

Datasets – What makes a good dataset? Behavior is interesting and meaningful when purposeful agents interact with each other and with objects in a given environment. The ideal behavior dataset identifies genuine agents acting freely in a well-defined scenario, and captures all spontaneous actions and activities. Filming should be continuous in order to allow a study of the structure of behavior at different scales of temporal resolution. Social behavior and agent-object interactions are of particular interest. Current most widely-used datasets for action classification, KTH [32], INRIA-XMAS [40], Weizmann [12], UCF Sports [30], Hollywood2 [25], YouTube [24], Olympic Sports [14] and UT videos [31] do not meet this standard: they are segmented, they are acted, the choice of actions is often arbitrary, they are not annotated by experts and they include little social behavior. Moreover, KTH and Weizmann datasets may have reached the end of their useful life with current state of the art classification rates of 94% and 99% respectively.

One of the first approaches to continuous event recognition was proposed by [43], although only a very small set of continuous video sequences were used. Virat [27] is the first large continuous-video dataset allowing the study of behavior in a well-defined meaningful environment. The

focus of this dataset is video surveillance and contains examples of individual human behavior and of interaction of humans with objects (cars, bags), containing more than 29h of video. Another useful dataset, collected by Serre and collaborators [15] focusses on single mouse behavior in a standard laboratory mouse enclosure. These datasets do not include instances of social behavior.

Features – Most action recognition approaches are based solely on spatio-temporal features. These features are computed in two separate stages: interest point detection and description. Popular spatio-temporal interest point detectors include Harris3D [19], Cuboids [8], and Hessian [13]. In datasets were the background contains useful information about the scene, as is the case for the Hollywood dataset, densely sampling points instead of running an interest point detector seems to improve results. Most effective spatio-temporal point descriptors are Cuboids (PCA-SIFT) [8], HOG/HOF [20], HOG3D [18] and eSURF [13]. More recently, relationships between spatio-temporal features have been used to model the temporal structure of primitive actions [5, 31].

Other features that have shown good performance are Local Trinary Patterns [41] and motion features derived from optical flow [9, 1]. Recently, space-time locally adaptive regression kernels (3DLSK) have been shown to reach state-of-the-art recognition on the KTH dataset with only one training example [33]. Biologically-inspired hierarchies of features have also been widely used [16, 34, 42, 22]. In [22] deep learning techniques were applied to learn hierarchical invariant spatio-temporal features that achieve state-of-the-art results on Hollywood and YouTube datasets.

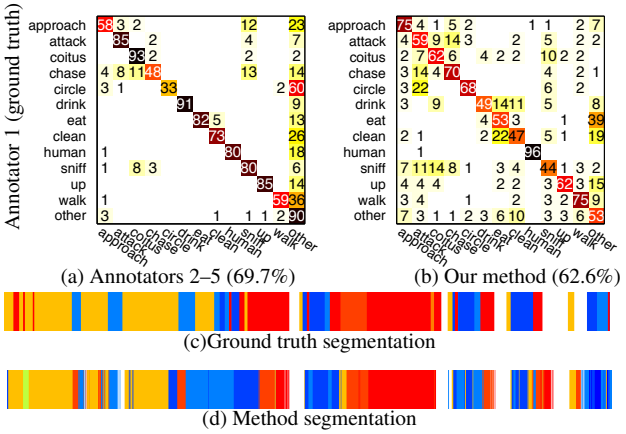
Another trend is to use an indirect representation of the visual scene as input to the classification, such as silhouettes, body parts or pose [29]. These approaches are generally sensitive to noise, partial occlusions and variations in viewpoint, except in the case of small animals such as flies, where trajectory features combined with pose information and body parts segmentation has proven to work well [7, 3].

Classifiers – The most common approach to classification is to use a bag of spatio-temporal words combined with a classifier such as SVM [36] or AdaBoost [11]. Other interesting classification approaches are feature mining [10] and unsupervised latent topic models [26].

Mouse behavior – Two works focussed on actions of solitary black mice on a white background [8, 15]. We tackle the more general and challenging problem of social behavior in mice with unconstrained color.

3. Caltech Resident-Intruder Mouse dataset

The dataset was collected in collaboration with biologists, for a study of neurophysiological mechanisms involved in aggression and courtship [23]. The videos always start with a male 'resident mouse' alone in a laboratory enclosure. At some point a second mouse, the 'intruder', is



(a) Annotators 2–5 (69.7%) (b) Our method (62.6%)

(c)Ground truth segmentation

(d) Method segmentation

Figure 2. Confusion matrices for comparison with expert's annotations on the 12 videos that were annotated by more than two experts. For results on the whole dataset, see Table 3. (a) Comparison between 'annotator1' and a group of 4 other annotators. (b) Comparison between 'annotator1' and the output of our approach. (c,d) Comparison of the segmentation of 2k frames of a video (available from project website) into behavior 'bouts'.

introduced and the social interaction begins. Just before the end of the video, the intruder mouse is removed. Behavior is categorized into 12+1 different mutually exclusive action categories, *i.e.* 12 behaviors and one last category, called *other*, used by annotators when no behavior of interest is occurring. The mice will start interacting by 'getting to know' each other (*approach, circle, sniff, walk away*). Once established if the intruder is a female, the resident mouse will likely court her (*copulation, chase*). If the intruder is a male, the resident mouse will likely *attack* it to defend its territory. Resident mice can also choose to ignore the intruder, engaging in solitary behaviors (*clean, drink, eat, up*). The introduction/removal of the intruder mouse is labeled as *human*. Figure 1 shows video frames for each behavior in both top and side views, gives a short description of each behavior, its probability p (how often each behavior occurs), and the mean and variance duration in seconds (μ, σ).

Each scene was recorded both from top- and side-views using two fixed, synchronized cameras. Mice being nocturnal animals, near-infrared in-cage lighting was used, thus the videos are monochromatic (visually undistinguishable from grayscale). Videos typically last around 10 min, and were recorded at 25fps with a resolution of 640x480 pixels, 8-bit pixel depth. The full dataset consists of 237 videos and over 8M frames.

Every video frame is labeled with one of the thirteen action categories, resulting in a segmentation of the videos into action intervals or 'bouts'. The beginning and end of each bout are accurately determined since behavior instances must be correlated with electrophysiological recordings at a frequency of 25Hz. Some behaviors occur more often than others, and durations (both intra-class and inter-

class) vary greatly, see Figure 1.

There were 9 different human expert annotators. Expertise of annotators varied slightly; they were all trained by the same two behavior biologists and were given the same set of instructions on how the behaviors should be annotated. Annotators used both top and side views in a video behavior annotator GUI developed in Matlab. This tool provides all the playback functionality needed to carefully analyze the videos and allows the manual annotation of behaviors into bouts (starting and ending frame). The tool is available from the project website. Each 10min video contains an average of 140 action bouts (without counting *other*). It typically takes an expert 7-8 times the video length to fully annotate it. The approximate time spent by experts to fully annotate the dataset was around 350 hours.

4. Proposed method

Most methods discussed in Section 2 deal with classification of pre-segmented short clips containing a single action. We approach the more complex problem of simultaneously segmenting video into single-action bouts and classifying the actions.

Our approach can be seen as an extension of Auto-context [35] to video. Auto-context has proven to perform well in high-level vision problems that benefit from learning a context model. Compared with other approaches, Auto-context is much easier to train and avoids heavy algorithm design, while being significantly faster. We find that incorporating temporal context information plays a crucial role in learning to segment videos, see Table 3.

Auto-context is easily described as a multi-step process. First, local features are computed from short sliding video time-windows (see Sec. 4.1) and used to train a 'base' classifier, outputting for each frame the probability that it belongs to each behavior. The process is repeated, adding to the original video features new features computed from the behavior probabilities of a large number of context frames. These new features provide information on the likely classification of frames preceding and following the current frame, therefore encoding temporal context and the transition probabilities between behaviors. The frames used for context can be either near or very far from the current frame, and it is up to the base classifier to select and fuse important supporting context frames together with local features. This process is repeated T times, or until convergence.

Our method uses AdaBoost [11] as the base classifier, where each weak classifier is a depth 2 tree, rather than the more common single stump. For each behavior, a binary classifier is trained by boosting on all training frames with labels indicating either the presence or absence of the behavior. Given $k = 1..K$ behavior types, each of the k binary classifiers will output a confidence $h^k(i) \in \mathbb{R}$ for that particular behavior being present in frame i [2]. Then, the final multi-label classification is achieved by assigning as final behavior type that which has the highest confidence level: $\hat{y}_i = \arg\max_k h^k(i)$. We found that classifier outputs do not need to be calibrated. Figure 3 shows the data flow of the proposed method.

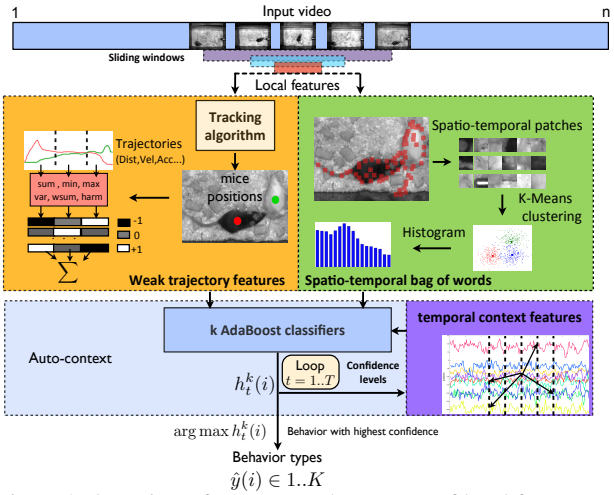


Figure 3. Overview of our approach. Two set of local features are first computed from video. Classifiers trained on these features estimate confidence levels h_t^k for each behavior. Temporal context features are built from the confidence levels and used to compute new confidence estimates in a new iteration. In our case, Auto-Context converges in 2 iterations.

4.1. Local features

To compute local features, sliding windows are centered at each frame and both types of local features are computed inside them. The optimal sliding window size is directly related to the duration of behaviors, as well as to the encoding of each feature type. After several experiments, we established that using a window of 9 frames (≈1/3 s) was best for spatio-temporal features, while a combination of two window sizes (75 and 615 frames, i.e. 3 and 25 s) was most discriminative for trajectory features, see Figure 5.

Spatio-temporal bag of words are first computed on each video using a sliding window centered at current frame. We benchmarked several existing approaches (Section 5) and we finally chose Cuboids+Pca-Sift [8]. All parameters were set to its original configuration and bag of words was computed using the standard procedures outlined in [39], except for a reduced codebook size of 250, which proved to give best results (see Table 1 in Supp. Material).

Weak trajectory features are computed from the set of positions $x_{m_i}(t), y_{m_i}(t)$ of each mouse $m_i \in [1, 2]$ for each top view video frame t, see Figure 3. These positions are computed by an unpublished tracking algorithm developed by our group, which is able to maintain identities of mice trough repeated detections.

From the positions, meaningful trajectory information is

[2] Confidence values h are the result of AdaBoost after T iterations before applying the final binary threshold, $h(x) = \sum_{t=1}^{T}(log\frac{1}{\beta_t})h_t(x)$ [11].

computed, such as distance between mice, movement direction, velocities and accelerations, see Figure 4(a). Then, the algorithm generates a large pool of weak features from this information, in a similar way to what is done for object detection [38]. Given a sliding window centered at current frame, the method divides it in non-overlapping regions and applies to each a set of 6 different operations (sum, variance, min, max, gaussian weighted sum and 'harmonic'). After each operation, the method sums all region values, where each region can count positively (+1), negatively (-1), or be completely ignored (0). This process is repeated for all possible region combinations and for each one of the trajectory values of Figure 4(a). The 'harmonic' operation transposes regions of value (-1), therefore detecting cyclic repetitions.

4.2. Temporal context features

As part of Auto-context, behaviors are first classified based only on local features, and then in subsequent iterations by adding to the feature set a list of temporal context features, computed from confidence levels h^k_{t-1} output of the binary classifiers at the previous iteration $t-1$. Temporal context features are computed at each time sample by taking first order statistics over a time-window centered at that time sample. Design details are given in Figure 4(b).

5. Experiments

The purpose of this section is two-fold: 1) analyze all method parameters and 2) test the approach on the Caltech Resident-Intruder Mouse dataset. In Section 5.1, a metric to measure the classification error is defined. In Section 5.2, a subset of the dataset is used to optimize different method parameters and report classification results of some widely used spatio-temporal features. Finally, in Section 5.3, we report the results of our approach on CRIM13.

5.1. Error metric

Given a video, an annotation is defined by a list of behavior intervals (behavior type, starting frame, ending frame). The metric for comparing two annotations should measure both their similarity as well as their relative completeness (whether all behaviors present in the first annotation are also present in the second annotation). The second is especially important due to the highly unbalanced nature of CRIM13. In fact, a classifier that predicts all frames to be *other* would reach over 50% accuracy on a simple frame-by-frame comparison with ground truth (see Figure 1).

We propose to use as metric the average of the confusion matrix's diagonal, where the confusion matrix values are the per-frame average agreement between annotations for each pair of behaviors. The average per-frame agreement, computed across all frames, measures the similarity between annotations for that pair of behaviors. Then, by taking the average of the diagonal, we favor classifiers that reach high similarity with ground truth across all behaviors.

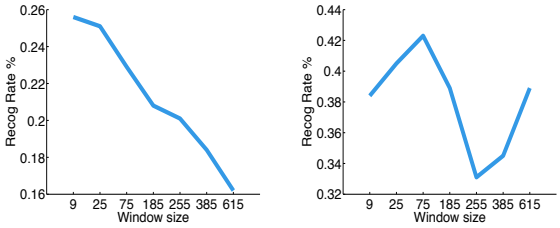


Figure 5. Sliding window sizes for both local feature types. (a) For spatio-temporal features, due to their encoding, a small window size gives best performance for all behaviors. Default size was set to 9 frames. (b) For trajectory features, best window size is directly related with duration of behavior: smaller window sizes detect short behaviors well, while larger detect long behaviors well. Default was set to use a combination of 2 windows of 75 and 615 frames each. See Table 2 in Supp. Material for more details.

5.2. Parameter analysis

To optimize the algorithm, we analyzed all method parameters. To avoid overfitting and to speed-up the process we used a small subset of the videos: 10 top view videos each for training and testing (300k frames), randomly chosen from the training set used in Section 5.3.

5.2.1 Spatio-temporal features

We chose to benchmark some of the most widely used spatio-temporal interest point detectors (Harris3D [19] and Cuboids [8]) and spatio-temporal descriptors (Cuboids Pca-Sift [8] , Hog/Hof [20] and Hog3D [18]) allowing all different combinations. We discarded the use of densely sampled features both because the background of our videos is uninformative, as well as computationally prohibitive. We also benchmarked the use of Local Trinary Patterns (LTP), [41], which due to its encoding should be more naturally suited for continuous behavior recognition. We used code from [8, 20, 18], extending functionality as needed to work on long videos.

Once spatio-temporal features are computed for all video frames, behavior classification results are achieved using a bag of words representation and an AdaBoost multilabel classifier, as outlined in Section 4. Spatio-temporal detectors are computed at a single spatial and temporal scale (3D patch size of $13x13$ pixels and 19 frames) and then bag of words is computed for each frame by looking at a window of 9 frames centered in the current frame, both best parameters, see Figure 5(a). To compute bag of words, the standard procedures outlined in [39] are used. Codebook size was benchmarked independently for each detector+descriptor combination with sizes ranging from 100 to 2000, see Table 1 Supp. Material. The best detector+descriptor combination was ran at multiple spatial and temporal scales ($\sigma = 2, 4, 8, \tau = 3, 5$). Table 1 shows the main results of each approach.

All approaches fail to segment the test videos due to the

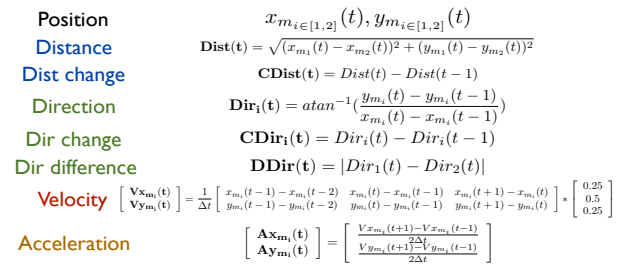


Figure 4. (a) Trajectory information computed from mouse tracks. Distance helps discriminate between solitary and social behaviors (mice far/close to each other), as well as to detect a transition from one to the other (*approach, walk away*). Velocity and acceleration helps detecting behaviors such as *attack, chase* or *clean* (stationary). Direction is important to distinguish between behaviors such as *sniff* (often head to head) with *copulation* or *chase* (head to tail). Pose estimation would clearly improve results in this aspect. (b) Temporal context features computed from confidence levels h^k_t output of each k binary classifier at previous iteration $t-1$, given a window of size sz centered at frame i. For each frame, 1536 context features are computed, using windows of $sz = 75, 185, 615$ to combine short, medium and long term context. Some of the features are also computed only on the previous or past frames to encode behavior transition probabilities.

Detector + Descriptor	Performance	Codebook size	fps
Harris3D+Pca-Sift	20.9%	250	2.7
Harris3D+Hog3D	18.7%	500	4.0
Harris3D+Hog/Hof	15.5%	500	1.1
Cuboids+Pca-Sift	**24.6%**	**250**	**4.5**
Cuboids+Hog3D	18.2%	250	8.7
Cuboids+Hog/Hof	19.8%	500	1.6
Cuboids+Pca-Sift multi-scale	16.4%	1000	0.8
LTP	22.2%	-	15

Table 1. Testing on a subset of CRIM13 was used to select Cuboids+Pca-Sift as the best of many state-of-the-art spatio-temporal features. Running Cuboids+Pca-Sift at multiple scales results in overfitting, with a drop of over 54% in performance from training to testing (not shown).

small size of the training set and a lack of context, resulting in a drop in performance of 30-40% from training (not shown). Differences between spatio-temporal features are very small, as in current datasets [39, 41]. Detectors seem to perform similarly, although Cuboids is much faster. As for descriptors, Pca-Sift outperforms the rest. Running Cuboids at multiple scales results in overfitting and does not improve performance at all, while also being much slower. LTP performs slightly worse than Cuboids+Pca-Sift, although being much faster. In the light of these results, we chose to use Cuboids+Pca-Sift for the rest of this work. However, our approach is in no way directly linked to the use of a specific set of spatio-temporal features.

5.2.2 Weak trajectory features

The only parameter of the trajectory features is the number of regions. We found that the optimal setting is to use the combination of 1, 2 and 3 regions, Table 2. With this setting, trajectory features outperform spatio-temporal features

	number of regions			
	1	[1,2]	[1,2,3]	[1,2,3,4]
Performance	29.7%	36.7%	**42.3%**	39.5%
# Features	114	684	2850	10260

Table 2. Benchmark of different number of regions for trajectory features computation. Trajectory features alone also suffer from a lack of temporal context to properly segment videos.

in the validation set. Their versatility makes them suffer slightly less from lack of temporal context. They are also much quicker to compute, with an average 155 fps (given the tracks, available from project website).

5.2.3 Classification settings

The only two parameters of the binary AdaBoost classifiers are the maximum number of weak classifiers (T) and the amount of frames sampled at each training iteration (S). To prevent overfitting and speedup training, each weak classifier is learnt using only a subset of S training frames, choosing a new subset at each iteration by randomly sampling with replacement. After evaluation, we chose $T = 250$ and $S = 1$k as optimal values, Figure 6. Note how sampling a small number of frames improves performance compared with larger numbers. As the number of frames increases, the weak classifiers overfit, resulting in a drop in performance.

5.3. Results

We report the results of our approach on CRIM13. The 237, 10min long videos were divided in two sets, 104 for training and 133 for testing. Table 3 shows the main results. To better study the role of each feature, we tried each separately: only trajectory features (TF), only spatio-temporal features from the side or from the top, both spatio-temporal features together (STF), and all of them combined ($TF + STF$). First column shows the performance

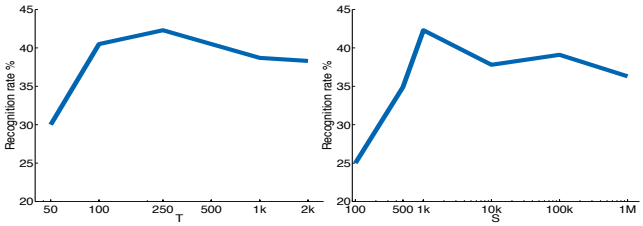


Figure 6. Role of T and S in classification. Experiments were carried on using only trajectory features (see Sec. 5.2.3).

Features used	Without Context	With Context
TF	52.3%	58.3%
STF	29.3%	43.0%
STF Top	26.6%	39.3%
STF Side	28.2%	39.1%
(Full method) TF+STF	53.1%	**61.2%**

Table 3. Method results on CRIM13. Context results were achieved after 2 Auto-Context iterations. Adding temporal context improves classification an average of 10% (14% on STF). For a comparison with experts annotations, see Figure 2.

of each combination of local features without context. Second column shows the results after adding temporal context features with Auto-context.

Without context, TF clearly outperforms STF, while the combination of both, $TF + STF$, improves classification 1% with respect to TF. The use of Auto-context improves performance 8% on the full method, 10% on average. In comparison, a simple temporal smoothing of the output labels only improves results 0.2%. Best results are achieved by the combination of both local features $TF{+}STF$, which improves TF 3%, reaching a final recognition rate of 61.2%. The upper bound for this task is 70%, which is the average agreement rate between expert's annotations, see Figure 2.

Although the metric used might suggest that the difference between TF and $TF + STF$ is small, analyzing each behavior separately shows that $TF + STF$ outperforms TF in 7 out of the 12 behaviors (*attack, copulation, chase, drink, eat, sniff, up*). Also, comparing the confusion matrices of STF from the side and STF from the top shows that the top view is best suited to detect 5 behaviors (*chase, circle, clean, sniff and walk away*) while the rest are best recognized from the side.

6. Discussion and conclusions

We present CRIM13, a new video dataset for the study of behavior with an emphasis on social behavior. The videos were collected and annotated by expert-trained annotators for a neurophysiological study of behavior [23]. CRIM13 is the largest and richest behavior dataset to date, containing over 8M frames and 12+1 different behavior categories. The videos are not pre-segmented into action bouts, therefore classification must proceed hand-in-hand with segmentation.

We proposed an approach for the automatic segmentation and classification of spontaneous social behavior in continuous video. The approach uses spatio-temporal and trajectory features, each contributing to the correct segmentation of the videos into behavior bouts. We benchmarked every component of our approach separately. On our dataset the combination of novel trajectory features with popular spatio-temporal features outperforms their use separately. This suggests that the study of behavior should be based on a multiplicity of heterogeneous descriptors. Also, we found that temporal context can improve classification of behavior in continuous videos. Our method's performance is not far from that of trained human annotators (see Figure 2 and Table 3) although there is clearly still room for improvement. While disagreement between human annotators lies almost entirely on the labeling of *other* behavior (less important), our method confuses some real behaviors. In fact, removing *other* from the confusion matrices results in a human performance of 91%, and 66% for our approach.

We believe that classification performance could be improved by adding new features derived from pose and applying dynamic multi-label classifiers that can prove more robust to unbalanced data. We will also further study the use of LTP features, due to their promising results and faster speed. Although in this work we were able to deal with behaviors of different durations, an open question is whether an explicit distinction between short and simple 'movemes', medium-scale 'actions' and longer and more complex 'activities' should be directly included into future models.

Acknowledgments

Authors would like to thank R. Robertson for his careful work annotating the videos, as well as Dr. A. Steele for coordinating some of the annotations. We also would like to thank Dr. M. Maire for his valuable feedback on the paper. X.P.B.A. holds a postdoctoral fellowship from the Spanish ministry of education, *Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I-D+i 2008-2011*. D.L. was supported by the *Jane Coffin Child Memorial Foundation*. P.P. and D.J.A. were supported by the *Gordon and Betty Moore Foundation*. D.J. is supported by the Howard Hughes Medical foundation. P.P. was also supported by *ONR MURI Grant #N00014-10-1-0933*.

References

[1] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *PAMI*, 32(2):288 –303, 2010. 3

[2] S. Belongie, K. Branson, P. Dollar, and V. Rabaud. Monitoring animal behavior in the smart vivarium. In *Workshop on Measuring Behavior, Wageningen, The Netherlands*. Citeseer, 2005. 1

[3] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson. High-throughput ethomics in large groups of drosophila. *Nature Methods*, 6(6):451–457, 2009. 1, 3

[4] C. Bregler. Learning and recognizing human dynamics in video sequences. In *CVPR*, 1997. 1

[5] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011. 3

[6] J. Candamo, M. Shreve, D. Goldgof, D. Sapper, and R. Kasturi. Understanding transit scenes: A survey on human behavior-recognition algorithms. *Intelligent Transportation Systems, IEEE Transactions on*, 11(1):206 –224, march 2010. 1

[7] H. Dankert, L. Wang, E. D. Hoopfer, D. J. Anderson, and P. Perona. Automated monitoring and analysis of social behavior in drosophila. *Nat Methods*, 6(4):297–303, Apr 2009. 1, 3

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005. 1, 3, 4, 5

[9] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, Nice, France, 2003. 3

[10] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, 2009. 3

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55:119–139, 1997. 3, 4

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 29(12):2247–2253, December 2007. 2

[13] G.Willems, T. Tuytelaars, and L. V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008. 3

[14] C.-W. C. J. C. Niebles and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 2

[15] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A. Steele, and T. Serre. Automated home-cage behavioural phenotyping of mice. *Nature communications*, 1(6):1–9, 2010. 1, 3

[16] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007. 3

[17] Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigentracking. In *CVPR*, volume 2, pages II–980. IEEE, 2004. 1

[18] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d gradients. In *BMVC*, 2008. 3, 5

[19] I. Laptev and T. Linderberg. Space-time interest points. In *ICCV*, 2003. 3, 5

[20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 3, 5

[21] G. Lavee, E. Rivlin, and M. Rudzsky. Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(5):489 –504, 2009. 1

[22] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. 3

[23] D. Lin, M. P. Boyle, P. Dollar, H. Lee, E. S. Lein, P. Perona, and D. J. Anderson. Functional identification of an aggression locus in the mouse hypothalamus. *Nature*, 470(1):221–227, 2011. 1, 3, 7

[24] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos " in the wild ". In *CVPR*, 2009. 2

[25] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 2

[26] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79:299–318, 2008. 3

[27] S. Oh and et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 2

[28] R. Polana and R. C. Nelson. Detection and recognition of periodic, nonrigid motion. *IJCV*, 23(3):261–282, 1997. 1

[29] R. W. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010. 1, 3

[30] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 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. 2, 3

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

[33] H. J. Seo and P. Milanfar. Action recognition from one example. *PAMI*, 33(5):867 –882, may 2011. 3

[34] G. W. Taylor, R. Fergus, Y. Lecun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010. 3

[35] Z. Tu. Auto-context and its application to high-level vision tasks. In *CVPR*, 2008. 2, 4

[36] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. 3

[37] A. Veeraraghavan, R. Chellappa, and M. Srinivasan. Shape-and-behavior encoded tracking of bee dances. *PAMI*, 30(3):463–476, Mar 2008. 1

[38] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 5

[39] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2010. 4, 5, 6

[40] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2-3):249–257, 2006. 2

[41] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, 2009. 3, 5, 6

[42] K. Yu, S. Ji, M. Yang, and W. Xu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010. 3

[43] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, volume 2, pages II–123. IEEE, 2001. 2