Recognizing Human Actions in the Motion Trajectories of Shapes

Melissa Roemmele*
roemmele@ict.usc.edu

Soja-Marie Morgens[†]
smorgens@soe.ucsc.edu

Andrew S. Gordon*
gordon@ict.usc.edu

Louis-Philippe Morency[‡]
morency@cs.cmu.edu

ABSTRACT

People naturally anthropomorphize the movement of nonliving objects, as social psychologists Fritz Heider and Marianne Simmel demonstrated in their influential 1944 research study. When they asked participants to narrate an animated film of two triangles and a circle moving in and around a box, participants described the shapes' movement in terms of human actions. Using a framework for authoring and annotating animations in the style of Heider and Simmel, we established new crowdsourced datasets where the motion trajectories of animated shapes are labeled according to the actions they depict. We applied two machine learning approaches, a spatial-temporal bag-of-words model and a recurrent neural network, to the task of automatically recognizing actions in these datasets. Our best results outperformed a majority baseline and showed similarity to human performance, which encourages further use of these datasets for modeling perception from motion trajectories. Future progress on simulating human-like motion perception will require models that integrate motion information with top-down contextual knowledge.

Author Keywords

Touch/Haptic/Pointing/Gesture; Animation; Crowdsourcing; Machine Learning

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

A skilled animator can create the illusion of life in nonliving objects by moving them in a humanlike manner, producing motion pictures with engaging narratives of anthropomorphic characters [32]. The perception of humanlike actions in object trajectories has been a longstanding interest in psychology, where its study bridges perceptual psychology and social psychology [3, 6, 7, 33]. In an influential early study, Fritz Heider and Marianne Simmel crafted a short film depicting the movements of two triangles and a circle in and around a box with a door, and asked subjects to describe what they saw [11]. These subjects instinctively anthropomorphized the two triangles and circle as humanlike characters, producing rich narratives about the social relationships, emotional states, and intentions of these objects. These studies led Heider to later propose an influential theory of how people attribute mental states to others in social interaction [10].

This relevance to social interaction has motivated several computer scientists to take an interest in Heider and Simmel's early film, with attempts to build automated systems that could perceive the film in much the same manner as Heider and Simmel's experimental subjects. Thibadeau [31] takes a symbolic approach to the perception of actions in the Heider-Simmel film. Beginning with the 2D coordinates of every line and arc in every other frame of the 1690-frame film, the system analytically generates symbolic descriptions of each frame that are matched to defined action schemas, such as opening the door or going outside the box. Pautler et al. [25] follows a related approach, beginning with object trajectory information from an animated recreation of the Heider-Simmel film. An incremental chart parsing algorithm with a hand-authored action grammar is then applied to recognize character actions as well as their intentions. These two rule-based approaches allow for the recognition of gross motor actions (moving to a location, hitting another character), but may not be appropriate for defining other motion trajectories like limping, shivering, tickling, and dancing. Current paradigms for AI recognition problems assume people only "know it when they see it," emphasizing the importance of systematically analyzing human data for these tasks, in contrast to the previous approaches. Accordingly, the problem of automatically interpreting motion trajectories may best be tackled using data-driven techniques applied to a wide array of crowdsourced examples.

In this paper, we describe our efforts to use machine learning to recognize human actions in the motion trajectories of shapes. To perform this task, we created novel animation data using a framework by which people can author animations in the style of Heider and Simmel. The resulting contribution, which we make available to the community, are two labeled datasets that enable a more systematic study of how people

*Institute for Creative Technologies, University of Southern California, Los Angeles, CA, USA

[†]University of California Santa Cruz, Santa Cruz, CA, USA

[‡]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

IUI'16, March 07–10, 2016, Sonoma, CA, USA
© 2016 ACM. ISBN 978-1-4503-4137-0/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2856767.2856793

perceive anthropomorphized actions in motion trajectories. We present initial results on two different supervised machine learning approaches for automatically recognizing actions in these datasets. In the first approach, we selected motion features to encode animations as sets of "spatial-temporal words" that were learned by a simple classifier. In our second approach, we replaced this feature selection process with a Recurrent Neural Network that automatically learned a feature representation of animations according to their action labels. The best results of these systems show significant improvement over a majority voting baseline and approximate a human measure of performance on this task, demonstrating the potential for future systems to exhibit human-like motion perception.

RELATED WORK

Because of its practical application to fields like robotics, human-computer interaction, and surveillance, there is a great deal of research devoted to recognizing human actions in video data. Increasingly utilized for this task are techniques for extracting the high-level 2D motion trajectories like those used in our work. For instance, Rao et al. [27] isolated changes in motion characteristics at particular spatial points in order to compute 2D motion trajectories for actions in videos (e.g. opening a door). Messing et al. [21] used a similar approach, where the velocities of particular key points on a person's body were tracked across videos of that person engaging household activities like eating or answering a phone. Such activities could then be recognized according to these "velocity histories" of the tracked keypoints. Vrigkas et al. [35] computed motion curves from videos using optical flow measurements. They identified the action associated with these motion curves by determining the longest common subsequences between these curves and those of known actions.

Some research on action recognition has successfully borrowed methods from text classification research. One such approach that we explore in this paper is to analyze discrete video segments as visual "words" that define the action portrayed in the video, in the same way textual words define the meaning of a document. Niebles et al. [23] used this technique with latent topical modeling algorithms, with the result that videos displaying similar actions (the "topics") were grouped together. Just as certain words are more important than others for determining the topic of a document, certain visual words may be particularly good cues for recognizing an action. Accordingly, Hoai et al. [13] automatically discovered the most discriminative temporal video segments that maximized recognition of mouse behaviors like sleeping and grooming. Like actions, gestures have also been recognized using visual word approaches [12, 36]. This work is particularly relevant to the current effort given that the animations in our dataset are generated by trajectories of underlying hand gestures.

Recent improvements on video-based action recognition are largely based on the new deep learning paradigm in computer vision. These models stack layers of neural networks to establish a "deep" hierarchical representation of the input data that maps directly to classification labels without the need for additional feature encoding. Baccouche et al. [2] and Karpathy et al. [14] observed significant performance gains over previous models using deep learning on existing action recognition datasets. We examine in this paper whether the benefits of deep learning on video classification also apply to the more abstract task of perceiving actions in animations of shapes.

Our work fits alongside other research exploring the connection between action perception and language. There is a growing body of work on automatically generating natural language descriptions of events shown in video [26, 29, 34]. Some effort has been given to the specific task of describing motion trajectories with verbs, as is the goal of our work. In particular, Koller et al. [17] designed finite state automata for labeling the motion trajectories of cars with German motion verbs. The automata encoded the attributes of a car's trajectory that have to be detected in order for the car to be described with a particular verb. Kojima et al. [16] employed a similar rule-based approach for doing this, using the trajectories of people's heads instead of cars. Action verbs were applied by evaluating predicates defining changes in motion for the people and objects in the video. Mathe et al. [20] automatically learned a semantic representation of motion verbs by discovering intervals with similar motion features across several videos depicting the same verb. Finally, Morrison et al. [22] used animations to categorize verb semantics by manipulating the motion features of animations and clustering the verbs people used to describe animations with similar motion features. Among these efforts, our work is the first to examine the verb labeling task as a machine learning classification problem where action labels are automatically acquired from a comprehensive dataset of motion trajectories.

DATA COLLECTION

Other researchers have created animations that resemble Heider and Simmel's film for use in their own studies. For instance, the Frith-Happe animations used in autism research [1] show two interacting triangles intended to elicit mental state attributions from observers. In Barrett et al. [3], research participants created similar animations by playing a game in which they acted out intentional actions with digital arrowheads, i.e. one player moved their arrowhead to "chase" or "fight" another player's arrowhead. However, the animations in these studies were not designed explicitly for automated action recognition across a large vocabulary of actions. To serve this goal, we created two new datasets of animated shapes whose motion trajectories are encoded transparently as time series data. In particular, each animation is represented as a series of frames sampled every 20 milliseconds, with each frame containing the position and rotation values of each shape in the animation at the time point for that frame. The position values are defined as (x,y) coordinates, and the rotation values are angles in degrees. In the first dataset of "Charades" animations (Figure 1), an animation depicts exactly one action and is annotated with a single label. In the second dataset of "Theatrical" animations (Figure 2), an animation is annotated with a sequence of actions whose depiction conveys a story, just like Heider and Simmel's film.

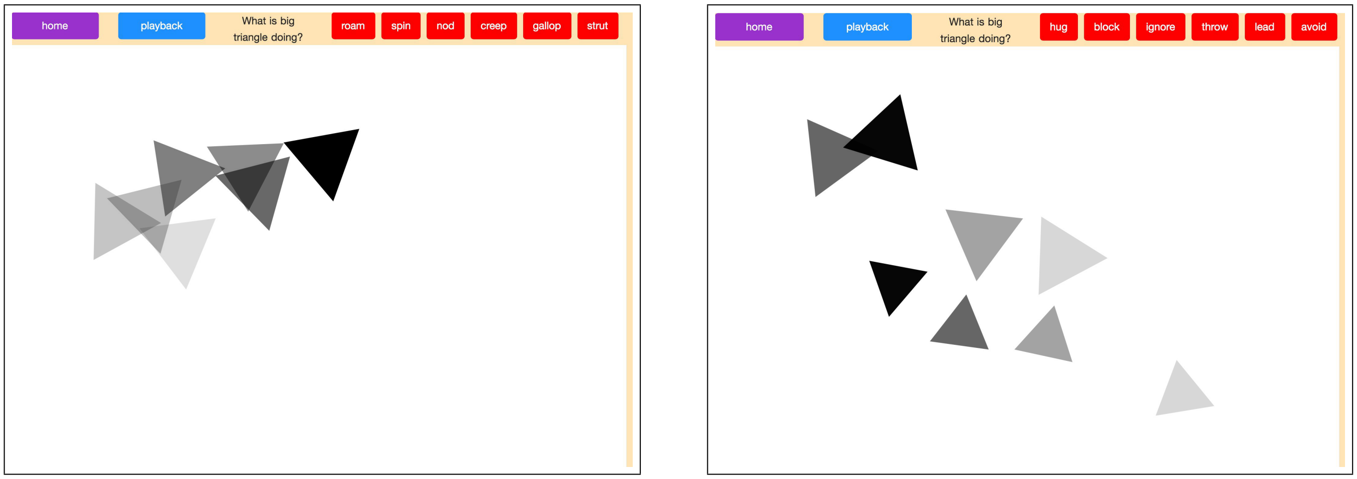

Figure 1. Examples of animations for the one-character action "roam" (left) and two-character action "ignore" (right) in the Charades game interface

Charades Animations

The Charades dataset was generated through a web-based game called *Triangle Charades* [28]. This game utilizes the same concept as the classic party game Charades, in which players convey concepts or entities using non-verbal language only. Here, players illustrate actions through a software interface by animating 2-D triangles resembling those in Heider and Simmel's film. Triangle Charades has two modes of play: "authoring" mode and "guessing" mode. In authoring mode, players are shown an English-language verb with a valency of either one (a one-character action like *spin* or *bolt*) or two (a two-character action like *hit* or *chase*). For one-character actions, players animate a single triangle as the agent of the action. For two-character actions, players are shown two triangles of distinct sizes, and instructed to animate the larger triangle as the agent of the action and the smaller triangle as the target of the action (e.g. "make the big triangle fight the little triangle"). Players animate the triangles by simply dragging them on a multitouch-enabled computer like an iPad. Animations cannot exceed 60 seconds in length. In guessing mode, shown in Figure 1, a player views an animation authored by another player, and then guesses from a set of six action labels which action the animation is depicting. A match between the guesser's selected action and the author's intended action is considered a correct selection. If the guess is incorrect, the guessed verb is eliminated and the player selects another verb until the correct one is selected, completing the guessing round.

The vocabulary of action labels for Charades animations was defined through guided intuition. The linguistic resource *English Verb Classes and Alternations* [19], which catalogs English verbs according to their syntactic behavior and semantics, was used to identify 200 verbs whose semantics involve whole-body motion. Since many verbs were synonymous, they were then manually clustered based on perceived similarity. The most canonical verb in each cluster was then selected to represent that action. For instance, the verb *hit* was picked to reference the action conveyed by the cluster of verbs *hit*, *collide with*, *jab*, *punch*, and *box*. The final vocabulary consisted of 31 one-character actions and 31 two-character actions, which are listed in Table 1. This dataset distinguishes between far more labels than any other known work on abstract action recognition, where typically only five or six labels were considered.

We used the crowdsourcing platform Crowdflower[1] to recruit people to play Triangle Charades. Additional players came through self-signup, yielding a total of 214 authors and 660 guessers for one-character animations, and 95 authors and 483 guessers for two-character animations. These players authored a total of 3041 one-character animations guessed a total of 9862 rounds (mean of 3.24 guessing rounds per animation), and 1762 two-character animations guessed 8655 rounds (mean of 4.91 rounds per animation). We used players' guessing data to validate that animations were perceived in accordance with the actions they intended to illustrate. To do this, we computed the number of guessing attempts it took guessers to correctly identify the action intended by the author of an animation, with 1 being the minimum number of guesses per animation and the maximum being 6. Only animations guessed by at least two players and guessed correctly in fewer than 3.5 average guesses (better than random chance) were included in the final validated dataset. Our final dataset contained 2060 one-character animations and 1158 two-character animations[2]. The mean length of these animations was 5.65 seconds for the one-character dataset and 6.45 seconds for the two-character dataset.

Theatrical Animations

In addition to determining which animations to include in the dataset, the guessing data also provides a measure of human performance on this action recognition task. In particular, recognition accuracy can be quantified as the mean number of guesses taken to select the correct action across all animations, where 1 is perfect accuracy on the task and 3.5 indicates no systematic recognition beyond random chance. This measure can also be used to examine recognition accuracy specifically on animations of the same action label. In the results

[1] http://www.crowdflower.com

[2] Available at https://github.com/asgordon/TriangleCOPA

1-character	*accelerate, bolt, bow, creep, dance, decelerate, drift, flinch, fly, gallop, glide, hop, jump, limp, march, meander, nod, roam, roll, run, scurry, shake, spin, stroll, strut, stumble, swim, trudge, turn, waddle, wave*
2-character	*accompany, approach, argue with, avoid, bother, capture, chase, creep up on, encircle, escape, examine, fight, flirt with, follow, herd, hit, huddle with, hug, ignore, kiss, lead, leave, mimic, play with, poke, pull, push, scratch, talk to, throw, tickle*

Table 1. List of action labels

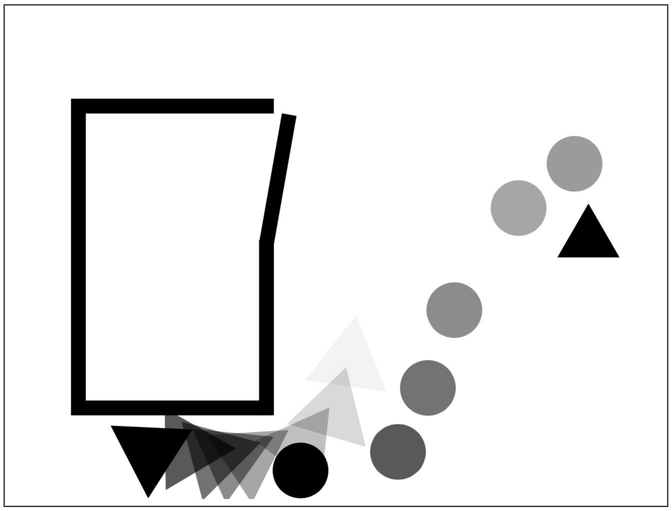

Figure 2. A Theatrical animation portraying the action sequence "creep up on" followed by "flinch"

section below, we use this analysis to directly compare automated recognition performance to human performance both overall and by action label.

Our second dataset of animations, which we term "Theatrical" animations, directly resemble Heider and Simmel's film (Figure 2). In addition to a "big" triangle and a "little" triangle, these animations feature a circle and a box with a hinged opening (the "door"). Each Theatrical animation is intended to convey a story by depicting a sequence of actions with different characters participating in each action. This dataset is less constrained than the Charades dataset, where only one action is depicted across an entire animation and the action type (one or two character) as well as character roles (big triangle as agent, little triangle as target) are pre-specified. To create the Theatrical dataset, we employed another web application called the *Heider-Simmel Interactive Theater* [8]. Here users apply the same touch-and-drag method as in Triangle Charades to author a movie (max 90 seconds) featuring the shapes and the door, without any instructions about what actions to depict. An animator on our team used the Heider-Simmel Interactive Theater to create 100 animations portraying sequences of both one-character and two-character actions. The mean length of these animations was 9.39 seconds. The same animator then annotated these actions using the Charades labels, establishing a gold standard test set for recognizing sequences of actions[3]. There were additional action labels in this gold standard that are not in Table 1, such as interactions between a particular character and the door and/or box (e.g. *knock*, *enter*, *exit*). After removing these labels, the mean length of the gold standard sequences for the current work was 2.24 action labels per animation.

EXPERIMENTS

The Charades dataset and Theatrical dataset offer two different opportunities for automated action recognition. In the Charades dataset, the goal is to predict a single action label for an entire animation. Conversely, the Theatrical animations contain more than one action per animation, so the task is to distinguish boundaries between sequential actions and label each action in the sequence. As we will discuss, the latter task is significantly more challenging, largely because recognizing transitions between actions is known to be as difficult as recognizing actions alone [37]. In this paper, we focus on the task of recognizing individual actions in Charades animations. We then demonstrate an initial application of the Charades-based models to the Theatrical animations in order to establish a starting point for future work on recognizing sequences of actions in motion trajectories.

We explored two alternative approaches to recognizing actions in Charades animations. In the first approach, we modeled animations as bags of spatial-temporal visual words, which were then used as features by a classifier to predict action labels. Our second approach applied the deep learning paradigm for classification, through which a recurrent neural network (RNN) was trained to predict labels from layered representations of motion trajectories. These approaches differ both in how they model the features as well as temporal structure of their data. First, the spatial-temporal words model relies on hand-selected features to encode animations, whereas the RNN automatically constructs features directly from the trajectory data. Second, the bag-of-words approach only considers temporal structure within a given interval of an animation rather than temporal relations between intervals. In contrast, the RNN models long-range temporal dependencies across an entire animation. The details of each model and their specific application to the current task are described below.

Spatial-Temporal Words Model

A spatial-temporal word is a categorical representation of an object's motion at a particular duration of the animation. Motion sequences can be encoded as sets of spatial-temporal words, an approach that has worked successfully for similar tasks such as video-based activity classification and gesture recognition [12, 23, 36]. To apply this approach, we split each animation into uniform-length intervals and computed the motion features of the shapes at each interval independently. The features we selected were those that have worked well for analyzing other types of motion trajectory data such

[3] Available at https://github.com/asgordon/TriangleCOPA

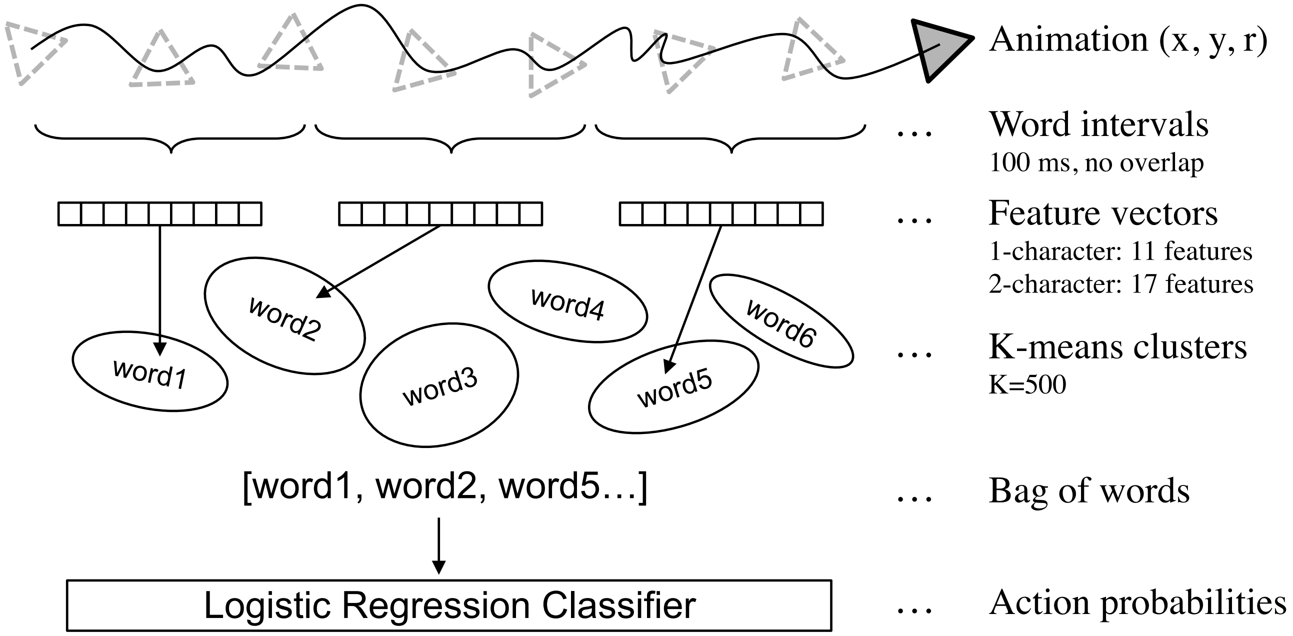

Figure 3. Spatial-temporal words model

as hand gestures [30] and pen sketches [9]. These features are all position-invariant, meaning that they are not affected by the absolute positions of the shapes in an animation but rather their change in position across frames.

Table 2 lists the motion features used by this model. The first 11 features constituted the feature set for one-character action recognition. For each interval in an animation, we calculated these features for the motion trajectory of the agent character performing the depicted action. The presence of two characters, one the agent and the other the target of an action, required additional features to be used for the two-character animations. For each interval in the two-character animations, we calculated the one-character features for the agent of the action. We then also calculated the mean of the six two-character features shown in Table 2 for the relative motion between the agent and target characters. With the addition of these features, the feature set for recognizing two-character actions contained 17 features total.

After computing its features, each interval was transformed into a word by categorizing it according to its feature-based similarity to other intervals. Intervals were assigned to clusters using the k-means algorithm with the Euclidian distance metric, and each interval was subsequently represented as the index of the cluster center closest to that interval. The set of cluster indices can be thought of as the “dictionary” of spatial-temporal words. Each animation was then encoded as a word vector composed of the number of times each word in the dictionary occurred in that animation. Because they have different feature sets, we generated separate dictionaries for the one-character and two-character datasets. We used each dictionary to transform the corresponding animations of that action type into bag-of-words vectors, which could then be used as input to a classification algorithm to predict action labels.

Figure 3 illustrates the full pipeline of the spatial-temporal words model. Interval length, offset length between intervals, and number of words (clusters) in the dictionary are all hyperparameters in this model. Once the animations are encoded as bag-of-words vectors, any classifier can be trained to assign labels from this feature representation. We chose Logistic Regression because it provides straightforward probability estimates describing the likelihood of each label being predicted for a given animation. This probability estimation enabled us to evaluate the model’s performance relative to human recognition, as we explain in the next section. We also chose to use Naive Bayes since it is commonly employed for text-based bag-of-words classification tasks and could possibly apply just as well to classifying spatial-temporal words.

Recurrent Neural Network Model

Encoding animations as spatial-temporal words enables them to be efficiently learned by a simple linear classifier, but this approach relies on assumptions that are difficult to verify. In particular, the trajectory features used in above approach (e.g. distance, rotation, velocity) were selected based on previous research and intuition, but it is possible that a different set of motion features would be more useful for the current recognition task. Moreover, even when the dictionary size is fixed, it is unclear if the k-means clustering will result in meaningful differences between words. If random changes in the initialization of the clusters yield a different assignment of intervals to words, it becomes difficult to determine what information is captured by the words. Morever, the hyperparameters of interval length, offset length between intervals, and dictionary size must be manually defined in this model. These decisions create extra overhead for the spatial-temporal word

	Name	**Description**
1-character features	Distance	agent's change in position between adjacent frames
	Rotation	agent's change in rotation between adjacent frames
	Angle	arctangent of Distance
	Angle Offset	difference between Angle and Rotation
	Velocity	derivative of Distance
	Rotational Velocity	derivative of Rotation
	Acceleration	derivative of Velocity
	Rotational Acceleration	derivative of Rotational Velocity
	Jerk	derivative of Acceleration
	Curvature	curvature of the agent's change in position between adjacent frames
	Angle Change	derivative of Angle
2-character features	Relative Distance	distance between agent and target at each frame
	Relative Angle	arctangent of Relative Distance
	Relative Velocity	derivative of Relative Distance
	Relative Acceleration	derivative of Relative Velocity
	Relative Jerk	derivative of Relative Acceleration
	Relative Angle Change	derivative of Relative Angle

Table 2. List of motion features used in spatial-temporal bag-of-words model

model since they can only be made by inefficiently trying all possibilities.

Recurrent Neural Networks (RNNs) avoid this overhead because they directly learn from input sequences without requiring any manual feature engineering. RNNs have become a state of the art technique for processing sequence data, and have recently demonstrated success specifically on action recognition tasks [2, 14]. We aimed to determine whether an RNN with no explicit knowledge of spatial-temporal words could recognize actions in Charades animations with comparable performance. To do this, we applied an RNN derived from the Elman network [5], whose general architecture consists of an input layer, one or more recurrent hidden layers, and an output layer all connected adjacently by weights. For our task, the input layer encodes the shapes' motion trajectories in the animation, while the hidden layers represent the continuous underlying state of the animation so far. The output layer is equivalent to a Logistic Regression classifier, using the values in the hidden layer to predict action label probabilities for the animation. Figure 4 illustrates this architecture with two hidden layers. The size (number of units) in each hidden layer is a hyperparameter. The weight matrices connecting the layers are the parameters that ultimately determine the probability estimates used to predict actions. The values of these parameters are updated through back-propogation during training (see [18]). For each animation, the model iterates through each frame of the animation and uses both the position and rotation data in the current frame and the hidden values at the previous frame to compute the hidden values at the current frame. The same recurrence is applied when there are multiple hidden layers: for each hidden layer, the state at the current frame is computed from the previous layer combined with the state of the current layer at the previous frame. Once the entire animation is processed, the values of the outermost hidden layer across the entire animation are averaged, and the average values are passed to the output layer. The Logistic Regression classifier (typically referred to as a softmax classifier) in the output layer uses these values to compute the probabilities of predicting each possible action label for that animation. When used to classify an animation, the RNN ultimately selects the action with the highest probability as the recognized action. As implied by the term "deep", RNNs are known to learn better representations of the input data as the number of hidden layers increases. We evaluated this claim for our task by training both an RNN with a single hidden layer and an RNN with two hidden layers. In both cases, the hidden layer interfacing with the output layer encodes a feature representation of the animations that can be used to predict action probabilities. The hidden layer representation can thus be viewed an alternative to the bag-of-words representation used in the spatial-temporal words model. We compared these two alternative feature encodings for the same animations by using Logistic Regression to classify them. We identified that if both models performed similarly, then we could conclude that the RNN feature representation was equivalent to the word-based representation for the action recognition task. Such a result would favor the use of RNNs for this task because they avoid the cost of manual feature engineering as well as the uncertain assumptions required by the spatial-temporal words model. We sought to evaluate this possibility in our experiments.

Methodology

For the experiments presented here, we randomly split each dataset into training, validation, and test sets, allocating 20% of the data to the validation set, another 20% to the test set, and using the remaining 60% for training. For each dataset, we trained the four models described above: the spatial-temporal words model with Logistic Regression (Words + LR), the spatial-temporal words model with Naive Bayes (Words + NB), the 1-layer RNN, and the 2-layer RNN. Additionally, we computed a baseline majority voting model that always predicted the most frequent action label in the test set. We used the validation data to set the interval length, offset length, and dictionary size hyperparameters for the spatial-

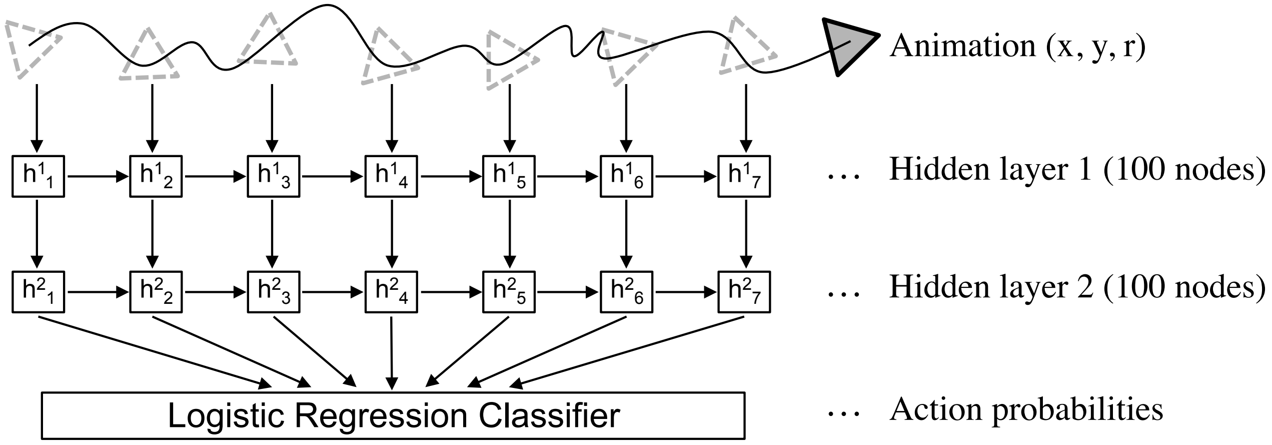
Animation (x, y, r)
h¹₁
h¹₂
h¹₃
h¹₄
h¹₅
h¹₆
h¹₇
… Hidden layer 1 (100 nodes)
h²₁
h²₂
h²₃
h²₄
h²₅
h²₆
h²₇
… Hidden layer 2 (100 nodes)
Logistic Regression Classifier
… Action probabilities

Figure 4. Recurrent neural network model

temporal words models. For both interval length and offset length between intervals, values of 5 frames, 10 frames, 15 frames, and 20 frames were evaluated as well as dictionary sizes of 100, 200, 300, 400, and 500. Similarly, we used the validation set to select the optimal hidden layer sizes for the RNN models, evaluating performance with 100, 300, and 500 hidden layer nodes. In training both the word-based Logistic Regression and RNN models, we performed a max of 1000 iterations of parameter updates, with early stopping if the training error had not improved in 50 iterations. For the RNN models, the RMSProp algorithm [4] was utilized to iteratively update the parameter weights of each model. After selecting the best performing configurations of all four models on the validation sets, we applied these models to the test sets, the results of which we discuss in the next section.

RESULTS AND DISCUSSION

The recognition accuracy of all models on the testing data appears in Table 3, with the best result from each dataset indicated in bold. Regarding the hyperparameters selected through validation, both spatial-temporal words models used an interval length of 100ms, an offset length of 0ms, and a dictionary size of 500 words. For the RNN models, the hidden layer size was 100 nodes (for both layers in the 2-layer RNN). We used the compute-intensive randomized test with stratified shuffling [24] to evaluate the statistical significance of differences in accuracy between models.

The accuracy of the majority baseline was 5.3% for the one-character dataset (predicting the action “hop” for all animations) and 5.6% for the two-character dataset (predicting “fight” for all animations). While all models exceeded this baseline, the difference was statistically significant only for certain models. Different patterns emerged from one-character and two-character experiments. The one-character recognition accuracy was lower across all models. The Words + LR model, with an accuracy of 12.6%, was the single model to significantly outperform the baseline. This model also significantly outperformed both of the RNN models. The results of the two-character models were more dramatic. Here all models showed significant improvement over the baseline. The Words + LR model demonstrated similar accuracy (12.5%) to the corresponding one-character model. The Words + NB model and 2-layer RNN performed significantly better than the Words + LR model with accuracies of 22.0%, and 25.0%, respectively. The 2-layer RNN for two-character animations obtained the best performance of all models, though its 25% accuracy did not significantly differ from the corresponding Words + NB model. While these results show that the spatial-temporal words representation is meaningful for this task, it’s interesting that this representation was much better utilized by Naive Bayes than by Logistic Regression. Naive Bayes assumes that each word is generated entirely independently from others, and while this seems incorrect for this task, it’s possible that it made recognition easier. The improvement of the 2-layer RNN over the 1-layer model, while not significant, suggests that it would be useful to explore even deeper RNNs for this task as additional layers may further improve performance. The two-character results show that the RNN models automatically built a feature representation as useful as the spatial-temporal words representation for the two-character data. Given that the spatial-temporal words representation demands more manual effort and makes more assumptions that may or may not be correct, RNNs may be a more straightforward approach to this problem. Moreover, while the current results leave it unclear, intuition suggests that the ability of RNNs to model long-range temporal structure would be informative for recognizing actions perceived in long motion trajectories.

One explanation for the weaker one-character results is that these actions are inherently harder to recognize than two-character actions when animated with shapes. However, as revealed in the next section, this performance disparity doesn’t appear in humans. This suggests the relative motion between the agent and target of an action was particularly informative to these models, whereas they were less sensitive to the structure contained in the motion trajectory of a single shape. Further work is needed to identify what features humans perceive in these trajectories that the current models fail to account for.

	Baseline	Words + LR	Words + NB	1-layer RNN	2-layer RNN
1-character	0.053	**0.126***†	0.085	0.080	0.073
2-character	0.056	0.125*	0.220*‡	0.185*	**0.250***‡

*Significantly better than baseline ($p < 0.05$)

†Significantly better than 1-layer RNN and 2-layer RNN ($p < 0.05$)

‡Significantly better than Words + LR ($p < 0.05$)

Table 3. Classification accuracy on Charades animations

	Baseline	Human	Words + LR	1-layer RNN	2-layer RNN
1-character	3.499	2.139	**2.623***†	2.817*	2.806*
2-character	3.574	2.114	2.419*	2.261*‡	**2.147***‡

*Significantly better than baseline ($p < 0.05$)

†Significantly better than 1-layer RNN and 2-layer RNN ($p < 0.05$)

‡Significantly better than Words + LR ($p < 0.05$)

Table 4. Classifiers' mean number of guesses until correct on 6-choice Charades task, compared to humans

Guessing Task

The accuracies in Table 3 are all fairly low, leading one to conclude that machine learning approaches perform poorly on this task. However, comparing the models' performance to human recognition suggests a different interpretation of the results. To make this comparison, test set accuracy can be evaluated according to the Charades guessing task, which measures the mean number of guesses until correct when the choice is constrained to only six labels. For every guessing round completed by a human player, the classifiers also performed this guessing task by predicting an action label for the animation from the same set of six actions shown to the player. Of these six labels, the model selected the one it assigned the highest probability, and this selection continued until the correct label was predicted. Table 4 shows the results of this task alongside human performance as well as a baseline approach. Here, the baseline selected actions simply in the order that they were presented to the player in the guessing round. We omitted the Naive Bayes model from this guessing task because its probability estimates are known to be poorly calibrated [38]. We again used stratified shuffling to evaluate the statistical significance of our results. The relative performance between models shows a similar pattern to Table 3, but new findings emerge from the comparison to human and baseline performance. Here, all models significantly outperform the baseline, signifying that they have all captured factors influencing action recognition to some degree. In accordance with the previous results, the Words + LR model was the best one-character guesser of all models at 2.623 mean guesses until correct. It made significantly better guesses than both the 1-layer and 2-layer RNN. For the two-character data, both the 1-layer RNN (2.261 mean guesses) and the 2-layer RNN (2.147 guesses) had significantly better guessing performance than the Words + LR model. Looking at human performance, players were generally better at guessing than any of the models, with players taking 2.139 mean guesses to select the correct one-character action in the test set, and 2.114 mean guesses for two-character animations. However, guesses made by the 2-layer RNN on the two-character animations did not differ significantly from the corresponding human guesses. Based on this we can conclude that the 2-layer RNN approximated human performance on recognizing two-character actions in motion trajectories.

The guessing task also enables us to examine recognition accuracy for each action label separately. Figure 5 visualizes the results in Table 4 to show the mean number of guesses for test animations corresponding to the same action label. For each action label in the one-character and two-character datasets, the guessing performance of the best model on the corresponding dataset is compared to human guessing for that action label. This analysis reveals interesting distinctions between the human and model recognition. In particular, while human performance was consistent across different actions, the models showed more variance in their ability to recognize specific actions. For instance, looking at the two-character actions recognized in fewer than 1.5 average guesses, only "play with" was recognizable to this degree for humans, compared to nine actions ("lead", "avoid", "push", "examine", "chase", "follow", "pull", "hug", and "encircle") for the 2-layer RNN. However, the model's strong guessing performance on these actions was countered by particularly weak performance on other actions. The actions "huddle with", "accompany", and "fight" all took more than 3 average guesses to identify, whereas even the least recognizable action for humans was identified in under 3 average guesses (2.750 guesses for "herd").

While an ideal comparison with human performance would measure human accuracy choosing from the full set of action labels, this six-choice guessing task is a more practical estimation of human performance. Choosing from over 30 labels for annotating a given animation is more cognitively demanding and time-consuming, and could deter a crowdsourcing approach to data collection. Moreover, this guessing task is useful in that evaluates the accuracy of probability estimates rather than just the accuracy of the best prediction. That RNNs are good at guessing suggests they are good at explicitly modeling the probability distribution over action labels.

Theatrical Recognition

Our second experiment was to move beyond single-action recognition in order to recognize sequences of actions in the Theatrical animations. We assessed the challenges of this

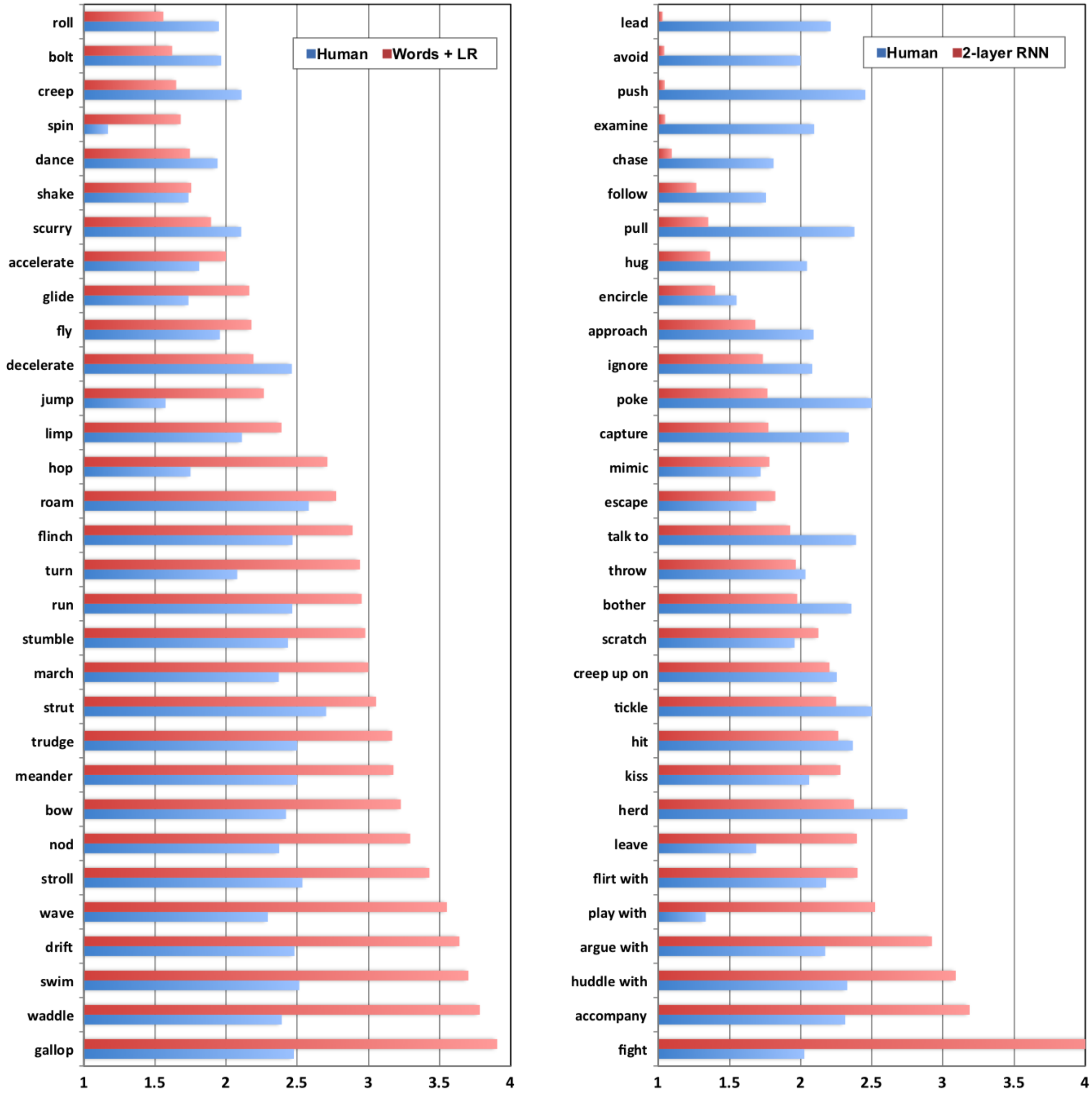

Figure 5. Comparison between humans and best model on mean number of guesses until correct for each verb in test set

task by using the 2-layer RNNs trained on the Charades animations to naively generate action sequences in which each action is predicted independently of other actions in the sequence. RNNs are well suited to this sequence labeling task because they explicitly model temporal relations across all frames of an animation. To establish a baseline approach to the Theatrical recognition task, we divided each Theatrical animation into segments of 150 frames (3 seconds) and applied both the one-character and two-character RNNs to recognize the action in each segment. In particular, for each character in the animation, the one-character classifier was given that character's motion trajectory in order to calculate the probabilities of the one-character labels for the action performed by the character. Additionally, for each ordered pair of characters in the animation, the two-character classifier was given those two characters' motion trajectories in order to determine the probabilities of the two-character actions where the first character in the pair is the agent of the action and the second character is the target. Thus, for the Theatrical animations with three characters (the big triangle, little triangle, and circle), we ran the one-character RNN three times (on each of the three characters), and the two-character RNN six times (on each of the six ordered pairs of characters). The probability estimates resulting from each on the nine RNN instances were pooled, and the action with the highest probability across all distributions was the action predicted for that animation segment. This procedure was repeated for each segment in the animation in order to yield a sequence of action labels.

Because the length of the predicted action sequence could differ from the number of gold standard actions for a particular animation, we evaluated the output in terms of precision and recall (without regard to the order of actions in the sequences) as opposed to accuracy. Precision was 1.4% and recall was 2.3%, with a resulting F-score of 1.8%, revealing that this recognition problem has a non-trivial solution. The poor performance can be attributed to the low accuracy on the previous Charades task, but also to the difficulty of automatically inferring what was predetermined in the Charades animations: the number of actions as well as the boundaries between them, and the roles of the characters participating in an action. This suggests that need for integrating motion trajectory information with top-down knowledge about the narrative context of an animation.

CONCLUSION AND FUTURE WORK

How and why people anthropomorphize abstract objects is a topic that has fascinated researchers across many disciplines. In this work, we showed how the high-level perception of human actions in motion trajectories can be modeled using machine learning, specifically by both a spatial-temporal bag-of-words model and a recurrent neural network. This is unique from previous work that attempted to analytically define trajectory features for actions based on a limited set of examples. We provide two new datasets that facilitate further analysis of this perceptual process. Our initial action recognition results on these datasets show the promise of further machine learning work on behavior classification from motion cues.

Our results suggest that accurate action recognition may require more than just detecting local motion cues, however. While using more sophisticated motion features might yield better recognition performance on the Charades dataset, the Theatrical animations, just like Heider and Simmel's film, are interpreted within a narrative context. Knowledge of this context is missing in the Charades animations even for humans, a likely reason why their own recognition is low. Contextual information such as the characters' actions in previous segments of the animation would likely provide a better model for recognition. New research has demonstrated how the recurrent neural network model used to recognize single-action animations in this work can be extended to model contextual dependencies between sequences of actions [2]. In order to successfully apply such an approach, it would be necessary to author and annotate additional Theatrical animations well beyond the current set of one hundred presented in this work.

We have focused on action perception as a bottom-up process, whereby motion reveals an action which in turn yields an explanation. But perception research suggests this is just as much a top-down process [15], as previous explanations for a shape's behavior influence how its motion is subsequently recognized. In particular, the beliefs, emotions, goals, social relationships, and personality traits attributed to the shapes in these animations lead to behavior interpretations which in turn generate further inferences about the internal states of these shapes. Simulating this process in machines requires deep commonsense reasoning that goes far beyond the task of action recognition and lies outside the scope of current machine perception research, but it is certainly a long-term vision for this work.

ACKNOWLEDGEMENTS

This research was supported by the Office of Naval Research, grant N00014-13-1-0286.

REFERENCES

1. Abell, F., Happe, F., and Frith, U. Do triangles play tricks? Attribution of mental states to animated shapes in normal and abnormal development. *Cognitive Development 15*, 1 (2000), 1–16.
2. Baccouche, M., Mamalet, F., and Wolf, C. Sequential deep learning for human action recognition. *Human Behavior Understanding* (2011), 29–39.
3. Barrett, H. C., Todd, P. M., Miller, G. F., and Blythe, P. W. Accurate judgments of intention from motion cues alone: A cross-cultural study. *Evolution and Human Behavior 26*, 4 (2005), 313–331.
4. Dauphin, Y. N., de Vries, H., and Bengio, Y. Equilibrated adaptive learning rates for non-convex optimization. In *Deep Learning Workshop, International Conference on Machine Learning* (2015).
5. Elman, J. L. Finding structure in time. *Cognitive science 14*, 2 (1990), 179–211.
6. Gao, T., McCarthy, G., and Scholl, B. J. The wolfpack effect perception of animacy irresistibly influences interactive behavior. *Psychological science 21*, 12 (2010), 1845–1853.
7. Gao, T., Newman, G. E., and Scholl, B. J. The psychophysics of chasing: A case study in the perception of animacy. *Cognitive psychology 59*, 2 (2009), 154–179.
8. Gordon, A. S., and Roemmele, M. An authoring tool for movies in the style of Heider and Simmel. In *Interactive Storytelling*. Springer, 2014, 49–60.
9. Hammond, T., and Paulson, B. Recognizing sketched multistroke primitives. *ACM Transactions on Interactive Intelligent Systems (TiiS) 1*, 1 (2011), 4.
10. Heider, F. *The Psychology of Interpersonal Relations*. Psychology Press, 1958.
11. Heider, F., and Simmel, M. An experimental study of apparent behavior. *The American Journal of Psychology* (1944), 243–259.
12. Hernández-Vela, A., Bautista, M. A., Perez-Sala, X., Ponce, V., Baró, X., Pujol, O., Angulo, C., and Escalera, S. Bovdw: Bag-of-visual-and-depth-words for gesture

recognition. In *21st International Conference on Pattern Recognition (ICPR)*, IEEE (2012), 449–452.

13. Hoai, M., Torresani, L., De la Torre, F., and Rother, C. Learning discriminative localization from weakly labeled data. *Pattern Recognition 47*, 3 (2014), 1523–1534.

14. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-Scale Video Classification with Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1725–1732.

15. Kinchla, R., and Wolfe, J. The order of visual processing: top-down,bottom-up, or middle-out. *Perception & psychophysics 25*, 3 (1979), 225–231.

16. Kojima, A., Izumi, M., Tamura, T., and Fukunaga, K. Generating natural language description of human behavior from video images. In *15th International Conference on Pattern Recognition*, vol. 4, IEEE (2000), 728–731.

17. Koller, D., Heinze, N., and Nagel, H. Algorithmic characterization of vehicle trajectories from image sequences by motion verbs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE (1991), 90–95.

18. LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521*, 7553 (2015), 436–444.

19. Levin, B. *English verb classes and alternations: A preliminary investigation*. University of Chicago Press, 1993.

20. Mathe, S., Fazly, A., Dickinson, S., and Stevenson, S. Learning the abstract motion semantics of verbs from captioned videos. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE (2008), 1–8.

21. Messing, R., Pal, C., and Kautz, H. Activity recognition using the velocity histories of tracked keypoints. In *IEEE 12th International Conference on Computer Vision*, IEEE (2009), 104–111.

22. Morrison, C. T., Cannon, E. N., and Cohen, P. R. When push comes to shove: A study of the relation between interaction dynamics and verb use. In *Working Notes of the AAAI Spring Symposium Workshop: Language Learning, an Interdisciplinary Perspective* (2004).

23. Niebles, J. C., Wang, H., and Fei-Fei, L. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision 79*, 3 (2008), 299–318.

24. Noreen, E. W. Computer intensive methods for hypothesis testing: An introduction, 1989.

25. Pautler, D., Koenig, B. L., Quek, B.-K., and Ortony, A. Using modified incremental chart parsing to ascribe intentions to animated geometric figures. *Behavior research methods 43*, 3 (2011), 643–665.

26. Ramanathan, V., Liang, P., and Fei-Fei, L. Video Event Understanding Using Natural Language Descriptions. *2013 IEEE International Conference on Computer Vision* (2013), 905–912.

27. Rao, C., Yilmaz, A., and Shah, M. View-invariant representation and recognition of actions. *International Journal of Computer Vision 50*, 2 (2002), 203–226.

28. Roemmele, M., Archer-McClellan, H., and Gordon, A. S. Triangle charades: a data-collection game for recognizing actions in motion trajectories. In *IUI* (2014), 209–214.

29. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., and Schiele, B. Translating Video Content to Natural Language Descriptions. *2013 IEEE International Conference on Computer Vision* (2013), 433–440.

30. Sadeghipour, A., Morency, L.-P., and Kopp, S. Gesture-based object recognition using histograms of guiding strokes. In *Proceedings of the British Machine Vision Conference* (2012).

31. Thibadeau, R. Artificial perception of actions. *Cognitive Science 10*, 2 (1986), 117–149.

32. Thomas, F. The illusion of life: Disney animation, 1995.

33. Tremoulet, P. D., and Feldman, J. Perception of animacy from the motion of a single object. *Perception 29*, 8 (2000), 943–952.

34. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R. J., and Saenko, K. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *Human Language Technologies: The 2015 Conference of the North American Chapter of the ACL*, Association for Computational Linguistics (2015), 1494–1504.

35. Vrigkas, M., Karavasilis, V., Nikou, C., and Kakadiaris, I. A. Action recognition by matching clustered trajectories of motion vectors. In *VISAPP (1)* (2013), 112–117.

36. Wang, T.-S., Shum, H.-Y., Xu, Y.-Q., and Zheng, N.-N. Unsupervised analysis of human gestures. In *Advances in Multimedia Information Processing*. Springer, 2001, 174–181.

37. Weinland, D., Ronfard, R., and Boyer, E. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding 115*, 2 (2011), 224–241.

38. Zhang, H., and Su, J. Naive bayesian classifiers for ranking. In *Machine Learning: ECML 2004*. Springer, 2004, 501–512.