Triangle Charades: A Data-Collection Game for Recognizing Actions in Motion Trajectories

Melissa Roemmele
roemmele@ict.usc.edu
Inst. for Creative Technologies
University of Southern California
Los Angeles, CA USA

Haley Archer-McClellan
archermcclellanh15@mail.wlu.edu
Dept. of Computer Science
Washington and Lee University
Lexington, VA USA

Andrew S. Gordon
gordon@ict.usc.edu
Inst. for Creative Technologies
University of Southern California
Los Angeles, CA USA

ABSTRACT

Humans have a remarkable tendency to anthropomorphize moving objects, ascribing to them intentions and emotions as if they were human. Early social psychology research demonstrated that animated film clips depicting the movements of simple geometric shapes could elicit rich interpretations of intentional behavior from viewers. In attempting to model this reasoning process in software, we first address the problem of automatically recognizing humanlike actions in the trajectories of moving shapes. There are two main difficulties. First, there is no defined vocabulary of actions that are recognizable to people from motion trajectories. Second, in order for an automated system to learn actions from motion trajectories using machine-learning techniques, a vast amount of these action-trajectory pairs is needed as training data. This paper describes an approach to data collection that resolves both of these problems. In a web-based game, called Triangle Charades, players create motion trajectories for actions by animating a triangle to depict those actions. Other players view these animations and guess the action they depict. An action is considered recognizable if players can correctly guess it from animations. To move towards defining a controlled vocabulary and collecting a large dataset, we conducted a pilot study in which 87 users played Triangle Charades. Based on this data, we computed a simple metric for action recognizability. Scores on this metric formed a gradual linear pattern, suggesting there is no clear cutoff for determining if an action is recognizable from motion data. These initial results demonstrate the advantages of using a game to collect data for this action recognition task.

Author Keywords

Games and Play; Animation; Crowdsourcing

IUI'14, February 24 - 27 2014, Haifa, Israel

http://dx.doi.org/10.1145/2557500.2557510

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The human imagination generates rich meaning from simple physical observations. In 1944, Heider and Simmel [6] experimentally demonstrated this by showing people a simple animated display of triangles and a circle (Figure 1). Participants in this social psychology experiment readily described the movement of the shapes in terms of human social interaction, where the shapes were human characters with psychological states like feelings, intentions, and desires. The stories participants told about the shapes were highly similar, most commonly describing two men (the two triangles) fighting for the affection of a woman (the circle). This ability to interpret motion patterns in an anthropomorphic way has been continually revisited by social scientists in the years since Heider and Simmel's foundational study [2, 4, 5, 7].

The field of artificial intelligence is also interested in this process by which people attribute feelings, intentions, and desires to others. Here, the motive of researchers is to design software that automates this social inference task. Such software has many practical applications, such as security monitoring and expressive user interfaces. Heider and Simmel's experimental task is a representative model of what a system must do in order to automatically interpret human behavior. This task abstracts away from the full complexity of human motion, reducing it to a simple trajectory displayed by a 2-dimensional shape. This simplicity is actually an advantage for an AI system. The system does not need to process a large amount of noisy perceptual input in order to do the inference task. Rather, because motion trajectories of shapes are straightforward to represent computationally, the system can focus on solving the behavior interpretation task itself.

But even this behavior interpretation task involves a multi-layered cognitive pipeline. At some point during this pipeline, as the shapes in the animation begin to be perceived as intentional characters, people judge what the characters are doing. The characters' motion trajectories are recognized as discrete behavioral actions such as fighting, chasing, or flirting, for instance. This action recognition task is required to further infer the character's

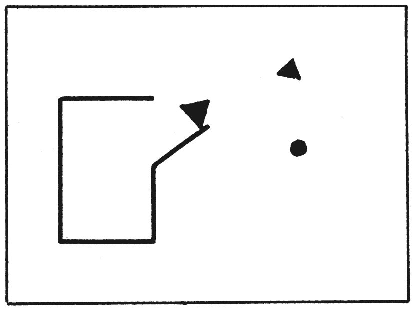


Figure 1. A frame from Heider and Simmel's film [6].

psychological states. Thus, if we want to build an AI system that does this behavior interpretation task, we first need to automatically recognize actions from motion trajectories.

There are two main challenges to performing this action recognition task. First, it is unclear exactly which human actions can be depicted with motion trajectories. No controlled vocabulary of recognizable actions currently exists, so we first need to define one. Once we have established this vocabulary, we need a large amount of motion trajectory data for each of its actions for use as training data in a supervised machine learning architecture. In this paper, we describe a game-based data-collection approach that resolves both of these problems. The game, called Triangle Charades, is a web application in which players animate shapes resembling those in Heider and Simmel's original experiment. By creating animations and evaluating other players' animations, players contribute to both of the above goals: they define which actions are recognizable from motion trajectories and they provide motion trajectory data for these actions. Using the resulting data we can then attempt to design a system that automatically perceives human actions in motion displays.

RELATED WORK

Psychologists and anthropologists have begun to more formally examine how people recognize human actions in motion trajectories, which has consequently motivated AI work on this task. There seem to be unique properties of motion displays that yield a perception of animacy, since not all displays are perceived in this way. Researchers in the visual perception community have tried to isolate the motion features that promote a human action-based interpretation [4, 5, 9, 11]. Klein et al. [7] revealed through eye-tracking methodology that animations perceived in terms of human action take longer to process than animations perceived merely as objective physical motion. This extra processing time for the animate displays may be due to people making inferences about the action the display evokes and its meaning in a social context.

Other work has focused on how, within animate displays, different motion cues yield different perceptions of action. Barrett et al. [2] collected motion trajectory data for intentional actions like fighting and playing, using a two-person software interface similar to the one used in our game. The players were assigned agents whose movement they controlled using a mouse. They manipulated the agents' movement to perform designated actions. For instance, when the action given was "chase", one player would make their agent chase the other player's agent, while the second player made their agent evade the first player's agent. The researchers then showed the motion trajectories generated by this method to a different group of participants. They found that often participants could identify the intended action associated with the motion trajectory just by looking at the trajectory data alone. However, they were only given six possible actions to choose from (chasing, courting, following, playing, fighting, and guarding).

If humans can recognize different actions based on motion data, then possibly so can a machine equipped with that data. There has been some effort in AI on tasks related to this one. Crick and Scassellati [3] collected motion trajectory data similar to the data we collect with our game, but through a very different process. They attached sensors to individuals and objects (e.g. a ball) participating in live-action playground games, and these sensors captured participants' positions at each point in the game. From this data, they could determine the degree of attraction and repulsion between participants in the game, which in turn enabled them to identify which action was occurring at any point in the game (e.g. "player A chased player B for 10 seconds"). Further, by putting sequences of actions together, they could determine which game was being played (e.g. tag versus catch). Young et al. [13] elicited motion trajectory data from people through an animation task like ours. In this work, artists animated characters on a table-top interface using physical pucks tracked by a motion capture system. The artists had the characters perform actions that were reflective of social roles like "lover" or "bully". The motion data from these animations was used to create characters that move autonomously in styles that display their designated social role. This is highly related to our work, but we proceed in the opposite direction: whereas Young et al.'s goal was to automatically establish motion from social information, our goal is to automatically establish social information (actions) from motion.

Researchers have started creating games for artificial intelligence problems because they provide an inexpensive way to elicit a lot of data from people. Games that collect data for computational tasks have been called "games with a purpose" [12]. While people play GWAPs for entertainment, they unwittingly provide data used to automate tasks that cannot otherwise be automated. There have been GWAPs designed for perceptual and social judgment tasks similar to the one described here [1, 10]. The success of game approaches on these tasks has motivated us to apply it to the action recognition problem.

SOLUTION

Our work is unique in that we want to identify all actions that have recognizable motion trajectories. In our work, an action is represented by an English-language verb (e.g. “hop”, “punch”, “slide”). Most English-language verbs do not have a motion trajectory representation that people can recognize. Rather than just relying on our own intuition about which actions are recognizable, we seek to determine this empirically. We can then define a controlled vocabulary of recognizable actions.

The game we designed, Triangle Charades, is based on the classic party game Charades, in which players must convey concepts or entities using non-verbal language only. Our game utilizes the same concept, except that here players must convey actions by animating 2-D triangles on a web interface. There are two modes of play in Triangle Charades. In both modes, the interface consists of a white background, or “stage”, on which solid black triangles move around, resembling the design of Heider and Simmel’s classic animation. Players can create motion trajectories for actions in “authoring” mode and attempt to identify the action depicted by other players’ motion trajectories in “guessing” mode. There is no requirement to play in one mode or the other, so players can simply switch modes whenever they choose.

Authoring mode

In authoring mode (Figure 2), players are presented with an action and they must create an animation depicting that action. To do this, they manipulate the movement of either one or two triangles, depending on how many characters are necessarily involved in the action. Single-character actions, such as “spin” and “tremble”, only require one character to perform the action, so players are given one triangle to manipulate. In contrast, two-character actions like “hit” and “chase” describe interactions between characters where one character is performing the action and the other is receiving it. In this case, two triangles appear, and players can manipulate both in order to depict the given action. One of the triangles appears reduced in size so that players can distinguish between the “big” triangle and the “little” triangle. The player is prompted to depict a two-character action via a command that appears above the stage: “big triangle [ACTION] the little triangle”. If the action is “chase”, for example, the player must animate the triangles so that it appears that the big triangle chases the little triangle. The prompt for single-character actions merely shows the action that the player must depict with the triangle. The interface enables players to control the triangles’ movement by simply touching them on the tablet, which establishes a dragging effect by which the triangles can be moved (translated and rotated). Since two-character actions require simultaneous animation of both triangles, a multi-touch tablet device must be used. However, a mouse-controlled device can be used to animate single-character actions. As the player drags the triangle(s), the game automatically records its successive movements to establish an animation. Players can view this animation by selecting “playback”. If a player starts animating and wishes to start over, they can select “reset” to discard their existing animation. There is a 60-second time limit for animations. If a player exceeds this time limit, their animation is automatically discarded and reset, and they receive an alert to complete the new animation within 60 seconds. When the player is satisfied with their animation for a given action, they can upload it to the game server by selecting “submit”. They will then see a new action to be animated. The submitted animation will become viewable to other players in guessing mode. It is possible for authors to animate the same action more than once.

Guessing mode

In guessing mode (Figure 3), players are shown an animation authored by another player, and prompted to identify the action depicted by the animation. There is no time limit for guessing, and players can replay the animation as needed by selecting “playback”. Rather than freely guessing actions for an animation, players are shown a set of six actions. From these options, the player must select which action the original author of the animation intended to depict. The other five options are randomly chosen from the set of all actions. A match between the guesser’s selected action and the author’s intended action is considered a correct selection. Players receive immediate feedback about whether their selection is correct. If the player is wrong, that option is removed and the player must make another selection. Players continue selecting actions until the correct one is selected. In the worst case, the only remaining action will be the correct one, which the player will have to select. As soon as the player selects the correct action, a new animation is loaded, along with a new set of options to choose from. Players are never given their own animations to guess. This process continues for as long as the player wants to keep playing in guessing mode.

Game Mechanics and UI Issues

Triangle Charades is coded in JavaScript and HTML. It is

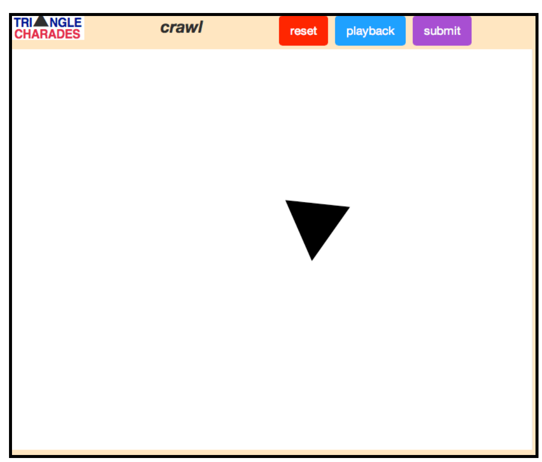


Figure 2. Triangle Charades in authoring mode. In this example, the player animates the action “crawl.”

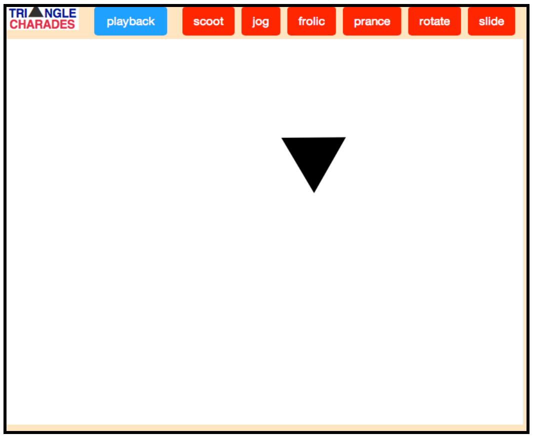


Figure 3. Triangle Charades in guessing mode. The user guesses which of six actions was intended in an animation.

accessible via the hyperlink http://charades.ict.usc.edu/. Players must log in on this page with a username and password. Accounts are required to play because Triangle Charades rewards points to players, and players' points accumulate across sessions of game play. A point scheme helps motivate players to author high-quality animations, and likewise provide careful guesses. Points rewarded in guessing mode are straightforward. If a player's first selection is correct (i.e. the one intended by the author), the guesser receives 10 points. Guessing correctly on the second, third, fourth, fifth, and final attempts yields 8, 6, 4, 2, and 0 points, respectively. Just as in live-action Charades, success in Triangle Charades requires collaboration between guessers and authors. Guessers rely on the authors to create animations whose depicted action is recognizable. Consequently, authors receive points for creating animations. Players automatically receive 10 points for every animation they author. Obviously, some actions are inherently not recognizable from animations, despite the skill of the author. However, we want to motivate players to create high-quality animations in spite of this. So, authors also receive an additional 1-point royalty every time another player correctly guesses their animation on the first attempt. Obviously, this additional reward for authoring is not immediate like the reward for guessing. Authors only see this royalty added to their score later as others players view their animation in guessing mode. To further incentivize players, we added three leaderboards to the main screen, listing players with the best acting and guessing abilities, and most overall points.

We encountered some interesting user interface design issues in programming the "dragging" behavior of the triangles. It is of course important that players be able to move (translate) the triangle from one point to another on the stage. However, it is also important to control the orientation (rotation) of the triangle. Manipulating the orientation allows players to point the triangle's "face" in different directions. This is key in expressing many of the actions that players must author. Establishing a movement paradigm by which players can simultaneously translate and rotate the triangle is not trivial. Our solution involves computing an angle between the point at which the player "grabs" the triangle (i.e. the point where they place their finger/mouse) and the triangle's center point. As the player drags the triangle across the stage, it rotates according to this angle. The effect is that the triangle moves in a relatively intuitive way, particularly if the player grabs the triangle at one of its vertices. A problem with this approach is that if the player grabs the triangle at a point too close to its center, the angle of rotation is unpredictable, making the triangle's dragging jerky. We implemented a simple fix to this issue by making the triangle "slippery" at its center. If the player tries to grab the triangle at a point within a certain distance from the center, the triangle remains stationary until the player reaches a point outside the center. In other words, the player might start dragging from the center, but as they drag their grab point "catches" on a better control point outside the center. The effect of this is actually barely noticeable to the player but it prevents any jerkiness in the triangle's movement.

Actions

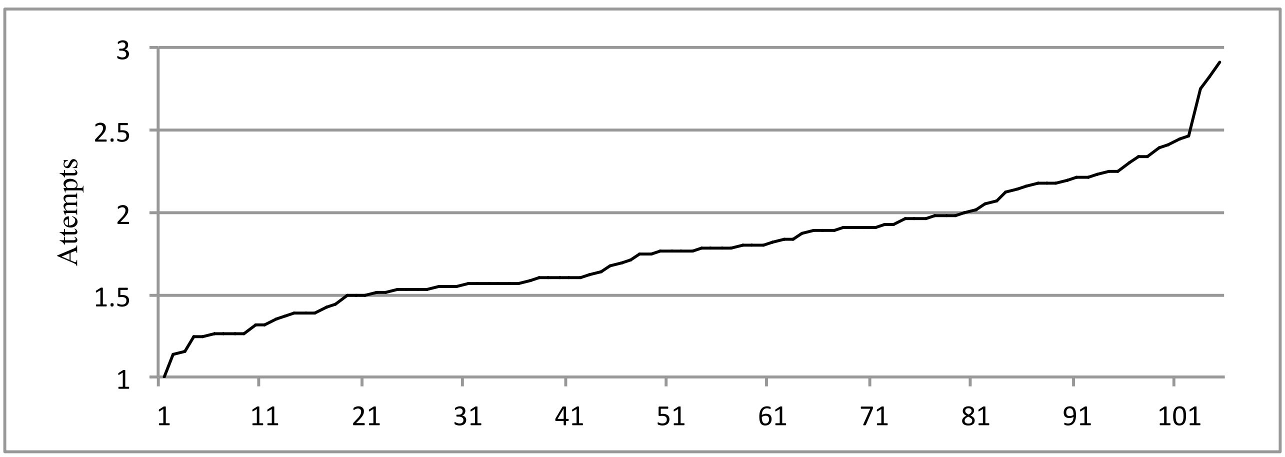
Guessing correctly in Triangle Charades can be difficult, as animations may not clearly depict their intended action. Sometimes, this is because the author did not do a good job of animating that action. However, it could be that the action is not easily depicted through triangle animation. If this is the case, the action should not be included in vocabulary of actions recognizable from motion trajectories. We define the term recognizability to mean the degree to which an action can be recognized from a motion trajectory. Triangle Charades allows us to quantitatively evaluate an action's recognizability. First, several authors animate the action. Then, those animations are presented to several players in guessing mode. In guessing mode, for each attempt, we identify how many attempts the guesser has already made, and whether or not this attempt yields the correct guess. Clearly, an action is a good depiction of an action if players typically guess that action on the first few attempts. In contrast, if it takes several attempts to identify the action represented by an animation, then the action might not be recognizable from a motion trajectory. Based on this idea, we can compute the average number of attempts it takes players to guess the action correctly from its animation. This metric represents an action's recognizability. To be clear, because recognizability is averaged over several animations, it is not subject to anomalous low-quality animations for otherwise recognizable actions.

There is another useful concept for determining membership in our action vocabulary, which we call distinguishability. Some actions have motion trajectories that are highly confusable; for instance, animations for "punch" may closely resemble animations for "hit". If an animation presented in guessing mode depicts one of these actions, users may commonly guess the other similar action, yielding an incorrect attempt. This suggests that the two similar actions should be merged in our vocabulary of actions. Distinguishability thus refers to the degree to which one action motion's trajectory is distinguishable from another action's trajectory. More formally, action X's distinguishability from action Y equals the proportion of guesses where Y is selected when X is actually the intended action. As a pairwise measure, computing distinguishability requires a lot more game data than computing recognizability, because there are only six actions to choose from for a particular guess. Each action needs several guesses before every single other action would appear in the set of options for those guesses.

Actions that are highly recognizable and highly distinguishable from all other actions should be included in a vocabulary of actions recognizable from motion trajectories. The game enables us to collect motion trajectory data for a large set of actions that potentially participate in this vocabulary. The game also enables us to filter this data: we then remove all actions/trajectories with low recognizability and low distinguishability. As mentioned above, the actions in the predefined set were all English language verbs. We knew that it was impractical to consider all English language verbs for this set, since most such verbs are clearly not expressible in the medium of 2-D whole-body motion trajectories. To determine the best candidates for this set, we consulted a linguistic resource, Levin's [8] *English Verb Classes and Alternations*. This book is intended to be a thorough categorization of verbs according to their grammatical behavior and meaning. We examined verb classes whose semantics involve whole-body motion. An example is the "run" verb class, which includes verbs similar to "run", such as "hop", "roll", and "scramble". Verbs from the selected classes were added as potentially recognizable actions to our game. This set of potentially recognizable actions included 105 single-character actions and 89 two-character actions.

PILOT STUDY

At the time of writing, we have implemented Triangle Charades and have used it to collect an initial dataset of action and motion trajectory pairs. Upon releasing the game, we recruited 87 pilot users to play it. These users played the game in both authoring mode and guessing mode, and the set of potentially recognizable actions was limited to single-character actions only. Two-character actions are more challenging to animate than single-



1. rotate	16. climb	31. flinch	46. hop	61. dance	76. walk	91. float
2. bolt	17. sneeze	32. meander	47. bound	62. strut	77. flap	92. traipse
3. roll	18. shake	33. bob	48. wave	63. prance	78. fly	93. trek
4. dart	19. descend	34. stroll	49. rush	64. bounce	79. scoot	94. jog
5. dash	20. zoom	35. accelerate	50. wobble	65. hurdle	80. march	95. cower
6. rise	21. swing	36. leap	51. depart	66. hurry	81. move	96. prowl
7. tremble	22. fall	37. oscillate	52. creep	67. wince	82. amble	97. promenade
8. ascend	23. limp	38. bow	53. wander	68. mosey	83. gallop	98. cringe
9. exit	24. roam	39. return	54. slither	69. hobble	84. slide	99. shuffle
10. convulse	25. vanish	40. loiter	55. frolic	70. sneak	85. hike	100. slow
11. jump	26. scramble	41. crawl	56. coast	71. scamper	86. run	101. clamber
12. wiggle	27. quake	42. scurry	57. hasten	72. trudge	87. collapse	102. swim
13. quiver	28. waddle	43. shudder	58. weave	73. stumble	88. swagger	103. skip
14. flutter	29. tiptoe	44. drift	59. recede	74. lumber	89. writhe	104. juggle
15. nod	30. turn	45. speed	60. saunter	75. glide	90. charge	105. clump

Figure 4. Recognizability scores for 105 single-character actions, represented by the mean number of attempts it takes to correctly identify that action in guessing mode. Scores are ordered lowest to highest.

character ones, because players must manipulate two different motion trajectories simultaneously. We intended to orient players to the game by asking them animate simpler actions first.

This pilot study resulted in data for 1013 authored animations and 5130 guessing attempts for animations. Based on this data, we computed the recognizability score for each action presented in the game. This data appears in Figure 4. Recognizability scores decrease by a gradual linear pattern, with no clear gap that could serve as a threshold for membership in the vocabulary of recognizable actions. This suggests that we will have to pragmatically define this threshold based on how well our machine learning paradigm can recognize actions from trajectories. The optimal threshold will maximize the number of actions in the vocabulary without compromising performance.

Distinguishability also determines vocabulary membership, by merging actions that have highly similar trajectories. However, this pilot study does not provide us with enough data to compute distinguishability scores for all actions. This is the next phase of this work, as well as collecting data for recognizable 2-character actions. Eventually, after we have determined our full vocabulary, remaining animations with low guessing performance will also be filtered from the data. This is a way of ensuring that our final dataset includes only high-quality motion trajectories.

Since actions are represented by English-language verbs, playing this game is partly dependent on English proficiency. As a tangent, we examined importance of this linguistic factor by asking 19 of the pilot users about their native language. Six of these players reported that English was not their native language. Though this is not a robust sample size, these players did seem to have more difficulty playing the game compared with native-English speakers. On average, the non-native English speakers required more guessing attempts to identify the action depicted by an animation (2.099, versus 1.733 for native speakers).

CONCLUSION

We designed a game to collect 2-D whole-body motion trajectories for human actions. This data will be used to train an AI system to automatically recognize actions from trajectories. Our pilot study shows that a game approach overcomes two existing inadequacies for this task. It provides a large set of training data, and it enables us to define a controlled vocabulary of the actions that are recognizable from motion data. For the latter, simple metrics like recognizability and distinguishability can be used to determine an action's membership in this vocabulary, though the precise membership threshold must be tuned based on system performance.

Games with a purpose are successful ways to collect data that does not require any specialized knowledge. This approach has been especially conducive to linguistic tasks, for instance. Triangle Charades benefits from this same advantage, since no special training is needed to create and interpret simple animations of triangles.

ACKNOWLEDGMENTS

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

REFERENCES

1. Ashraf, G., Why, Y. P., & Islam, M. T. (2010). Mining human shapes perception with role playing games. 3rd Annual International Conference on Computer Games, Multimedia and Allied Technology, Singapore.
2. Barrett, H., Todd, P., Miller, G., and Blythe, P. (2005) Accurate judgments of intention from motion cues alone: A cross-cultural study. Evolution and Human Behavior 26 (2005):313-331.
3. Crick, C. & Scassellati, B. (2008) Inferring Narrative and Intention from Playground Games. 7th IEEE International Conference on Development and Learning (ICDL 2008), Monterrey, California, August 2008.
4. Gao, T., McCarthy, G., and Scholl, B. (2010) The Wolfpack Effect: perception of animacy irresistibly influences interactive behavior. Psychological Science 21(12):1845-1853.
5. Gao, T., Newman, G., and Scholl, B. (2009) The psychophysics of chasing. A case study in the perception of animacy. Cognitive Psych. 59:154-179.
6. Heider, F. and Simmel, M. (1944). An experimental study of apparent behavior. American Journal of Psychology, 13, 1944.
7. Klein, A. M., Zwickel, J., Prinz, W., & Frith, U. (2009). Animated triangles: an eye tracking investigation. Quarterly journal of experimental psychology, 62(6), 1189–97.
8. Levin, B. (1993). English verb classes and alternations: A preliminary investigation. Chicago: University of Chicago Press.
9. Michotte, A. (1963) The Perception of Causality (trans. T. R. Miles & E. Miles). New York: Basic Books.
10. Pearl, L., & Steyvers, M. (2010). Identifying Emotions, Intentions, and Attitudes in Text Using a Game with a Purpose, 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text.
11. Tremoulet, P., and Feldman, J. (2000) Perception of animacy from the motion of a single object. Perception 29 (2000):943-951.
12. Von Ahn, L., & Dabbish, L. (2008). Designing games with a purpose. Communications of the ACM, 51(8), 57.
13. Young, J. E., Igarashi, T., & Sharlin, E. (2008). Puppet master: Designing reactive character behavior by demonstration. 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (pp. 183-191).