Motion Chain: A Webcam Game for Crowdsourcing Gesture Collection

Ian Spiro
Courant Institute
New York University
New York, NY 10003 USA
ian@movement.nyu.edu

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract
This paper describes the development and preliminary design of a game with a purpose that attempts to build a corpus of useful and original videos of human motion. This content is intended for use in applications of machine learning and computer vision. The game, Motion Chain, encourages users to respond to text and video prompts by recording videos with a web camera. The game seeks to entertain not through an explicit achievement or point system but through the fun of performance and the discovery inherent in observing other players. This paper describes two specific forms of the game, Chains and Charades, and proposes future possibilities. The paper describes the phases of game design as well as implementation details then discusses an approach for evaluating the game's effectiveness.

Author Keywords
Games with a purpose; serious games; computer vision; crowdsourcing; gesture

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

General Terms
Design, Experimentation, Human Factors.

Motivation

The study of human motion is the underlying focus of the Movement Lab's work in computer vision. The traditional approach in this field involves building complex statistical models to automate the extraction of high level information from pixel data alone, or informally, to make computers that can see. Working in this area, we are constantly reminded of the incredible power and flexibility of the human perceptual system, as contrasted to the flaky performance of computer vision. Face detection is one example of a mature computer vision technology that works well; yet we can barely take the next step, to accurately identify and parse bodies. This means we are nowhere near what might be considered the holy grail of computer vision: to take as input any video clip and output a textual summary. We cannot reliably tag all objects and actors in a clip, let alone describe the relations between them and the relevant verbs in a video.

The vision community is working to develop ever more sophisticated algorithms to push the accuracy of automated visual tasks. One general class of approaches, called machine learning, takes examples and counterexamples which are fed into a model that "learns" such that it gets the right answer, described in [1]. At first, the model performs poorly, no better than chance; but with each additional example, accuracy may improve. The learning is successful when generalization is achieved, whereby never-before-seen input images can be correctly identified. A key ingredient for the vast majority of machine learning systems is the training data. And yet the research community tends to focus on the development of new models instead of data collection, which is treated as an abstraction. This project is an attempt to address the need for large datasets to drive machine learning solutions and to do so in the context of a game.

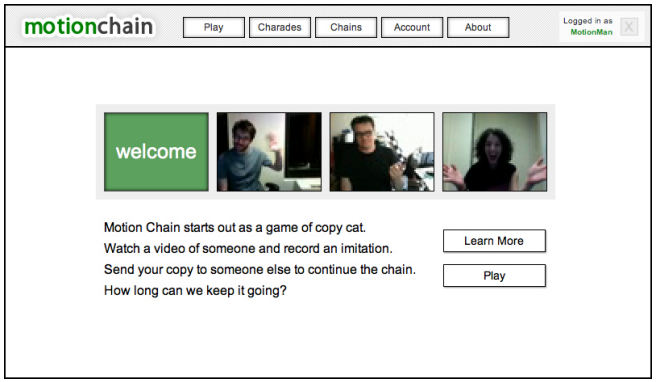

Figure 1: The Motion Chain website, located at http://www.motionchain.com.

Related Work

One aspect of modern society, described by Clay Shirky in [8], is an overall increase in free time and more ways for people to spend it. Shirky explains that people are now motivated in their free time to do creative and useful things, whereas the previous era focused on the consumption of media. Examples include the public creation of Wikipedia, a collaborative effort to solve protein folding problems as in [3], and the detailed annotation of images and videos in [7]. These are examples of web-based crowdsourcing, wherein unpaid Internet users collaborate to achieve something good for society. More examples and a taxonomy of crowdsourcing can be found in [5].

The concept of Games with a Purpose, or GWAP, was pioneered by Luis von Ahn and described in [10]. The first GWAP, called The ESP Game, had users label images through a website. By adding competitive and collaborative elements to the game, people had fun in the course of performing a task that might otherwise be seen

Figure 2: A sample chain showing several players making a scratching gesture.

as work. The users were not paid for their efforts but elected to participate freely because the experience was fun and therefore intrinsically rewarding. In the case of The ESP Game, users correctly tagged millions of images which later helped power Google's image search.

Most crowdsourcing to date has involved the generation of labels or other text by users, though examples of more media-oriented systems exist. In [4], users were paid to draw pictures of sheep for an art experiment, while [6] had volunteers trace over frames of a recording to effectively rotoscope an entire music video. A video project found at [2] invited users to submit imitations of frames from a recorded video by using their web cameras, then combined these frames to create a unique visual experience. Our lab demonstrated in [9] that the resulting dataset could be used for novel machine learning applications.

Design Concept

The present work is an attempt to extend the scope of GWAPs to a richer class of user inputs. Typical GWAPs are used to obtain sparse information such as image labels. The input is rich, a collection of pixels roughly a megabyte in size, whereas the output is sparse, a short string smaller than a kilobyte. Motion Chain is a GWAP that motivates users to produce megabytes of rich output in the form of video.

The basic setup of Motion Chain is to let users interact with each other via short video clips and text. This is a subset of what can be done already on YouTube, but YouTube lacks the constraints or context to suggest using it for such a game. To develop my concept, I looked to parlor games of the Victorian era for inspiration. Such games rarely involve equipment or complex rules and gameplay is typically informal, with a focus on fun and creativity instead of points. Two such parlor games, Telephone and Charades, are directly adapted here.

Chains

In Telephone, one player whispers a phrase to someone else, who in turn, passes the message on. After several iterations, the phrase can morph into something completely different. Motion Chain takes this concept to the video domain. Players watch a short video clip of a person and attempt to copy it. The player can practice any number of times before recording and uploading a response. Next, the player is encouraged to send her clip to a friend, as specified by an email address. The friend receives an email encouraging him to play and he can, in turn, copy and send it on. Users are only allowed to see the full series of clips after submitting to that chain. The fun of the game lies in the discovery of how a gesture morphs over time. Another element of fun, a result of the web-based nature of the game, comes from discovering the other people who have participated in a chain, whether friends, acquaintances, or strangers.

Charades

This classic game asks players to convey words and phrases through mummery. Players take turns acting and guessing, which is made fun when hard prompts require lateral thinking and cleverness, both to act out and guess. A connection is formed between actor and guesser, as they have managed to communicate despite the no talking constraint. This web-based interpretation of Charades allows users to record clips and submit them to a public area of the site. Other players can see submitted charades and type in a guess, which also becomes public. So as not to spoil the answer, a player can't see the intended answer or other players' answers until she has submitted her own guess, which further motivates participation.

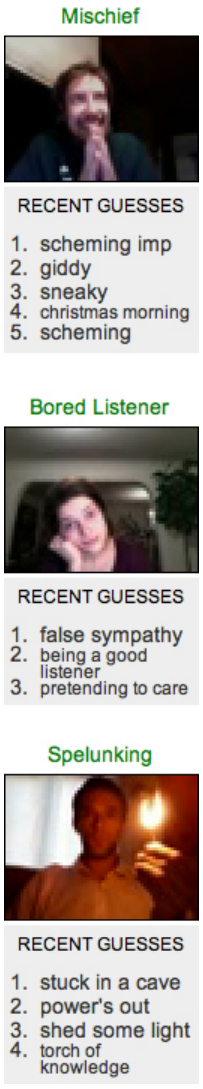

Figure 3: Some sample charades and recent guesses.

Design Process

The design process began with the development of a sandbox of code including form elements and Flash video units. I did not have a concrete idea of the gameplay but knew it would involve recording, playback, and text input. Once these components were complete, I could more rapidly prototype combinations in PHP and HTML and do informal user-testing on colleagues. I presented several prototypes at weekly lab meetings to see how others reacted and to prompt discussion. Once it was possible to record, I demonstrated it and had others try it. The most obvious thing to record was exactly what I had done, a clap. Watching several clap videos side by side was immediately compelling, and suggested the idea of a copying game as a starting point.

The second phase of development was to combine sandbox components and a database layer to create Chains. To launch the alpha test, I selected 40 friends and colleagues and invited them to copy a variety of arbitrary gestures I happened to produce during the testing of the recording widget. Within days, the alpha testers collectively produced over 300 short video clips.

The third phase involved more serious engineering, such as the addition of proper user logins, web 2.0 features like voting, invites, reporting of inappropriate content, and the development of a points system. The intention of the points system is not to make the game more competitive but to help steer user behavior. If all users playing Chains want to create new chains but never want to copy other players, the game will not work. With the point infrastructure, I can reward players with a point for a copy, and charge some number of points to start a new chain. Likewise, I can offer points to encourage voting.

The nature of Motion Chain is collaborative and social so I recognize that the project will only succeed if it reaches some critical mass of users. Therefore current development is centered around social media integration and general promotion of the site. The aforementioned point system is potentially inhibitory to play, so until critical mass is reached, it is simply disabled. Another current focus of development is to collect a large set of starter content, such as example charades and chains. I can curate this content with the hope of creating a good experience for new users 'out of the box.'

Figure 4: The recording interface. The previous recording is shown to the left of the live camera input.

Implementation

Motion Chain takes the form of a website, located at http://www.motionchain.com. The site was built on LAMP: Linux, Apache, MySQL, and PHP. These layers interact to route data between users and a database that tracks game state. Video files are stored remotely on the server, uploaded via a recording widget built in Flash. PHP acts as the glue to bind web server to database and implement most of the control logic. PHP is both very flexible as a language and potentially hard to manage

from the perspective of software engineering. To facilitate rapid prototyping with a strong development paradigm, I used a PHP framework called CodeIgniter. This simplified many coding tasks, such as the implementation of user accounts and password management. The web server for Motion Chain is located in the cloud, hosted on Amazon's EC2 service. The advantage of this service is the abstraction of hardware such that one can swap out one server for another with just a few clicks, simplifying upgrades if and when web traffic jumps.

The Flash recording widget required significant development. Flash remains the only way to reliably capture video from users on a variety of (non-mobile) platforms. The standard Flash solution for recording involves a remote media server and an adaptive approach to bandwidth allocation. For users with slow connections, the video quality goes down to prevent lag. For this project, upload time is less important than video quality so frames are captured to memory in the Flash plugin then uploaded to the server after recording is complete. Frames are uploaded individually as compressed JPEGs then stitched into a movie on the server. After making many videos with the widget, I realized the need for one special recording feature. When a user appears in a webcam preview on screen, it is natural for him to look at himself instead of the camera. Thus the user tends not to make eye contact with future viewers. The recording widget remedies this by moving the preview window up to the top-middle of the screen, closer to the most typical camera location, resulting in improved eye contact.

It's worth noting that the use of Flash, and thus personal computers (as opposed to mobile devices), results in a distinct and useful constraint on input. In this setup, the camera is nearly always fixed. This fixed camera helps many computer vision tasks, since background subtraction can be employed and the video will tend to be less blurry than a similar video recorded on a mobile device.

Purpose

To use machine learning in service of motion-oriented computer vision tasks, we require example videos of people and corresponding text labels. If we had a thousand short videos of people performing a particular gesture, such as clapping, we would have a good chance of training a state-of-the-art clap detector. Previously, researchers might have paid hundreds of students to physically come to a lab to be recorded for such a purpose. Motion Chain has the potential to produce the same set of data, for free and in a short period of time. The games described above, Chains and Charades, are designed for the purpose of fun more than for science. I first have to obtain critical mass before I can get more specific with data collection; if I achieve a steady user base, I can introduce game prompts that are less immediately fun but more useful to our research agenda. The game could mix these prompts in with more fun ones, or try appealing to user altruism by explaining that certain prompts support scientific research.

Assessment

Our system needs to be nominally functional so as to facilitate various game scenarios, but fun itself is what users want, as opposed to a maximally efficient interface. With basic recording and playback in place, I can focus on developing the user experience and game scenarios. I recognize there could be many compelling gameplay modes beyond the two described above. The instruction for Chains is "copy this motion" but that string can be changed to anything else. I hope new variants will emerge over time, as more users get deeper into the game. In the meantime, I can focus on producing and finding the most

interesting videos for Chains and Charades. With the built-in logging tools, I can make estimates about the experience of users' playing habits, to study which game types and particular videos get the most views or responses and what patterns of sharing emerge. One priority is to study and increase the rate of obtaining new players. I can use word of mouth, social media, and online advertising to promote the site and recruit potential players but can't guarantee they'll stick around for more than a few seconds. Given the dynamic, data-driven nature of Motion Chain, it's possible to adapt the control logic to create A/B testing scenarios. When a user first comes to the site as a guest, she is shown sample chains and charades. I want to present the particular set of example videos that makes her most likely to take the next step and sign up for an account. Randomly selecting different sets of videos for different users can be correlated to the conversion rate for each group.

Discussion

In this paper, I have described a new direction for Games with a Purpose that takes advantage of web-enabled social media, broadband connections, and the ubiquity of cameras. Using web cameras as an input for GWAP brings a new intimacy to games that could go deeper than interactions between icon-based avatars. Motion Chain is inspired by old parlor games, and yet has the potential to be something completely new. While the web creates an experience that lacks the immediacy of in-person gaming, it can produce a shareable trace of gameplay and facilitate play between people separated in space and time. If the experience becomes so compelling that people play voluntarily and invite friends to join, Motion Chain will be a platform for creating great datasets to aid computer vision research.

Acknowledgements

I would like to thank Christoph Bregler, Graham Taylor, and George Williams for technical assistance and many helpful discussions. Many thanks to the alpha testers who tested and invented charades: Phil Dhingra, Raphael Holmes, Kiefer Katovich, Carrie Kemper, Doug Kenter, Devon Sherman, and Molly Tanenbaum.

References

[1] C. Bishop. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[2] C-Mon and Kypski. One frame of fame. http://oneframeoffame.com/.

[3] F. Khatib, S. Cooper, M. Tyka, K. Xu, I. Makedon, Z. Popović, D. Baker, and F. Players. Algorithm discovery by protein folding game players. *Proceedings of the Nat'l Academy of Sciences*, 2011.

[4] A. Koblin. The sheep market. In *Proceeding of the seventh ACM conference on Creativity and cognition*, pages 451–452. ACM, 2009.

[5] T. Malone, R. Laubacher, and C. Dellarocas. Harnessing crowds: Mapping the genome of collective intelligence. *Report: CCI Working Paper*, 1, 2009.

[6] Milk, Chris. The Johnny Cash Project. http://www.thejohnnycashproject.com/.

[7] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[8] C. Shirky. *Cognitive surplus: Creativity and generosity in a connected age*. ePenguin, 2010.

[9] G. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2729–2736. IEEE, 2011.

[10] L. Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.