Active Capture and Folk Computing

Ana Ramírez
Garage Cinema Research
Group for User Interface Research
Computer Science Division
University of California at Berkeley
anar@cs.berkeley.edu

Marc Davis
Garage Cinema Research
School of Information Management and Systems
University of California at Berkeley
http://garage.sims.berkeley.edu
marc@sims.berkeley.edu

Abstract

The domains of folk computing applications touch on areas of interest to people around the world but are of pressing need to those in the developing world who often lack access to basic services and rights: especially health care, education, nutrition, and protection of human rights. In this paper we describe how a new paradigm for media capture, called Active Capture [3-5], and toolkit support for creating applications of this type work toward supporting the development of multimedia applications and interfaces for folk computing.

1. Introduction

In his article on folk computing, Ramesh Jain asks "Do we want to develop applications for every human on this planet or only for the 10 percent of privileged people in the developed world?" [1]. Our thinking about developing applications for every human on this planet, particularly for those in the developing world, resulted in the following taxonomy of the design space of such applications. The domains of folk computing applications touch on areas of interest to people around the world, but are of pressing need to those in the developing world who often lack access to basic services and rights: especially health care, education, nutrition, and protection of human rights. Underlying this range of application domains we see four basic patterns shaping the space of applications.

- $A \rightarrow B$ (information dispersion)
  Information traveling from a central source such as a government agency or corporation to individuals
- $A \leftarrow B$ (information gathering)
  Information traveling from individuals to a central destination
- $A \leftrightarrows B$ $A \leftrightarrows C$ $B \leftrightarrows C$ (messaging)
  Information traveling within networks of individuals
- $A \leftrightarrows B \circlearrowleft$ (diagnosis or training)
  Information traveling that changes the sender's knowledge about the user (diagnosis) or the user's knowledge about the information sent (training) based on iterative feedback from the user

In this paper we focus on interactive multimedia technology that could support the development and deployment of folk computing applications in the areas of diagnosis, training, and information gathering.

Jain describes the linguistic and literacy situation for the largest population yet to be touched by the information and communication technology (ICT) revolution: "Most people in developing countries don't speak or understand English and are illiterate in their native language" [1].. These constraints motivate the importance of the use of multimedia input and output as interfaces for the applications developed to reach the majority of people in the developing world.

The intersection of Jain's writing on folk computing [1] and Davis's writing on the future of multimedia computing [2] points toward a collection of important steps that need to be taken toward the goal of deploying ICT globally. 1) The roles of text and multimedia in ICT must be reversed. 2) Multimedia must become a first-class citizen on the web and in ICT. 3) The interfaces geared toward this population must be time, location, and person-aware. 4) The asymmetry between producing and consuming multimedia must reach a balance similar to that of producing and consuming text in the developed world. 5) Toolkit support for building interfaces that use multimedia as the main form of computer-human interaction and communication must be developed.

In this paper we describe how a new paradigm for media capture, called Active Capture [3-5], and toolkit support for creating applications of this type, work

toward supporting the development of multimedia applications for folk computing.

Among other important areas, multimedia interfaces have the opportunity to improve the access to preventative medical screening. Consider the following example. The doctors in small villages are not often trained to diagnose many of the common dermatological diseases in the developing world. Many of these debilitating diseases are caused by parasitic infections that can be more effectively treated or even cured if they are caught early enough. Telemedicine can currently enable doctors in villages to take digital images of skin conditions and send them via the internet to doctors in the major cities who have the expertise to diagnose common skin diseases (e.g., leishmaniasis).. The doctors in the small villages currently have to be trained to take the images necessary to diagnose the various skin diseases. A system in the paradigm of Active Capture could automate this process, eliminating the necessity of training doctors in taking the appropriate digital images, standardizing the images taken, and thereby enabling more routine, private, and cost-effective preventative medical screening. The system would interact with the patient, asking her to position parts of her body in front of the camera in order to capture the necessary images or footage.

2. Active Capture

"Active Capture" is a new paradigm in multimedia that brings together capture, interaction, and processing and exists in the intersection of these three capabilities (See Figure 1). Current textual interfaces largely exclude media capture and exist at the intersection of interaction and processing. In order to incorporate capture into an interaction without making the processing impossible, the interaction must be designed in such a way as to be able to leverage and simplify context from the interaction. For example, if the system needs to get a shot of the patient's face, it can interact with the patient to get them to face the camera and use simple, robust parsers (such as an eye finder) to aid in the contextualized capture, interaction, and processing.

We currently have two working interaction scenarios that demonstrate this new capture, interaction, and processing paradigm. These scenarios were designed and built in the context of automated video capture applications for automatic video customization and personalization, but the interaction techniques developed could also be fruitfully applied to the creation of multimedia interfaces for folk computing applications.

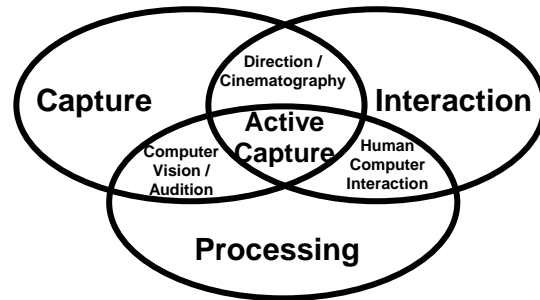The more complex of the two interaction scenarios

Figure 1. Active Capture integrates capture, processing, and interaction

engages the user in an interactive feedback loop to get a shot of her turning her head to look at the camera (See Figure 2). This application uses simple motion and eye finding technology in an interactive feedback loop to ensure a shot of the user turning her head. The system checks to make sure that she is not looking at the camera at the beginning of the turn, that her head is moving during the turn, and that she is looking at the camera at the end of the turn. These simple checks are enough to ensure a head turn because they are combined with the interactive feedback loop and make use of the context inferred from the instructions given and the physical actions performed by the user.

Once the shots of the user screaming (from the simpler interaction scenario) and turning her head are captured, the system automatically keys the shots, spatially and temporally scales them, and adds lighting layers to enable them to be automatically inserted in pre-made templates of commercials or movie trailers (e.g., MCI, 7Up, Godzilla, Terminator 2).

One of the advantages of Active Capture is the increased symmetry between the creation and consumption of multimedia. By using human-computer interaction design to interactively simplify the context of capture and thereby enabling computer vision and audition algorithms to work robustly, Active Capture radically reduces the skill and effort needed to create high quality reusable media assets. The ease of multimedia asset creation in Active Capture could also be beneficial for information gathering and messaging applications for folk computing.

3. Active Capture Automation Language

In the dermatological medical screening example above, an Active Capture system would attempt to get frontal and side face shots of the patient. The system may ask the patient to "Please look at the camera" (or use nonverbal cues to get the patient to face a certain direction), but the patient may not know where the

camera is, may not understand the request or stimulus, or may have something obscuring her eyes such as a scarf or sunglasses. In trying to determine whether the patient is looking at the camera, the system looks for the patient's eyes, but if the patient is not facing the camera or has her eyes obscured, the system cannot find them. This means there is an ambiguity between the system's understanding of what the patient is doing and what the patient is actually doing. In order to remedy this situation, the system must try again; for example, the system might say "I can't see your eyes, perhaps you have a head scarf on or are wearing sunglasses. Please remove any scarves or sunglasses and look at the camera." This corrective interaction technique is known as *mediation*. Mediation techniques are used to resolve ambiguity in systems when there exists an apparent discrepancy between the system's model of the state of the world and the actual state of the world, in this case, the physical orientation of the patient.

In order to bring capture and mediation into the interface, Active Capture applications must incorporate the notion of time. Mediation and the notion of time are two of the most challenging aspects to building successful end user applications with multimedia I/O.

There is currently no language or library support for building Active Capture applications. Many tools provide timing support (e.g., Director, Flash, and

toolkit [6], but to our knowledge there does not exist a language or toolkit that supports mediation, timing information, and feedback loops for interaction using multimedia input, output, and analysis.

The design cycle for developing an Active Capture scenario begins with a brainstorming session to come up with the required steps, and how the system will check for ambiguity between its understanding of, and the actual state of the user. For example, the system may need to take footage of the front and side of the patient's face. In this case the system might use an eye detector or face detector to make sure to get an image of her face.

The next step in the brainstorming session is to come up with what to do if the patient is not in a position to get the appropriate shots, i.e., what mediation techniques to use. This step requires brainstorming about what could be causing the ambiguity, (e.g., the patient is looking away from the camera.)

The next step is implementation. Currently, the entire interaction with mediation and links to media analysis are implemented in C++ using a finite state machine and iteratively tested. The testing often reveals timing mistakes (how long to wait for the participant to react to a command before trying a mediation technique) and missing mediation paths (there is ambiguity in the system, but the system doesn't know about it).

Correcting the timing directly and adding mediation paths to the state machine is currently tedious and time consuming, extending the length of the design cycle. Based on experience from the implemented Active Capture applications, we have developed a design for toolkit support for authoring Active Capture applications. We call this toolkit and language ACAL (Active Capture Automation Language). ACAL aims to
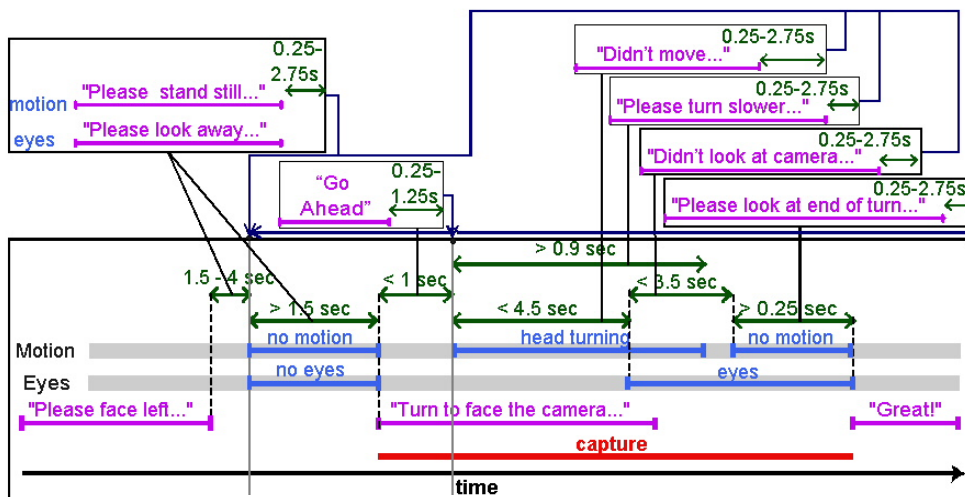

Figure 2. Visualization of "Head Turn" Active Capture scenario. The application directs the guest to turn her head to look at the camera by ensuring a set of constraints are satisfied through a control process with interactive feedback

SMIL), others provide support for feedback loops or loops in general (e.g., Authorware, Java, C++), and still others provide access to media analysis modules (e.g., Intel Open CV, VideoScript, Papier-Mâché). There is also recent work on mediation support in a

shorten the time it takes to complete one cycle in an iterative design process and make it easier to incorporate known types of mediation and mediation techniques [7].

Most design cycles begin with a sketch or rough idea of the application at a high level. The complexity of combining timing and mediation into an interactive feedback loop without hiding the main goal of the application makes this step in the Active Capture application design cycle particularly difficult. Figure 2 depicts our solution to this step. The main timeline depicts the goal of the application and the steps necessary to successfully complete the interaction. The smaller timelines depict the mediation feedback loops. The timing annotations describe how long to wait for a correct state before mediating.

The design of ACAL will support the brainstorming and rapid prototyping process by suggesting possible mediation techniques to use. The implementation will provide special constructs allowing mediation, capture, and timing to be expressed together and mediation templates based on typical and proven mediation techniques. The use of special templates in a toolkit restricts the range of applications it is possible to build with the toolkit, possibly restricting new innovative approaches, but it also eliminates many poorly designed applications. As is common with toolkit design, ACAL, by design, restricts the types of applications that can be implemented as compared to using a generic programming language such as C++.

An example mediation template would be one for the simplest form of mediation, repetition. The parameters to the mediation template include what conditions must be met and for how long, how long to wait before repeating the mediation technique, and what to do on the first, second, *etc.* times this condition is not satisfied. This template might look something like this: If the patient's eyes cannot be found for 10 seconds after 15 seconds, ask her or demonstrate to her how to take off any glasses or scarf she may be wearing, if that doesn't work, describe or indicate where the camera is and ask her to look at it again while still making sure she is not wearing any glasses or scarf.

4. Contextual Interviews

As we developed strategies by which to improve our Active Capture applications, we realized that a more thorough investigation of the design space would benefit not only the design of our current Active Capture scenarios, but that of *any application in which a computer system could be used to automatically capture, analyze and provide corrective feedback to physical human action.* In an effort to inform the design of Active Capture scenarios and design patterns for use in computer-human interaction, we conducted a series of contextual interviews with human-human interaction experts [7].

Some of the people interviewed include a film director, a golf instructor, and a 911 emergency operator. These interviews revealed successful direction and mediation techniques used by experts in human-human interaction under different circumstances. For example, the 911 operator is an expert in communicating and getting feedback over a low bandwidth connection (the phone). Our interviews uncovered numerous strategies employed by experts to guide specific human actions including different design strategies, direction and feedback strategies, and mediation strategies [7]. The results of the contextual interviews and the resulting design space will provide the structure of the mediation templates in ACAL, and guide which templates to provide.

5. Conclusion

As part of our future work, we plan to formulate the results from the contextual interviews—design strategies and design space analysis—into structures and guidelines in ACAL. Applications in the Active Capture paradigm aided by the ACAL toolkit under development will enable the roles of text and multimedia in ICT to reverse as well as helping multimedia to become a first-class citizen on the web by decreasing the asymmetry between producing and consuming multimedia. Active Capture and ACAL will aid the development of folk computing applications by enabling the efficient and reusable authoring of multimedia and multimodal interfaces that leverage the knowledge of human-human direction and mediation strategies.

7. References

[1] R. Jain, "Folk Computing," *IEEE Multimedia*, vol. 8, pp. 96, 2002.

[2] M. Davis, "Garage Cinema and the Future of Media Technology," *Communications of the ACM (50th Anniversary Edition)*, vol. 40, pp. 42-48, 1997.

[3] M. Davis, "Active Capture: Automatic Direction for Automatic Movies (Video)," ACM Multimedia 2003, Berkeley, CA, 2003.

[4] M. Davis, J. Heer, and A. Ramirez, "Active Capture: Automatic Direction for Automatic Movies (Demonstration Description)," ACM Multimedia 2003, Berkeley, CA, 2003.

[5] M. Davis, "Active Capture: Integrating Human-Computer Interaction and Computer Vision/Audition to Automate Media Capture," ICME 2003, Baltimore, MD, 2003.

[6] J. Mankoff, S. E. Hudson, and G. D. Abowd, "Providing Integrated Toolkit-Level Support for Ambiguity in Recognition-Based Interfaces, SIGCHI 2000, The Hague, The Netherlands, 2000.

[7] J. Heer, N. Good, A. Ramirez, J. Mankoff, and M. Davis, "Presiding Over Accidents: System Mediation of Human Action," SIGCHI 2004, Vienna, Austria, Forthcoming 2004.