Auto-classifying Salient Content In Video

Alistair Morrison, Paul Tennent, John Williamson, Matthew Chalmers
Department of Computing Science, University of Glasgow, Glasgow, G12 8RZ
{morrisaj, pt, jhw, matthew} @ dcs.gla.ac.uk

In this position paper we present a procedure with which large volumes of video data can be automatically processed to extract only that which is salient given the current context of analysis. This technique is presented as an extension to *Replayer*: a software toolkit for use in the development cycle of mobile applications. By augmenting the cameras used to capture a mobile experiment we are able to establish both a location and heading for each camera, and thus generate a field of vision. Locations of each trial participant are also recorded and compared against camera visual fields to determine which periods of user activity have been recorded. A report of initial testing is given, whereby the technique's use is demonstrated on a trivial application.

Video, Evaluation, Data Analysis, Data Capture.

1. INTRODUCTION

When dealing with qualitative analysis of large sets of data, the sheer volume of recorded information can make detailed analysis a time consuming and labour-intensive task. In the case of a hypothesis-driven experiment, much of the data may not be relevant, so examining this data looking for periods of interest can waste a lot of an analyst's time; considering the specific case of video data, an analyst may be required to watch thousands of hours of footage in search of evidence. Here we present a new technique, implemented as part of *Replayer* [1], designed specifically to aid in this pursuit. Replayer is an evaluation tool for the combined analysis of video data and recorded system logs. The toolkit already temporally synchronises quantitative log data with mixed media recordings, allowing analysts to pinpoint specific events in the timeline of an experiment and jump to the periods of video showing the timeframes at which these events occurred. However, there is no guarantee that all or any of the video streams captured at these instants will have captured the event of interest. This is a particular issue in the evaluation of multi-user mobile applications, where a roaming camera will struggle to follow all the participants' movements.

The technique presented here augments video recordings with the location and heading of each camera. With this information Replayer is able to inform an analyst which events are likely to have been captured, and automatically tailor video playback to show only these periods. A further application of this technique is identifying all the periods of video footage that capture a particular person, as he or she moves in and out of the visual fields of multiple cameras. In this paper we will explain the implementation of this system, and use a trivial mobile application to demonstrate how effective this facility can be.

2. RELATED WORK

Previous work with GPS enabled cameras includes RealityFlythrough by McCurdy et al. [2], which related to placing images and video streams from camera-enabled mobile phones in 3D space to create an amalgamated panoramic scenes. GeoVector (www.geovector.com) have produced a number of applications which dynamically deliver content to mobile devices based on a GPS reference and a heading from an electronic compass. Beeharee and Steed [3] have used occlusion information to filter dynamically generated content provided based on users' locations, removing that which cannot be seen due to visual occlusion.

3. OVERVIEW OF REPLAYER

The Replayer toolkit [1] has been developed to support the evaluation and development cycle of mobile computing systems. It can be used in usability testing or by computing or social scientists in studies into the use of mobile applications. Logged systemic data, video and audio recordings can be examined, as can textual notes recorded both in synchrony with the trial and post hoc. Mixing quantitative and qualitative analysis techniques, Replayer is a powerful tool for examining data recorded about a system, providing many different techniques for synchronising, visualising and understanding the data. Each visualisation component is linked to every other to support brushing [4]; any selection made in one immediately makes a corresponding selection in another. For example we may have a

FIGURE 1: Replayer analysing heterogeneous data. A period of the graphed accelerometer logs (left) has been selected, highlighting in green the corresponding time in the video timeline (bottom) and jumping the video to the appropriate section.

graph showing all the system events for a given participant on a timeline, and a map showing a spatial distribution of those events. Selecting one event on the timeline would highlight the location on the map at which the event occurred. This is also applied to video data – selecting the event on the timeline would also show any video captured at that time by each camera, jumping to exactly the correct frame in each recording. Figure 1 shows an example of Replayer visualising heterogeneous data. This paper describes a refinement of this technique specifically related to video.
The following discussions involve a number of different roles in using and evaluating applications, and it is worth clarifying vocabulary at this stage. Replayer is a desktop tool for data analysis. It is intended that Replayer be used by *analysts* looking into the results of user trials of mobile applications. *Participants* in these trials will have their activity logged by code within the mobile application and be filmed on video by *camera operators* (often part of the analysis team).

4. CAPTURING DATA

When performing an evaluation of a mobile system, the capturing of video becomes a challenge. In traditional lab-based experiments, one or two cameras would generally be able to record everything, and typically these cameras could be affixed to tripods and subsequently ignored while the experiment was captured. In a mobile experiment, it is common to use both fixed and roaming cameras. With a fixed camera, participants will move in and out of the camera's field of vision. A fixed camera at some distance may provide an overview of the entire experiment, but typically the range makes this less than ideal for detailed analysis of a participant's actions. A roaming camera is one carried by a dedicated camera-operator for this experiment. Generally the footage from roaming cameras is of better quality than that of fixed cameras as they are able to follow participants around; however in order to capture every participant's actions the ratio of camera-operators to experimental participants must be 1:1. This may not be logistically feasible, so it is likely that there will not be continuous video data for every participant.
The technique presented here offers three specific benefits.

4.1 Automatically find video of a logged event

Replayer allows us to visualise a timeline of all recorded events. By recording information on each camera's location and in which direction it was pointing, it can be established whether or not each of these events has been captured. On selecting an event of interest, Replayer is able to return only video streams in which that event was seen to take place.

4.2 Compiling all the video for a single participant

It is common in video analysis to use a more exploratory-driven approach to investigating a given dataset. In this form of analysis, an analyst is not searching for specific events, but rather closely examining a participant's activity across many hours of video. In this case, Replayer allows the analyst to select a single user to examine and can skip playback of the multiple streams of video to show only the periods where the participant of interest is in view.

4.2 Filtering out participants

With the current trend of privacy concerns, particularly when performing experiments with children and teenagers, it may be the case that some participants could withdraw their consent for the use of their video footage. In a similar manner to the last example, Replayer allows us to filter videos for periods containing such users, returning only video *excluding* them, and thus allows the presentation of only 'safe' data.

5. INITIAL TRIALS

In initial trials to demonstrate the effectiveness of this technique, we developed ColourLogger, a trivial application for mobile devices, running Microsoft's PocketPC 2003 framework, and equipped with a Global Positioning System (GPS) receiver. The application's interface showed three buttons, marked red, green and blue. Participants were asked to walk around a fairly small area looking around for objects of these colours and, on discovery of such an item, to press the appropriate button. The time of the each button press was recorded in a system log. Additionally, a continuous log was maintained of the participant's GPS position and the number of GPS satellites currently available. This latter value helps determine whether the GPS values are valid for a given time. In the experiment we had two participants, each running the device on a Hewlitt Packard iPaq hx2410, using a SysOnChip compact flash based GPS receiver.

5.1 Data Capture from the ColourLogger Experiment

As the aim of this experiment was to demonstrate the possibilities of the data extraction technique described in this paper, the actual ColourLogger application is minimal. As described above, the recorded data from the device was limited to the button events and location information. The video recorded in this experiment came from four cameras, three of which were in a fixed locations, and one which roamed around following the participants. The fixed cameras were in this case mobile phones, capturing video at 176x144 resolution. While this is not particularly high quality, it serves to demonstrate a technique by which many low cost cameras can be used to add to the variety of video streams for a given experiment. The roaming camera was a more traditional CCD-based digital video camera, augmented using a Hewlitt Packard iPaq 5550 interfaced to the MESH inertial sensing platform [5] which provides GPS tracking along with tri-axis accelerometer, gyroscope and magnetometer sensing capabilities. Data from these sensors are logged at 100Hz (1Hz for the GPS device). Onboard hardware filtering is applied to

the inertial readings, rolling off at around 20Hz. The device is attached to the base of the camera so that the position and orientation of the camera and sensing platform are linked. The physical location of the camera is logged via the GPS, while the orientation is obtained via the magnetometers and accelerometers. The magnetometers measure the yaw angle, and the accelerometer readings are used to estimate the roll and pitch from the effect of the Earth's gravitational field. Knowledge of the roll and pitch is used to correct for variations in the magnetic field as the device is tilted and thus obtain accurate yaw estimates. Standard strap down inertial sensing techniques are used to perform this tilt-compensation. The field-of-view of the camera is a given by a cone extending from the measured location of the camera along the yaw angle estimated from the combined accelerometer and magnetometer readings. This is used to estimate the potential visibility of targets. In order to get locations and bearings for the fixed cameras, this augmented camera was positioned next to them and an annotation was made in the logs, allowing for post hoc synchronisation.
Once complete, the data captured from the experiment consisted of the following: four video recordings; one log from the augmented camera showing location and bearing; one single-line log from each of the stationary cameras, showing location and bearing; one log from each participant showing location and another showing timestamps of button events. These data were subsequently read into the Replayer toolkit and synchronised using the QCCI [6] technique.

5.2 Establishing capture information

A novel component has been added to the Replayer toolkit to perform analysis across the augmented video data. Log files for each camera give timestamped location and bearing information. Lens width and range are also stated for each. Each participant's recorded button clicks and sampled locations are checked against the camera logs to assess which periods of participant behaviour have been captured. For each user log value, the last recorded position and bearing of each camera is checked and to calculate the field of vision. The location of the event is then checked to see if it is within one or more of these fields (see Figure 2).

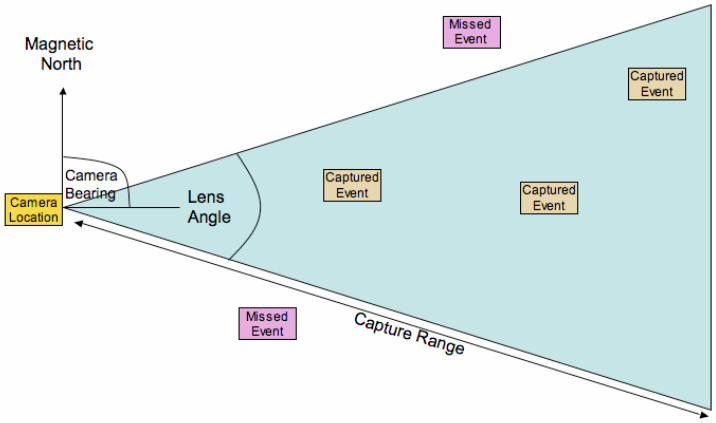

FIGURE 2: Spatial distribution of events in a mobile application experiment. A triangle is created for each camera showing the field of capture. Each event is tested to see if it is within one or more of these triangles.

5.3 Analysing the Data

Once accessible within the Replayer toolkit, several options for examining the data are available. A number of examples of use are demonstrated. The graph in Figure 3 shows logged events over time. The *x*-axis covers the time from the beginning until the end of the experiment. Glyphs are placed on the *y*-axis dependent on event type (in the trial application, each of the three button clicks) and are coloured by participant. An analyst may be particularly interested in one type of event. Existing Replayer functionality allows one row of this graph (one type of event) to be selected, which would instruct the video component to show only the corresponding time periods. Of course, there is no guarantee that the video recorded at these times will cover these events. This selection can now be further filtered, so that events uncaptured on video are coloured grey in this view. This shows the analyst exactly which logged data can be enhanced by the context provided by video footage.

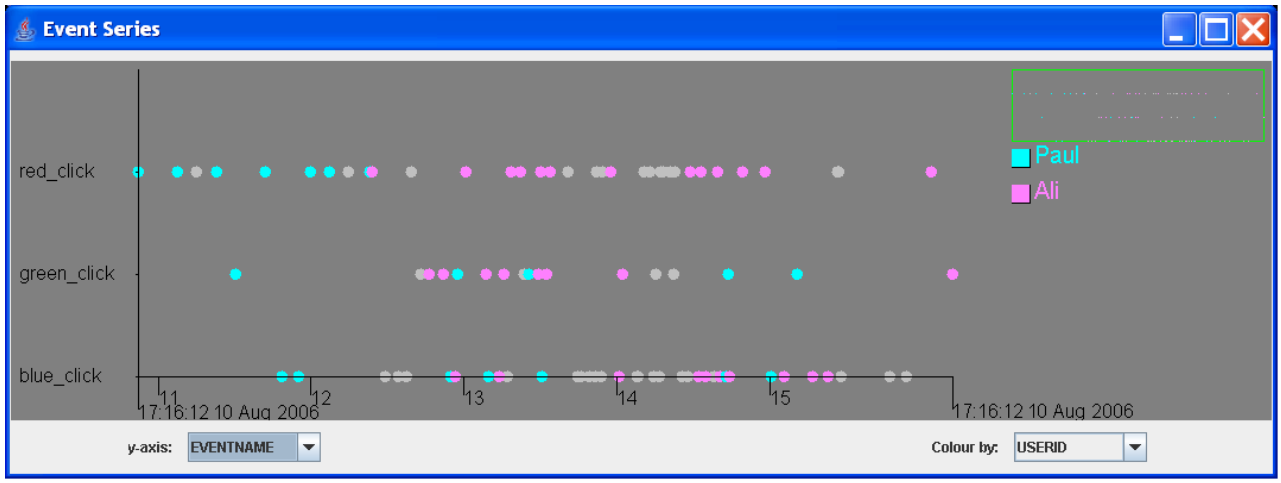

FIGURE 3: A timeline showing logged event occurrence over time (*x*-axis). Different events are positioned at different heights on the *y*-axis. Events are coloured by participant ID, and the events that were not captured on any cameras are greyed out.

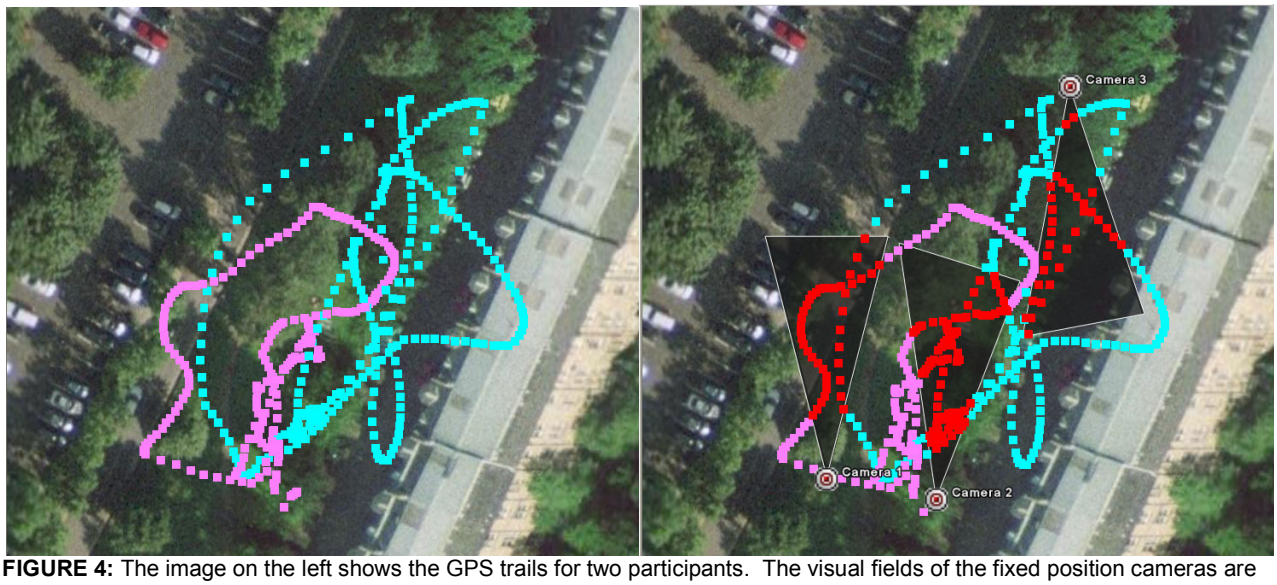

FIGURE 4: The image on the left shows the GPS trails for two participants. The visual fields of the fixed position cameras are added to the image on the right, and the objects captured by the cameras are highlighted in red.

Replayer shows spatial data by linking to Google Earth (http://earth.google.com/). The screenshot in Figure 4 shows logged GPS trails for two participants as they moved around the area in which the experiment took place. In the image on the right, having processed camera location and bearing information, the events that fall into each cameras' sights are highlighted in red. The fixed camera locations are also rendered on this screenshot, with their fields of view shown as semi-transparent triangles. From this view, it is easy to get an overview of how much of the participant activity has been captured, and the range at which each of the events was captured. Those at closer range are likely to be more clearly visible than those just on the periphery of a given camera's range. It would also be useful in re-positioning cameras for future experiments, to capture more data.

A third visualisation, shown in Figure 5, shows two streams of video footage and a timeline for each. An analyst has selected to see data from one of the two participants and the timelines have been automatically highlighted in green over the periods where the participant has been calculated to be in view. On playback, Replayer will skip the periods where the user is out of view of all cameras and show multiple streams concurrently in the timeframes where the participant was captured by more than one camera. This playback is synchronised with other components so that, for example, glyphs on event graphs are highlighted as they occur.

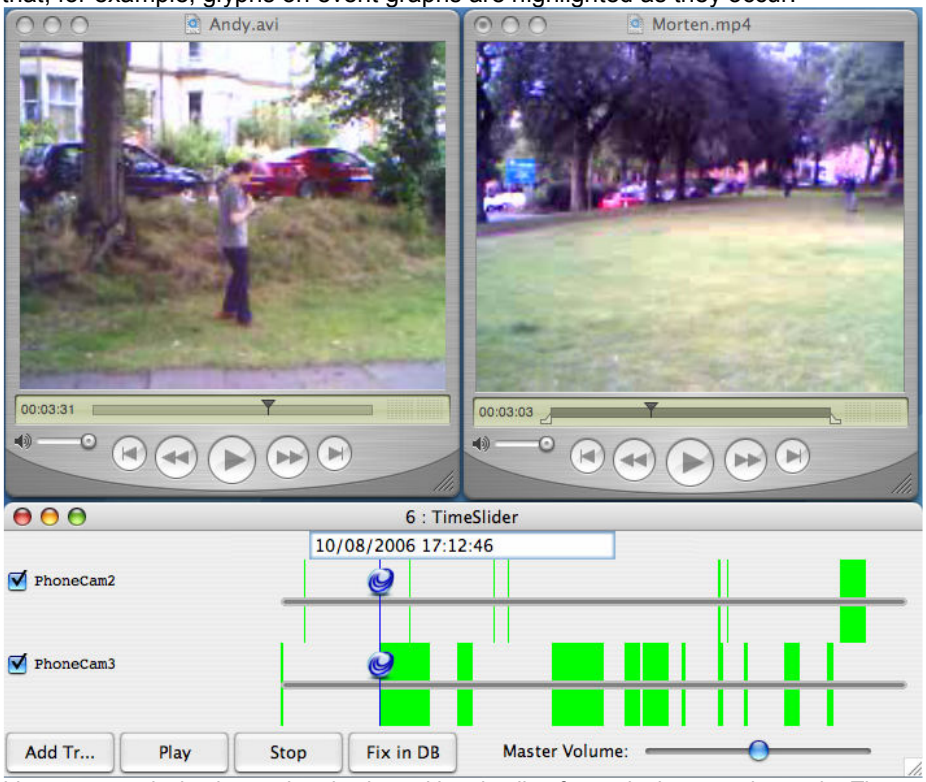

FIGURE 5: Two video streams playing in synchronisation, with a timeline for each shown underneath. The analyst has selected one trial participant, and the green highlighting on the timelines shows the periods in each of the videos where the user appears. Video playback can be set to skip periods where the participant does not appear in either video stream.

Of course the value of examining the data of such a trivial system is minimal. The aim is to demonstrate the potential of this technique rather than learn anything of value from this application. Work on this analysis is in the early stages. Further evaluation, comparing the visualisations with the video footage, would be required to ascertain the accuracy with which Replayer classifies participant capturing.

6. FUTURE WORK

As we further refine this technique one particular use we envisage is in the orchestration of a system trial. If the sort of data described here could be uploaded and analysed in real time it would be possible to show a histogram of the amount of activity captured for each participant. An orchestrator seeing inequality in this histogram can direct roaming cameras to concentrate on particular participants as required. Another area we aim to examine is that of occlusion. In its current incarnation this system does not take account of the fact that a wall or other occluding item may be between the camera and the participant. A feature being slowly integrated to Google Earth is a 3D representation of all the buildings in a given city; we hope to use this information to limit the fields of view from each camera to reflect this occlusion. Additionally, sensor information is available in 3D, so, in the case of roaming cameras, we also intend to allow analysts to quickly discard footage where the camera is pointing at the ground or sky.

7. CONCLUSION

We have presented a technique for analysts of data recorded in mobile application evaluations. Specific recorded user activity can be queried, to automatically skip large amounts of irrelevant video footage and focus on that which is salient. This reduction can include showing video in which a particular participant appears, or showing only that area of video in which a particular user event has been captured. We devised, implemented and recorded a trivial application to demonstrate how this method might be used, and showed some of the capabilities provided by this extension to the Replayer toolkit. Finally we indicated where we intend to take this work in the future. We believe this is a valuable addition to the already versatile Replayer toolkit and hope that this will be a technique picked up and used commonly in the evaluation of mobile computing systems.

ACKNOWLEDGEMENTS

We thank Morten Proschowsky, Andrew Crossan and Jody Johnston for assistance with running the experiment.

REFERENCES.

[1] A Morrison, P Tennent and M Chalmers (2006) *Coordinated Visualisation of Video and System Log Data*, in proceedings of 4th International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV2006), London, UK.
[2] N J McCurdy, J N Carlisle and W G Griswold (2005) *Harnessing Mobile Ubiquitous Video*, in proceedings of ACM Conference on Human Factors in Computing (CHI2005), Portland, Oregon, USA.
[3] A Beeharee and A Steed (2005) *Filtering Location-Based Information Using Visibility,* in Proceedings of Location- and Context-Awareness (LoCA 2005), Munich, Germany.
[4] R A Becker and W S Cleveland (1987) *Brushing scatterplots*, Technometrics, 29:127-142, 1987.
[5] S Hughes, I Oakley and S O'Modhrain (2004) *MESH: Supporting Mobile Multi-modal Interfaces*, in proceedings the Seventeenth Annual ACM Symposium on User Interface Software and Technology (UIST2004), Santa Fe, New Mexico, USA.
[6] P Tennent and M Chalmers (2005) *Recording and Understanding Mobile People and Mobile Technology*, in proceedings of the First International Conference on e-Social Science, Manchester, UK.