

# Using Location, Bearing and Motion Data to Filter Video and System Logs

Author details removed for anonymous review

**Abstract.** In evaluating and analysing a pervasive computing system, it is common to log system use and to create video recordings of users. A lot of data will often be generated, representing potentially long periods of user activity. We present a procedure to identify sections of such data that are salient given the current context of analysis; for example analysing the activity of a particular person among many trial participants recorded by multiple cameras. 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 model the field of view for each camera over time. Locations of trial participants are also recorded and compared against camera views, to determine which periods of user activity are likely to have been recorded in detail. Additionally the stability of a camera can be tracked and video can be subsequently filtered to exclude footage of unacceptable quality. These techniques are implemented in an extension to *Replayer*: a software toolkit for use in the development cycle of mobile applications. A report of initial testing is given, whereby the technique's use is demonstrated on a representative mobile application.

**Keywords:** Video, auto-classification, analysis toolkit, log synchronisation, visualisation.

## 1 Introduction

Evaluating and analysing a mobile application, especially one involving multiple participants and locations, can be a difficult task. Multiple observers may be needed, often recording multiple streams of video that complement system logs from the application running on multiple devices. The sheer volume of recorded information can make detailed analysis a time-consuming and labour-intensive task. This can be the case where an analyst employs an exploratory approach, for example using ethnographic techniques to reveal patterns of use, or when the analyst carries out a hypothesis-driven experiment. Much of the data may not be relevant or complete, and so examining all the data looking for periods of interest can consume much of an analyst's time; considering the specific case of video data, an analyst may be required to watch tens or even hundreds of hours of footage in search of evidence. Here we present a new technique designed specifically to aid in this pursuit. The technique is implemented as part of *Replayer* [12]: an evaluation tool for the combined analysis of video data and recorded system logs. *Replayer* already temporally synchronises quantitative log data with mixed media recordings, so that recorded system and interaction events in an experiment can be analysed within the same tool. Analysts

can therefore 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, or if they have, whether the quality of the video will be acceptable. This is a particular issue in the evaluation of multi-user mobile applications, where a roaming camera will likely struggle to capture all the participants' movements.

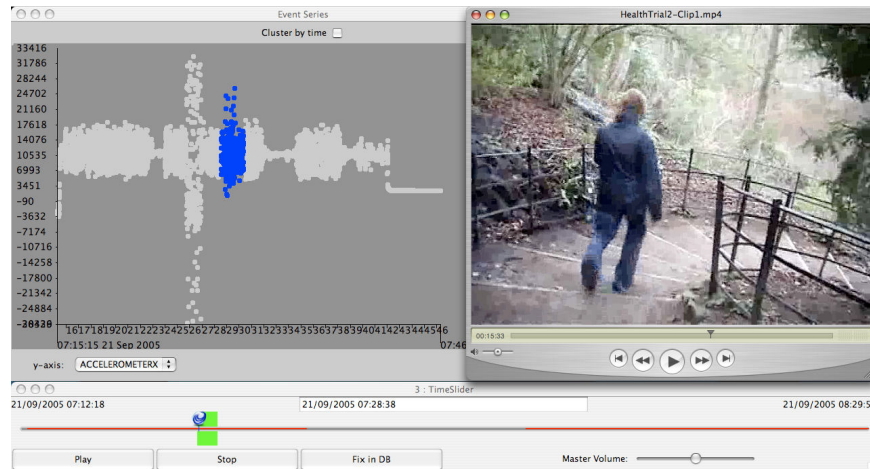
The technique presented here augments video recordings with the location and heading of each camera as well as data about the cameras' motion. With this information, Replayer is able to inform an analyst on which events are likely to have been captured, and automatically tailor video playback to show only these periods. A further application of this technique is in 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. Finally, this technique allows the system to automatically discard all the video in which the camera is shaking excessively – a common problem when recording mobile systems as roaming camera operators are often forced to run. In this paper we explain the implementation of these new techniques, and use a mobile application to demonstrate how effective these facilities can be.

The following section provides an overview of Replayer: the software toolkit to which the described techniques are an extension. This is followed by a description of related work in the field. The four main benefits offered by the ability to classify data in this manner are outlined in Section 4, before the technical details of the process are described in Section 5. An experimental trial is described in Section 6 and the results are analysed. This is followed by a description of future directions for this work and finally our conclusions.

## **2. Overview of Replayer**

The Replayer toolkit [12] 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. As in the simple case shown in Figure 1, logged system data as well as video and audio recordings can be examined in a synchronised manner, along with textual notes recorded either during a system trial or post-hoc. In the example of the figure, selecting an area of a graph of accelerometer readings also selects video corresponding to the times of those readings.

The work of qualitative methods of video analysis, such as ethnography, is time-consuming but affords rich detail of the user experience of systems – detail that may be unavailable from quantitative data such as system logs. In a complementary way, quantitative methods allow for rapid indexing and for overviews such as statistical distributions and visualisations, but may abstract away from the subjective experience of users. The analytic practices associated with these complementary approaches are often carried out separately, or even in opposition to each other, but Replayer is intended to allow for tighter coupling and integration of these different forms of data and these different forms of analysis.



**Fig. 1.** A simple case of analysing heterogeneous data in Replayer. The analyst has selected a period of time in graphed accelerometer logs (top left), which triggers Replayer to highlight in green the corresponding section of the video timeline (bottom) and to cue the video of that section (top right).

By mixing quantitative and qualitative analysis techniques, Replayer is a powerful tool for examining data recorded about a system and its use, providing many different techniques for synchronising, visualising and understanding the data. The example of Figure 1 shows how interactions performed in one component are reflected in another. In fact, each visualisation component in Replayer is linked to every other in this way to support brushing [3]; any selection made in one immediately makes a corresponding selection in another. For example, we may have a 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. As shown above, this is also applied to video data – selecting the event on the timeline also shows any video captured at that time by each camera, jumping to approximately the correct frame in each recording. Similarly, in the example in Figure 1, selecting a section video timeline (bottom of the figure) would highlight the accelerometer data logged during the time period of the section.

The techniques described in this paper extend the synergistic combination of quantitative and qualitative analysis that Replayer is based on. Faced with large volumes of video data, we have chosen to use quantitative data about location and motion to assist existing qualitative analysis techniques, such as ethnography, rather than replace them. In this paper, we outline techniques that use quantitative location and motion data to focus on particular sections of video that might benefit from detailed qualitative analysis, and to allow those doing qualitative analysis to better relate their usual material – video – to visualisations of quantitative data that are made more usable by their tight coupling in interaction. Findings and results from our user trials of Replayer, intended to help people carrying out their own user trials in this

hybrid or synergistic way, are the subject of forthcoming work. Here, we focus on a novel extension to Replayer that itself shows useful synergy in approaches.

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).

### 3. Related Work

Replayer grew out of work on evaluating multi-user ubiquitous computing systems [6], in particular mobile multiplayer games. More traditional tools for analysis tend to focus on parts of the analysis task, such as Transana ([www.transana.org](http://www.transana.org)) for transcribing audio and video. Others are limited in the tools for analysis and visualisation that they support. For example, The Observer ([www.noldus.com](http://www.noldus.com)) allows synchronised playback of up to four video streams alongside sensor data (e.g. physiological data), but relies on simple graphs and tables of quantitative data. Replayer allows more complex interactive filtering and selection of data, and also integrates statistical tools such as mutual information, and sophisticated visualisation tools such as force-based layouts of multidimensional data. DR.ReplayTool [8] is beginning to move beyond systems such as The Observer in its support for complex categorisation schemes and shared repositories of both raw data and analytic results, and also supports a timeline-based tool to display video, but its tools for interaction and visualisation are currently very limited. The VizWear system [10] allows interaction with multiple synchronised video streams, along with event data logged from a wearable computing system. Navigation and viewing is basically linear; sophisticated searching or ordering operations are not implemented.

An early system synchronising video was developed by Badre et al. [2] which used a video tape/CD based system and made use of captured event streams to synchronise time-stamped events with the time-code on a video. This was designed for static lab based usability testing and was limited by the technology available at the time, but follows a similar technique to that which underpins Replayer's logs-video synchronisation.

Our system takes advantage of the trend towards positioning systems such as GPS being integrated with devices such as mobile phones and cameras. Previous work with GPS-enabled cameras includes RealityFlythrough by McCurdy et al. [11], which placed images and video streams from camera-enabled mobile phones in 3D space to create an amalgamated panoramic scene. Beeharee and Steed [4] have used occlusion information to filter dynamically generated content provided based on users' locations, removing that which cannot be seen due to visual occlusion. Using hardware similar to that described in this paper, Sawahata et al [13] describe a video capture system augmented with an inertial sensing platform and a GPS unit. The time-series from the accelerometers and gyroscopes is classified with a hidden Markov model; the system can distinguish walking, running and standing behaviour. The video

stream is indexed with these activity levels, and the physical location at which they occurred. Conversely, Aiwaza et. al. [1] use a combination of shot and pan detection (from the video stream itself) and sensed brain data to segment video streams from wearable cameras to produce summaries of activity. Brain activity in the alpha and beta bands is used to estimate attention, so that regions where the user actively paid attention can be extracted from the video stream.

Employing a completely different form of hardware and one that limits the system to use in pre-prepared areas, de Silva et. al. [7] use pressure sensitive floor tiles to sense the location of people within a building. A number of video cameras are present within the test location. The floor tile data is used to produce a video which automatically hands-over between cameras as the user goes in and out of shot. The hand-over algorithm minimises camera switches while maintaining good coverage. In a commercial rather than research area, GeoVector ([www.geovector.com](http://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.

## **4. Benefits of Classifying Video**

When performing an evaluation of a mobile system, video capture 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 an 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. Additionally because of the simple fact that the camera is being carried around, it is often the case that the camera-operator is unable to keep the picture steady – especially in fast paced experiments. Large amounts of footage may therefore be unusable even if the camera was pointing towards interesting activity.

The technique presented here offers four specific benefits.

### **4.1 Automatically find video of a logged event**

Replayer allows an analyst 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 is likely to have

been captured on video. On selecting an event of interest, Replayer is able to return only sections of video in which that event may be 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 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.

#### **4.3 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 withdraw their consent for the use of their video footage. In a similar manner to the example in Section 4.1, Replayer allows us to filter videos for periods containing such participants, returning only video *excluding* them, and thus allows the presentation of only 'safe' data. Another application of this technique might be to exclude trial coordinators from videos. Multiple camera operators might appear in each others' recorded footage. Such periods could be filtered out from a presentation should this be desirable.

#### **4.4 Filtering out video of unacceptable quality**

In some cases, roaming camera-operators are unable to maintain a steady image when capturing in the field. They may need to run to keep up with participants, or have to pay attention to external factors such as traffic or perhaps another evaluator. In these cases where the operators' attention is withdrawn from the camera, this often results in unusable footage. Replayer now allows an analyst to automatically check for such footage and automatically skip past it.

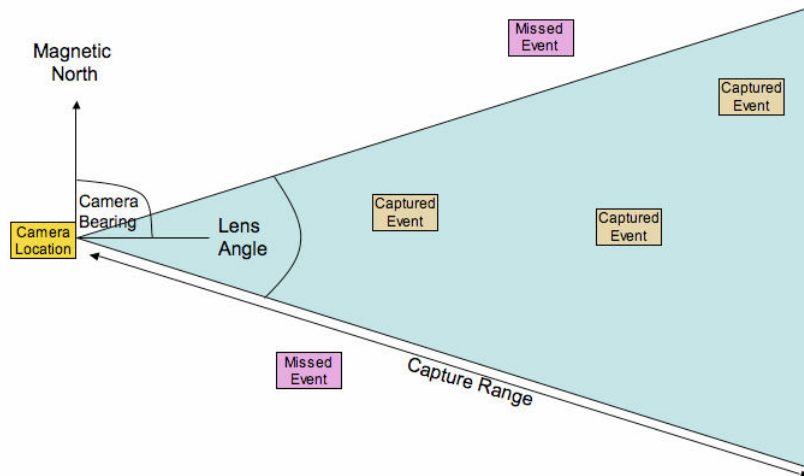
The following section describes the processes by which these techniques operate.

### **5. Auto-classifying Video Content**

A novel component has been added to the Replayer toolkit to automatically classify video to detect the users present in shot in any particular frame, the user activity being performed and the stability of the camera while the shot was being taken. This is achieved by logging during the experiment not only participant location, but also the camera's location and bearing. To analyse the experiment data, Replayer parses the

collection of log files and automatically detects the periods of participant activity that have been recorded. The classification can then be viewed using Replayer's existing visualisation components, as illustrated in Section 6.2.

The log files required for each camera contain information required to calculate the field-of-view at any particular instance. Timestamped location and bearing information are required, as are lens width and range. With static cameras that will stay in one fixed position during the course of an experiment, it is a simple matter to record this information at any time before or after the experiment. 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 to calculate the estimated field of vision. Figure 2 illustrates how the location of the event is then checked to see if it is within one or more of these fields.



**Fig. 2:** Spatial distribution of events in a mobile application experiment. A triangle is created for a static camera showing its field of capture. Each event is tested to see if it is within view.

Figure 3 shows an example of two participants' logged GPS trails that have been recorded during a trial and classified using this technique. In the image on the right, having processed camera location and bearing information, the events that fall into each camera's sights are highlighted in red. The fixed camera locations are also rendered on the map, with their fields of view shown as semi-transparent triangles. It is worthy of note that although the examples presented here use GPS locations, any positioning system can be used, so the technique could be used on data recorded indoors.

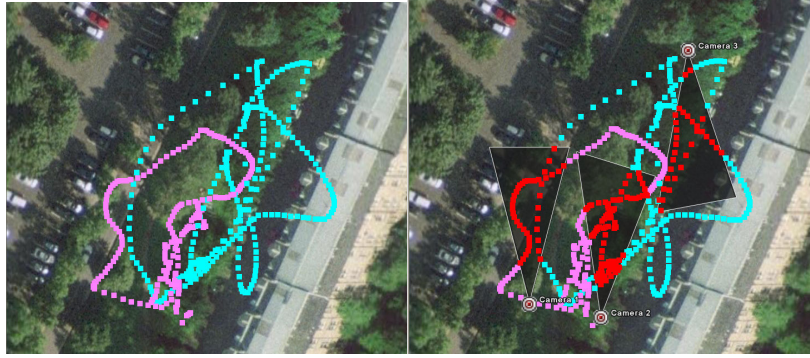
In the trial described in the following section, the roaming camera was augmented with a PDA interfaced to the MESH inertial sensing platform [9], which provides

GPS tracking along with tri-axis accelerometer, gyroscope and magnetometer sensing capabilities. Replayer is not restricted to this particular hardware, and will operate on any logged data representing camera operator positions and headings.

In addition to classifying recorded activity by camera bearings, the quality of footage can be gauged by the stability of the camera over short periods of time. Accelerometers in the MESH sensor pack record motion in the X, Y and Z axes sampled at a rate of 100Hz. By averaging the derivatives of motion in each dimension over short windows of time, the level of camera instability for that window can be judged:

$$\sum_{a \in \{1,2,3\}} \frac{1}{k} \sum_{t=T}^{T+k} \left( \frac{x_a(t) - x_a(t-1)}{\Delta t} \right)^2 \quad (5.1)$$

over a window length  $k$  for each axis  $a$ .



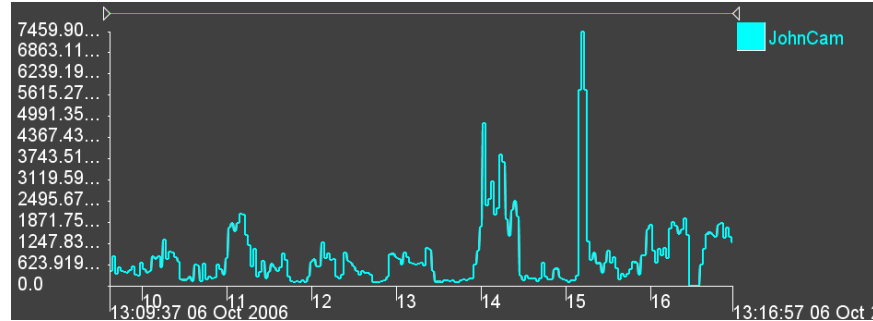
**Fig. 3.** The image on the left shows the GPS trails for two participants. The visual fields of the fixed position cameras are added to the image on the right, and the objects captured by the cameras are calculated and highlighted in red<sup>1</sup>.

Visualisation components in the Replayer interface can give overviews of the stability over time. Figure 4 illustrates, where the instability of the augmented camera is graphed over time. By looking at this graph there are clear periods where the camera was fairly still, others where the operator was likely to be walking and periods of high motion where the operator appears to be running. Such periods of high instability (when the footage is likely to be blurry) can then be filtered out at the analyst's discretion. Conversely, periods where the camera operator was running might indicate an occurrence of interest, which the operator was keen to capture. By selecting the high values (as was illustrated in Figure 1) or those immediately preceding or following, the video playback can instantly jump to this period.

The following section details an experiment where these techniques were put into practice in the evaluation of a simple mobile application.

<sup>1</sup> Note that the participants did not enter the building, and the slightly misleading impression that they did is due to the angle from which the satellite image was captured.





**Fig. 4.** Level of motion detected in augmented camera over time.

## 6. Example of Use

In initial trials to demonstrate the effectiveness of these technique, we developed ColourLogger, a simple representative application for mobile devices. The application's interface shows three buttons, marked red, green and blue. Participants were asked to walk around a small area looking around for objects of these colours and, on discovery of such an item, to press the appropriate button. Although ColourLogger is a simple application it is perfectly adequate to fully illustrate the benefits of the presented techniques.

### 6.1 Data Capture from the ColourLogger Experiment

The trial was conducted in an area of approximately 1000m<sup>2</sup>. The trial zone constitutes mainly of grass, with roads on either side. Two participants walked around this area with the ColourLogger application for ten minutes and were asked to record a button press for anything on the ground they encountered. ColourLogger was run on a Hewlett Packard iPaq hx2410 running Microsoft's PocketPC 2003 framework and using a SysOnChip compact flash GPS receiver. The time of each of the participants' button presses was recorded in a system log. A second continuous log was maintained recording timestamped GPS positions for the participant, along with the number of GPS satellites currently available. The latter value helps to determine the quality of the GPS fix.

Five video cameras were used to record activity in the experiment. Four cameras were set up in fixed locations and the fifth was carried by a camera operator who 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 that many low cost cameras can be used to add to the variety of video streams for a given experiment.

The roaming camera used was a more traditional CCD-based digital video camera, augmented using a Hewlett Packard iPaq 5550 interfaced to the MESH inertial sensing platform [9], which logs accelerometer, gyroscope and magnetometer data 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 correctly 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.

To ensure the veracity of the heading magnetometer readings, the camera mount shown in Figure 5 was constructed and used in the filming process. This rigidly fixes the roaming camera and MESH sensor pack together – maintaining a fixed relationship between their orientations – while providing a convenient grip for the operator. Critically, the mount also magnetically isolates the magnetometer from the disturbances induced by the battery and other metallic components within the camera, which would otherwise have significantly distorted the heading data. The accuracy of the heading data was confirmed by cross-checking the magnetometer readings against a standard magnetic compass during a calibration phase at the start of the field trials. An additional advantage to this mount is the fact that the weight of the PDA, which controls and records data from the MESH sensors counterbalances the camera, increasing the stability of the footage. The handheld camcorders typically used in such trials for their light weight and relative inexpensiveness are notorious for producing shaky footage simply because of the manner in which they are held. This mount system serves to reduce this, with the trade-off of adding more weight.



**Fig. 5.** The camera mount used in the trial. A PDA logging GPS locations, bearings and motion is attached to the base of the mount at a sufficient distance from the camera to be isolated from its magnetic effects.

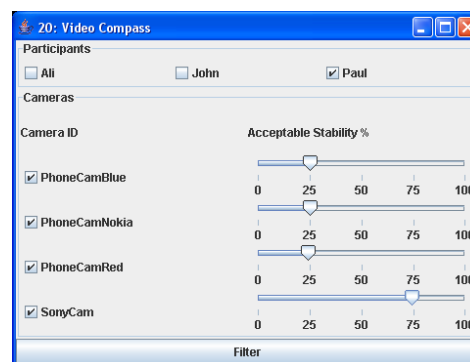
The field-of-view of the roaming camera is given by a cone extending from the measured location of the camera along the yaw angle estimated from the combined

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, the augmented camera was positioned next to them and an annotation was made in the logs, allowing for post-hoc synchronisation. The range of each camera was estimated to be 20 metres. Although participants are visible in the footage beyond this distance, the intention was to classify the periods where they were sufficiently close to observe in reasonable detail. This value may be altered as desired.

Once complete, the data captured from the experiment consisted of the following: five video recordings at a variety of qualities and stored in a variety of codecs; 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 logs and video files were subsequently read into the Replayer toolkit and synchronised using the QCCI [14] technique: an efficient method based on recording video footage of PDA screens displaying system times and using this information to calculate offsets for video files. To verify synchronisation, a whistle was blown at the beginning and end of the trial and recorded by all the cameras. Using Replayer's 'Play All' feature to play all the videos simultaneously verifies that the whistle is heard in each stream at the same time.

## 6.2 Analysing the Data

Once these logs have been parsed by Replayer, several options for examining the data are available. A number of examples of use are demonstrated below. The interface shown in Figure 6 can be used to filter the data. The checkboxes at the top allow an analyst to show or exclude data captured at times where the selected participants were in view. Similarly, checkboxes on the left allow this filtering process to involve any or all of the cameras. The sliders on the right set the minimum allowed stability for each camera, so that periods where there is a lot of camera movement can also be filtered out.

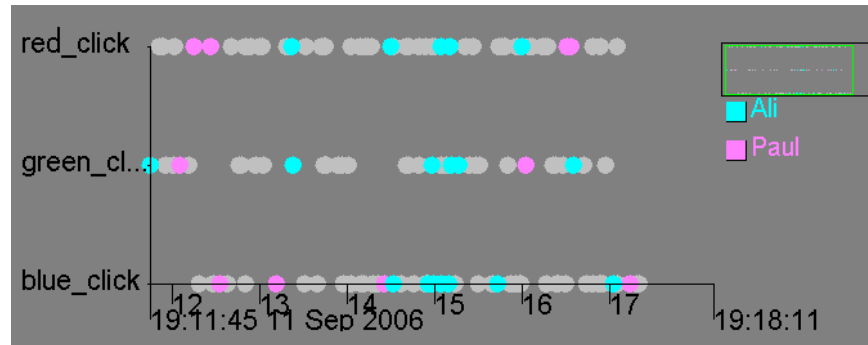


**Fig. 6.** The interface for the filtering controls.

### Visualising which events have been captured

The graph in Figure 7 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. The analyst can zoom or pan this graph, with the green bow in the top right corner giving a context of how the current view corresponds to the full graph.

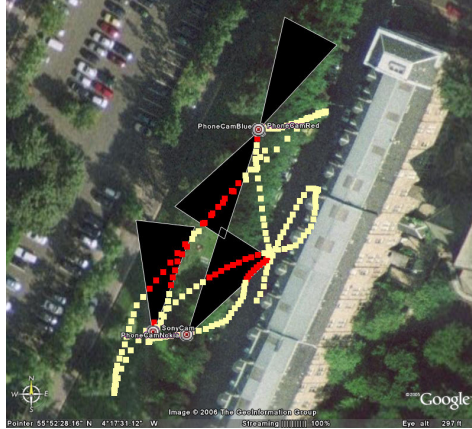
An analyst may be particularly interested in one type of event. Existing Replayer functionality allows one row of this graph (one type of event) to be selected, which would instruct the video component to show only the corresponding time periods. Of course, there is no guarantee that the video recorded 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. Events captured but above the threshold for acceptable instability are similarly filtered. This shows the analyst exactly which logged data can be enhanced by the context provided by video footage.



**Fig. 7.** An *Event Series* component consists timeline showing logged event occurrence over time (x-axis), and the named events distributed along the y-axis. Events are coloured by participant ID, and because of the selection done in another component, the events that were not captured on any cameras are greyed out.

### Spatial distribution of captured events

Replayer shows spatial data information by plotting points in Google Earth (<http://earth.google.com/>). The screenshot in Figure 8 shows logged GPS trails for one participant as he moved around the area in which the experiment took place. The locations and fields-of-view of the fixed camera locations are also shown. 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. Such an image would also be useful in re-positioning cameras for future experiments to capture more data.



**Fig. 8.** GPS trail for one participant during the trial. The four static cameras are shown, with the portions of activity captured on camera coloured red.

While this particular form of visualisation is very effective when dealing with fixed cameras, it becomes more complex when dealing with roaming cameras. Such spatial distributions are temporally collapsed, showing all the positions over a fixed period of time. When handling a roaming camera, there is not a single place to draw the visual field of the camera. Alternatively, an animation can be displayed: the Replayer mapping component can be set to ‘replay’ the recorded data, displaying only one icon for each participant and camera at a time, and showing how they moved in real-time during the trial. The roaming camera can then be shown to move around the trial area, with the triangle rotating to show the bearing at that time. This can be viewed in synchronisation with the video streams, to show in real time both the geographical location of the roaming camera and the video recorded at that time.

### **Playback video of a single participant’s activity**

A third visualisation, shown in Figure 9, displays five 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. As can be seen, the green areas around the marker thumb show periods of participant presence in the timelines labelled PhoneCamBlue, SonyCam and a short period in RoamingCam. This matches with the footage from the cameras shown above.

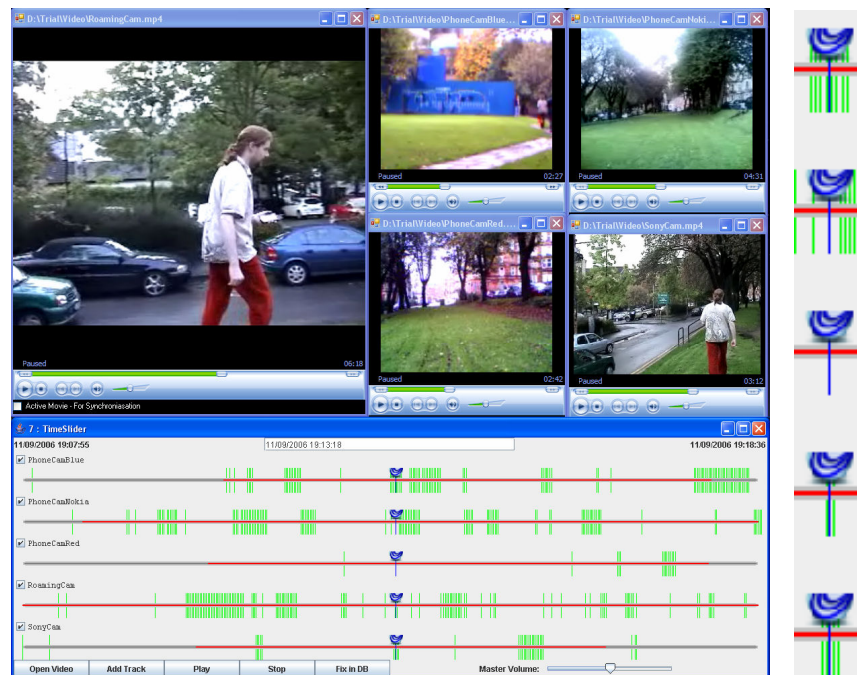
In the figure, all the streams are being played together. The lower part of the image shows a timeline for the experiment, which is shown to run from 19:08 until 19:18. A separate timeline is shown for each video stream, with the areas of each drawn in red showing the periods of the trial for which that camera has footage. As all the cameras were switched on and off at different times, these do not match, but Replayer synchronises the streams for concurrent playback. The thumb markers drawn on each timeline show the current position of the footage in each stream and the green areas of highlighting show the periods at which the participant was in view.

On playback, Replayer can be set to play all footage in real time or to show only the periods where the participant was recorded. In the latter mode, the system will display only those video streams that contain the participant, hiding those streams where no data is being displayed. Periods where the participant is outwith the view of all the cameras will be skipped entirely and multiple streams will be shown concurrently in the when the participant was captured by more than one camera.

The playback is also synchronised with other components so that, for example, glyphs on event graphs are highlighted as they occur.

### Filter video by camera stability

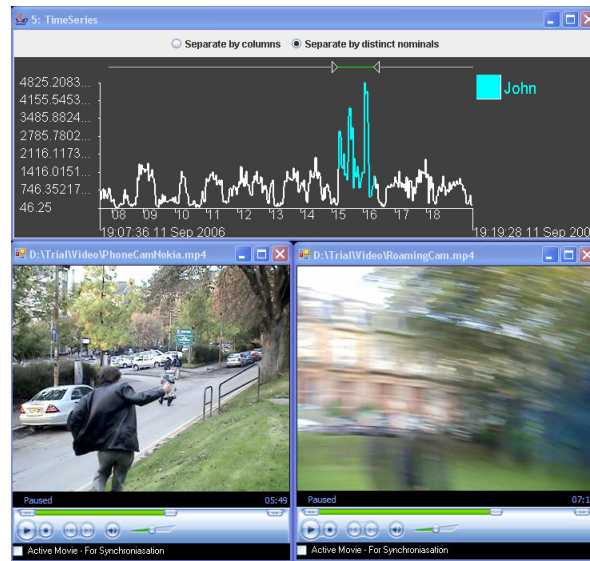
The final visualisation, shown in Figure 10, demonstrates the ability to analyse video data by camera stability. Replayer has processed the accelerometers logs from the augmented camera and created a graph of camera motion over time, as shown in the top of the figure. The video component at the bottom left shows footage from a static camera, whereas the one on the right is the augmented roaming camera whose stability has been graphed.



**Fig. 9.** Five video streams playing in synchronisation, with a timeline for each shown underneath. The analyst has selected to view all the video for a specific participant and the green highlighting on the timelines shows the periods in each of the videos where the participant appears. As can be seen, the marker is within the green area for three of the timelines, indicating that the participant should be in view of three of the cameras at this point. Video playback can be set to skip periods where the participant does not appear in any video streams. A larger image of the thumb positions is shown on the right for clarity.

The analyst has decided to select a period of high instability from this graph. This is achieved by dragging the triangular markers at the top of the tool, which highlights the selected period in blue. As a result of this selection, the two video components below the graph jump to the beginning of the selected period. The static camera has actually filmed the roaming camera operator at this time, and it can be seen that he is filming something to his right, but appears unsteady, perhaps running or losing his balance on the slope. A participant is visible in the footage recorded from the roaming camera at this point, but, as suggested by the graph, this is occurring at a period of high motion and the video stream is unstable and blurred.

The discovery of this footage is perhaps not of great benefit to an analyst, but this example serves to show how periods of camera instability can be successfully detected automatically, so that they may be filtered out of trial playback to leave only reliable footage, should that be the analyst's desire.



**Fig. 10.** The graph at the top of the figure shows instability over time of a roaming camera in the experiment. A period of high instability has been selected by an analyst, highlighting it in blue and selecting this period to view in the video components. The video at the bottom right shows the footage recorded from the roaming camera, while the video in the bottom left shows footage from a static camera that happens to have recorded the roaming cameraman at this instant. While it would be more likely that an analyst would want to filter *out* such footage, this example serves to demonstrate that periods of camera instability can be reliably identified.

### Accuracy of Results

To assess the accuracy of the system on this trial data a brief evaluation was performed. Video data was filtered to show only those periods where the system had



judged participants to be in view. The video was then manually reviewed to verify the system's findings. Several such comparisons were made for different cameras and filtering on different participants, with the system found to have correctly identified the periods at which participants were in view with results ranging between 61% and 82% accuracy.

These results are very dependent on the technology being employed. As GPS positioning is generally not guaranteed to provide pin-point locations, the results are not expected to be perfectly accurate in all cases. Subjective impressions of position accuracy were made by examining GPS trails and logs showing the number of satellites a participant's PDA could see at any given time. Smooth GPS trails probably indicate a good representation of a participant's actual route, whereas a more scattered display of points in the trail suggests more noisy data. From these impressions it appeared that one participant's position had been logged more accurately than the others. Of all the evaluation comparisons, the classification of this user's appearance on static cameras was found to be performed with the greatest accuracy. The example that gave percentage accuracy in only the low 60s was analysing the classification with the roaming camera of the participant who had the poorer GPS logging. In this case, inaccurate position logging of either the participant or the camera operator could lead to errors.

This analysis was not performed in a carefully managed lab environment, but has demonstrated that the presented techniques can operate with a fair degree of accuracy under realistic usage conditions. As one would expect, the hardware used in the trial did not all perform to the same standard, and the more precise sensors led to better classification of video. It is worth noting that the presented techniques are not dependent on GPS technology; as positioning technologies continue to improve, so will the accuracy of this technique.

### **Summary**

In performing qualitative evaluation of a pervasive computing system, an analyst may wish to examine many hours of video recorded on several cameras for each of several trials. Such a volume of data is very time consuming to work through looking for specific events or well-filmed actions. Although an analyst may wish to examine all the available footage in a thorough evaluation, there is clear benefit in being able to quickly find the most interesting periods without resorting to a linear search.

With the techniques presented here, an analyst can select a particular logged event of interest (for example, a specific type of user interaction, or a recorded system event such as entering into wireless network range) and be shown every occurrence of these on a timeline. This collection of events can be filtered to leave only those that were captured on video and those recorded when the camera was sufficiently steady to provide a clear image. Should the analyst wish, the recorded events can be further analysed on a map to see how far they occurred from the camera, and only those that occurred close to a camera can be selected. Having made this selection, the analyst can watch the events of interest across several concurrent video streams.

The simple ColourLogger application demonstrated here was intended to be representative of many mobile applications, and to allow for our experimentation and development to take place. Of course, the value of examining the data collected about the use of ColourLogger is minimal, but it has served as a stepping stone towards use



of the new Replayer extension in the trials of new mobile applications that are now in development. It is anticipated that these techniques will play an integral part in the running of system trials involving multiple analysts and multiple participants using our systems over a large geographic area and over a significant period of time. Other future work is described in the next section.

## **7. Future work**

As we further refine this technique one particular use we envisage is in the orchestration of a system trial. The sort of data described here could be uploaded and analysed in real time, something not implausible given the prevalent availability of wireless communications in modern devices; indeed all the devices used in our example were equipped with 802.11 WiFi and could have been sending data directly to a server. It would then be possible, for example, to show a histogram of the amount of activity captured for each participant. An orchestrator seeing inequality in this histogram could direct roaming cameras to concentrate on particular participants as required. Of course this brings up issues of assured continuous connectivity, something that [5] points out may not be as simple to achieve as it seems. 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 into Google Earth is a 3D representation of all the buildings in a given city; we hope to use this information to limit the modelled fields of view from each camera to reflect this occlusion. Additionally, sensor information is available in three axes, 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.

## **8. Conclusion**

We have presented a technique for analysis of data recorded in mobile application evaluations. Specific recorded user activity can be queried, to automatically skip irrelevant video footage among the volumes of data recorded by multiple cameras, and to focus on that which is salient. This reduction can include, for example, showing video in which a particular participant appears, or showing only that area of video in which a particular system event has been captured. Additionally we demonstrated a technique by which video of an unacceptable image stability can be automatically discarded. When performing an evaluation of a larger scale system, potentially hundreds of hours of video may be captured. The technique described in this paper allows an analyst to quickly locate specific data of interest within that footage. We devised, implemented and recorded a representative application to demonstrate how this method might be used, and showed some of the capabilities provided by this extension to the Replayer toolkit. This extension of Replayer does require some additional hardware; however, the hardware used is becoming increasingly available in commodity devices. We believe this is a valuable addition to

the already versatile Replayer toolkit, and suggest that it could be widely applicable in the evaluation of mobile computing systems. It serves as another example of the benefits arising from combining quantitative and qualitative data, and from combining associated analytic approaches, in synergistic ways.

## Acknowledgements

Removed for anonymous review.

## References

1. Aizawa, K., Ishijima, K., Shiina, M.: Summarizing Wearable Video, In Proceedings of International Conference on Image Processing (ICIP2001), Thessaloniki, Greece (2001) pp. 398-401.
2. Badre, A. N., Guzdial, M., Hudson, S. E., and Santos, P. J.: A User Interface Evaluation Environment Using Synchronized Video, Visualizations and Event Trace Data. Software Quality Journal, Volume 4, Number 2 / June, 1995, Springer.
3. Becker, R.A., Cleveland, W.S.: Brushing Scatterplots. Technometrics, Vol. 29 (1987) pp.127-142.
4. Beeharee, A., Steed, A.: Filtering Location-Based Information Using Visibility. In Proceedings of Location- and Context-Awareness (LoCA 2005), Munich, Germany (2005) pp. 306-315.
5. Chalmers, M., Galani, A.: Seamful Interweaving: Heterogeneity in the Theory and Design of Interactive Systems. In Proceedings of ACM Symposium on Designing Interactive Systems (DIS 2004), Massachusetts, USA (2004), pp. 243-252.
6. *Removed for anonymous review.*
7. de Silva, G. C., Ishikawa, T., Yamasaki, T. and Aizawa, K.: Person Tracking and Multicamera Video Retrieval Using Floor Sensors in a Ubiquitous Environment. In Proceedings of International Conference on Image and video retrieval (CIVR2005), Singapore (2005) pp 297-306.
8. French, A., Greenhalgh, C., Crabtree, A., Wright, M., Hampshire, A. and Rodden, T.: Software replay tools for time-based social science data. Presented at the 2nd International Conference on e-Social Science, Manchester (2006).
9. Hughes, S., Oakley, I., O'Modhrain, S.: MESH: Supporting Mobile Multi-modal Interfaces. Presented at the Seventeenth Annual ACM Symposium on User Interface Software and Technology (UIST 2004), Santa Fe, New Mexico, USA (2004).
10. Lyons, K., Starner, T.: Mobile Capture for Wearable Computer Usability Testing In Proceedings of The International Symposium on Wearable Computing (ISWC 2001), Zurich, Switzerland (2001) pp. 69-76.
11. McCurdy, N.J., Carlisle, J.N., Griswold, W.G., Harnessing Mobile Ubiquitous Video. In: Proceedings of the ACM Conference on Human Factors in Computing (CHI2005), Portland, Oregon, USA (2005) pp. 1645-1648.
12. *Removed for anonymous review.*
13. Sawahata, Y., Aizawa, K., Bakker, E. M. et al (Eds.): Indexing of Personal Video Captured by a Wearable Imaging System. In Proceedings of International Conference on Image and Video Retrieval (CIVR 2003). Urbana, IL, USA (2003) pp. 342-351.
14. *Removed for anonymous review.*