ABSTRACT
Web-accessible video sites – such as YouTube - currently comprise several of the most trafficked sites in the world. Mobile smartphone penetration is also at an all-time high, as is user-appetite for innovative mobile services. This paper anticipates the desire for video content understanding on mobile smartphones by on-the-go users. The implemented tool provides an innovative, compact, visual way for users to use their smartphone to understand the content of a video of interest before download, and the approach goes beyond the prevalent VCR-like controls and static keyframes of today. With a large e-commerce ecosystem evolving around mobile video, this work is extremely topical. We present our early design and implementation details and show how to support deeper mobile video-understanding than the current limited state of the art.

Categories and Subject Descriptors
H.5.2 [User Interfaces]: Screen Design, H.4.3 [Comm. Applications]: Information browsers, H.3.3 [Information Search and Retrieval]: Search process

General Terms
Human Factors

Keywords
Designing services or mobile devices, Mobile and online video, Temporal media browsing and understanding.

1. INTRODUCTION & MOTIVATION
In a very short time period, YouTube has become one of the biggest video databases in the world [1]. Featuring millions of videos, each one about 9 Mbytes big and several minutes long, thousands of new videos are uploaded each day. While YouTube user-generated videos are often short – minutes, not hours - iTunes, MSN, and Google Video offer short, episodic, and full length content. The Web ecosystem is now seeing revenue opportunities in video downloads and rentals (e.g., iTunes). To complement this model we would expect that sophisticated video summarization/understanding tools would emerge to help customers quickly understand the content of the video they are interested in. However, most Web sites offer very poor and limited tools for video understanding and mobile sites - with tools for mobile smartphones - even less.

In this paper, “video understanding” means the act of browsing through video content in order to create a mental model of the content to some sufficient degree. The user’s sufficiency requirements may include determining conceptual details such as: “Is a goal is scored in the first 10 minutes of this football match clip?”, “Does it have a scene in which two men fight onboard a helicopter?”, “Does it have a scene in which a cat fall off a ledge after a baby scares it?”. Note that the above types of questions are generally not easily answered by indexing, nor would they likely be called out in any editor’s textual description of the video. Thus the benefits of content-based video browsing are clear in these cases. On the other hand, the paradigm is not as useful when the videos are very short, of low quality, or if sound is the medium of interest in the user’s mental model.

1.1 Related Work
There are few effective tools for video content browsing on mobile devices. Even on the Web, none would seem to be as powerful or compact as the proposed system. YouTube [2] does not provide informative “preview” information for videos apart from a single author-chosen keyframe. YouTube’s Warp tool does show the relationships between videos in a graphical way, but not fine-grain details of the content within a given video. YouTube’s Java application for smartphones cannot preview content apart from a single keyframe. MotionBox [3] and other similar sites use the prevalent technique of showing a static keyframe strip below the movie. Guba [4] employs a 4x4 matrix of keyframes for any given video – but the representation is non-navigable. Upon request, Internet Archive [5] lays out one keyframe for each minute of the video in question, to allow a somewhat weak view of the video content. Finally, note that the current art also finds videos marked with “tags” but that the tag paradigm’s weak semantics make it infeasible for deep video understanding (i.e., as in the above sufficiency requirements). In summary, our work is innovative because it is a compact but highly effective way for users to quickly, systematically, and nonlinearly browse video content in order to make a “watch/no-watch”, “rent/rent-rent”, or “download/no-download” decision.

2. MOBILE TILES - CONCEPT
The MobileTiles tool runs on a smartphone, presents video content to the user, and allows interactive browsing. Several reasonable assumptions are made: 1) a backend video repository provides media (e.g. keyframes, metadata, etc.) to the tool, 2) the
smartphone has at least a 340x240 color screen, 3) the videos in question are sufficiently long that simply fast-forwarding through them is not an acceptable way to build a mental model.

Several features of the browsing tool are user-configurable, including the number of tile-rings (an important visualization tradeoff). The ring options are: a) No rings around the focus which takes up the entire screen (this is the default “playback” configuration), b) One ring around the focus (e.g. Figure 4), c) Two rings around the focus (e.g. Figure 5, allows fine grain view of content and meta-content (e.g. scenes)). Computationally, the tool’s main concerns are: assignment of media fragments to screen regions, rendering of the interface accounting for screen size, and managing fragment playback. A representative high-level view of the entire ecosystem is shown in Figure 2. The MobileTiles application on the smartphone uses a data connection (cellular, Wi-Fi) to access a video storefront for video selection; an associated media server serves of metadata and keyframes to the smartphone on demand.

3. IMPLEMENTATION & RESULTS
We selected Java Platform, Micro Edition (Java ME) as a basis platform for development of the mobile client for content browsing. The Mobile Information Device Profile (MIDP) 2.0 on top of the Connected Limited Device Configuration (CLDC) 1.1 constitutes the most widely adopted Java ME application platform used in mobile phones today. Flexible user interfaces and built-in network capabilities of this platform provides all required functionality for MobileTiles application. At the same time, world-wide prevalence of MIDP 2.0 platform allows us to support this application on a great variety of mobile smartphones. We used Sun Java Wireless Toolkit 2.5 for CLDC as an emulator of this Java ME platform.

Consider the structure of MobileTiles application in Figure 7. BrowserMidlet is the main entry-point class of MobileTiles. It defines key elements of the application – initialization, set of supported commands, events handling and life cycle methods. The BrowserCanvas class is the heart of the MobileTiles – it constitutes the graphical user interface of application. The most important responsibility of BrowserCanvas is to create the layout for related images and draw them. Each frame image in the layout is characterized by two elements: position of the image in the canvas and frame range it belongs to. This corresponds exactly to the class structure: the object of the FrameData class encapsulates instances of FrameImage and FrameRange classes. Management of frame images layout is delegated to FrameTree class. This class populates a tree of BaseTreeNode objects which contains corresponding FrameData instance as a node data. The ImageProvider interface enables a pluggable approach to get required image. Classes implementing this interface provide different means to get the image for given size and id. The JarImageProvider loads the corresponding images bundled in the jar file with application (in a real deployment the class would obviously get frames from a networked Media Server instead of a local resource). The FramePlayer class enables the playback of the movie corresponding to the provided frames.

The main challenge in the development of MobileTiles is to create an effective and flexible system for the frame images layout. This system should resolve the following difficulties:

- support different number of ring configurations
- effectively fill in available canvas space
- build and display the correspondence between neighboring frames belonging to different rings

Figure 1. Keypad mapping (partial)
Visually, MobileTiles presents a central keyframe in the screen’s center, surrounded by 1 or 2 outer “rings”. The keyframes in the rings are temporally ordered from the top-left and sampled equally from the current visible frame range. The center keyframe is called the “focus” and the view range is the portion of the video surrounding the focus frame that is currently under scrutiny. Thus in Figure 5 the entire video range is currently represented surrounding the focus and can be understood by scanning the keyframes clockwise from the top-left. The entire tiled view is clickable via the keypad (or touch screen). When using a keypad the keys are mapped as shown in Figure 1. Clicks upon individual keyframe tiles result in a refocusing of the representation on the new region of the video using the selected tile as the new focus. Special clicks are possible, such as keypad “5” (clicking on the focus frame) which corresponds to dividing the current range of view in half (i.e., zoom). Thus, browsing both temporally (forwards/backwards) and zooming up and down into detail (showing more or less frames per time period) are easy and quick. The tool allows for the instantaneous playback of any region of the video including: a) just the range currently being examined, b) the whole video, c) play all tiles at once.

Figure 2. Ecosystem and related components
In this implementation we assume that frame content is available in a square shape as this makes the layout simple and optimal. We bind the frame image layout with corresponding tree. This tree is populated using the following rules:

- key frame image corresponds to root node of the tree
- children of particular node correspond to smaller frame images neighboring to frame image of that node:
  - corner frame node has 5 children (4 side nodes and 1 corner node) (green color, solid outline)
  - side frame node has 2 children (2 side nodes) (yellow and pink colors; long and short dashed outlines, resp.)
- frame range of parent node is distributed uniformly between its children

The colored layout in Figure 3 shows the correspondence between frame images and tree nodes: root nodes are color, corner nodes are green and side nodes are yellow and pink.

A special keystroke changes the number of rings visible at any time: from 0 to 2 inclusively (0 rings means the user sees only the given frame which fills the screen). In figure 4 the user uses only 1 exterior ring, which is sufficient to see the level of detail she desires at the moment.

Figure 5 shows the tool in a state where the view range is essentially the entire video and the focus is essentially on the median frame of the video. This is the “reset” view and presents the video content at a glance to begin the browsing session. Note here that the video content’s scenes and cuts are revealed automatically here from this view. By scanning the rings temporally (clockwise from top left) we see the opening credit the scenes and the finally the ending scenes.

### 3.1 Results

The results we have seen are promising, on both emulators and MIDP smartphones; the tool operates extremely intuitively.

Figure 4 shows the tool as it appears on a smartphone and explains the configurable options and feature set.

Figure 6 shows the tool zooming into detail from the previous view: thanks to some clicking the user now sees a range of frames that is only 40 wide, centered around frame 230. Note here that the user can now understand in more detail the “shots” and “scenes” in the video content as we see the transitions from scene to scene as we scan the first and second “rings” temporally.

### 4. CONCLUSIONS

We have designed and implemented a novel and effective nonlinear video content understanding tool for mobile smartphones. Currently, no smartphone offers such an interface and the Websites of video repositories offer only limited, static, keyframe or timeline (linear) based modes, none of which allow a comparable degree of content understanding to MobileTiles.

The tool follows user centered design and user experience best practices (to the extent that they exist for mobile applications) by:
1) using a culturally and universal metaphor (i.e., click-me tiles), applied consistently throughout the tool, 2) focusing the user’s attention on the key information (i.e., the center focus frame) while providing peripheral information in the rings, 3) helping the user maintain local details while also seeing a global context.

The MobileTiles proof-of-concept is designed to run on Java J2ME MIDP enabled smartphones; J2ME is one of the main smartphone development environments and has huge market penetration. Nothing prevents us from porting this tool to other mobile OS’s like Symbian, Windows Mobile 6, OS X. Our implementation work had to overcome the inherent limitations of mobile devices (e.g., screen size, memory, and response time) and provides a generic architecture solution that currently runs upon the Java ME platform (but conceptually on a spectrum of mobile smartphones). The classes supporting the user interface of the tool were specifically designed with mobile devices in mind and can adapt to different screen sizes and user preferences. In the end, it allows video navigation in convenient, simple, and quick way.

Current and future work includes additional porting to various Java-enabled devices, creating a systematic approach to accommodating differing smartphone screen sizes, and accommodating different movie frame proportions from Media Server.

5. REFERENCES