An Affordable Wearable Video System For Emergency Response Training

Deen King-Smith, Aravind Mikkilineni, David Ebert, Timothy Collins, and Edward J. Delp

Video and Image Processing Lab (*VIPER*)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

ABSTRACT

Many emergency response units are currently faced with restrictive budgets that prohibit their use of advanced technology-based training solutions. Our work focuses on creating an affordable, mobile, state-of-the-art emergency response training solution through the integration of low-cost, commercially available products. The system we have developed consists of tracking, audio, and video capability, coupled with other sensors that can all be viewed through a unified visualization system. In this paper we focus on the video sub-system which helps provide real time tracking and video feeds from the training environment through a system of wearable and stationary cameras.

These two camera systems interface with a management system that handles storage and indexing of the video during and after training exercises. The wearable systems enable the command center to have live video and tracking information for each trainee in the exercise. The stationary camera systems provide a fixed point of reference for viewing action during the exercise and consist of a small Linux based portable computer and mountable camera. The video management system consists of a server and database which work in tandem with a visualization application to provide real-time and after action review capability to the training system.

Keywords: wearable video, mobile video, emergency response, mobile surveillance

1. INTRODUCTION

The concept of a wearable computer is not new. Research and development in the area of wearable computing has existed since the late 1970s [1]. With advances in technology, form factors of these devices have become increasingly smaller and compact. Early prototypes were awkward and cumbersome. Originally existing as backpack based systems tethered together with wireless communication, these systems have now been transformed into completely unobtrusive Internet connected multimedia devices small enough to fit in a shirt pocket, or even woven into the fabric of the shirt itself [1]. This small form factor combined with the power of a desktop computer has given rise to a host of novel applications including training of emergency response personnel.

Many emergency response units are currently faced with restrictive budgets that prohibit their use of advanced technology-based training solutions. These solutions typically include a core set of features such as personnel and asset tracking capabilities, information from environmental sensors in the training environment, live audio and video feeds, coupled with remote centralized command and control which allow for viewing and review of the exercise. Capturing this information is valuable because it provides unique reviewable perspectives during exercises that directly enhance the effectiveness of the training. Examples of these types of systems can be found in use by many military and government agencies such as the US Army, FBI, and Department of Homeland Security.

While very effective in their use, these solutions and tools are often built in a proprietary manner by a single contractor which significantly increases the associated cost of the system. What is needed is a modular system that can be used with a number of different products from a diverse set of vendors with minimal modification. The goal of our work is to create an affordable state-of-the-art mobile emergency response training solution

Address all correspondence to E. J. Delp, ace@ecn.purdue.edu

Multimedia on Mobile Devices 2009, edited by Reiner Creutzburg, David Akopian, Proc. of SPIE-IS&T Electronic Imaging,
SPIE Vol. 7256, 72560E · © 2009 SPIE-IS&T · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.805734

through the integration of low-cost commercially available products. Specifically we will accomplish this through the use of a wearable video tracking system. The system we have developed provides the core set of features that are found in current solutions using a modular platform in which other tools may be integrated. Our system consists of tracking, audio, and video capabilities, coupled with other sensors that can all be viewed through a unified visualization system. In this paper we describe the video sub-system. The design, development, and prototyping of the video sub-system will be discussed along with an analysis of design challenges encountered during development.

2. RELATED WORK

Many wearable video systems have been proposed. Ranging from detection of eye contact [2] during personal life blogging [3] to developing novel methods of managing the data generated [4] [5], there are many applications that have made use of wearable video. Research at CMU has studied the usefullness of video in collaborative work enviroments [6], while researchers in the UK have focused on leveragin werable video for the sharing of experiences [7]. This this section we present in detail two examples of a werable video based systems.

2.1. NetMAN

NetMAN was designed to enhance the collaboration between field-based and office-based techniques when responding to technical support calls [8]. This project enables field technicians to contact and "use" colleagues at the the home office. The wearable system consists of a personal computer-like device with a head-mounted display(HMD). The head-mounted display provides an integrated video camera, microphone and speaker. This configuration provides the user with real time audio conversation capability with experts at the home office and also provides for transmission of video over a wireless 802.11 network. Video is transmitted at a rate of up to 2 frames/sec with a resolution of 200 x 200 pixels. This is then integrated into a central database which is accessible by the technician in the field. The technician is able to document their work using a shared notebook-based software which makes information entered into the system available to co-workers though a centralized database [8]. Initial development and testing of this system at the University of Oregon showed that a head-mounted display, while providing significant benefits in terms of usability and access to information in the field, was not a viable solution for long term use due to the size and weight.

After many experiments the conclusion was reached that contrary to previous research [9], the video quality directly correlated with the usability of the video. Thus the video was deemed poor and not adequate for use. The poor video quality was primarily due to the low frame rate and lighting which resulted in motion blur. While the system extends the capabilities of field based personnel, there were limitations that adversely affected the overall performance.

2.2. Self-Cam

Self-Cam is a system developed by the Affective Computing group at the Massachusetts Institute of Technology (MIT). Affective Computing is described as computing that relates to, arises from, or deliberately influences emotion or other affective phenomena. Self-Cam is a tool for reflection upon one's own expression, and allows one to see themselves from an outside point of view while engaged in interaction with others [10]. The system consists of a chest mounted camera on a boom, which is attached to a small portable computer carried in a pouch by the user providing 380 lines of resolution. The camera, which is powered by a 12-volt rechargeable battery pack, provides video that is digitized by a USB video capture device at 30 frames per second before being sent to a portable computer for processing in real time [10]. Self-Cam is designed to work with Mindreader, a computer vision based system that uses head and face movements to attempt to recognize a person's affective and cognitive state [11]. The combination of Self-Cam and Mindreader was used as part of a study to evaluate improvement of social cue recognition by autistic persons [10].

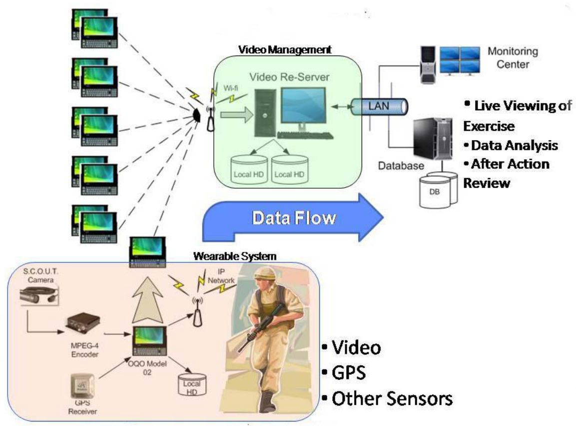

Figure 1. Video system diagram

3. SYSTEM DESIGN

Our over-all objective is to create a wearable video system that can transmit video, location, and other sensor information in real time to a remote data center over an IP network. Figure 1 shows a diagram of the video system. The system collects video and Global Positioning System (GPS) data for use in emergency response training. We primarily focus on the capture and transmission of video and location data from the user. Our goal was to keep the overall per unit cost of the video system as low as possible while maximizing its functionality. To do this we used off the shelf components that are readily available and that provide the following functionality.

Hardware:

- Small durable mountable camera (Sec. 3.1)
- Real-time USB hardware MPEG encoder (Sec. 3.2)
- GPS receiver (Sec. 3.4)
- Compact portable computer (Sec. 3.3)
- Central database (Sec. 3.5)
- Video management server (Sec. 3.5)

Software:

- Video management software on the portable device
- GPS management software on the portable device
- System management software on the server
- Database software

3.1. Camera

The VIOTAC S.C.O.U.T. camera (Figure 2) was chosen for its light weight and wide range of mounting options, providing the most flexibility for our scenario. The camera is also very durable given the proposed operating environment and provides an analog composite output consisting of approximately 500 vertical lines resolution at 20-30fps. The camera is powered by a rechargeable battery pack, which provides approximately 3 hours of usable time.

Figure 2. VioTAC S.C.O.U.T. camera

Figure 3. Sensoray 2250 video encoder

3.2. Video Encoder

There exists many USB video encoders on the market, and many require an external power source. For our purposes it was essential that the device be powered via the USB bus. Given this restriction, the range of choices is reduced. The first encoder investigated was a generic USB powered MPEG-4 encoder. The problem with this unit was that it is optimized to work with a specific set of programs and does not have an open source API. After additional evaluation, the Sensoray Model 2250 USB MPEG Frame Grabber was chosen as the encoder (Figure 3). This device provides an open source Linux and Windows APIs which enable direct control and manipulation of the device.

- Video Inputs: Composite, S-video
- Input Video Formats: NTSC, PAL
- Output Video Formats: MPEG1, MPEG2, MPEG4
- Output Resolution: 320x240, 720x480
- Data rates: Constant and variable up to 6Mbps

Figure 4. OQO Model 02 portable computer

3.3. Portable Computer

The requirements for this component are a small footprint, USB port, network connectivity via 802.11, and long battery life. Laptops are becoming increasingly smaller and PDAs are starting to incorporate features that can at times put them on par with laptops which increases the number of choice for devices that can meet our requirements. The first choice for this device was the Sony UX-280P. This device comes preloaded with Windows XP, and provides all the required capabilities. However, during initial testing it was realized that the UX-280P does not work well with the selected encoder. We ultimately chose the OQO Model 02 (Figure 4). The OQO provides a more compact design and meets the feature requirements.

The OQO is designed around a 1.5GHz VIA C7M Ultra-Low Voltage processor, has 1GB of RAM, and weighs about 1 lb. Storage on the device is provided by a 60GB hybrid hard drive which gives the device better battery life and improved performance. Network connectivity is provided by an 802.11 a/b/g internal network card, and the device comes pre-loaded with Windows XP. Ultimately we decided to use Debian GNU/Linux on the OQO. This choice was made because Linux offered much more control over the operation of the device and provides the ability to performance tune the OQO for the specific task of Video and GPS collection.

3.4. GPS Receiver

The Global Positioning System (GPS) is widely used for navigation related tasks. For our design we desired a compact GPS device with a USB interface and an intergrated antenna. The device chosen was the the Pharos i500 GPS Receiver (Figure 5).

This device has an internal patch antenna, provides 10-meter accuracy, weighs 18 grams, and has a maximum start time of 35sec. The data output from the device follows the NEMA-0183(V3.1) standard, using the WGS-84 Datum. The receiver is capable of tracking up to 12 satellites simultaneously, while only needing 3 for an accurate fix.

3.5. Servers

Our system incorporates two servers, a central database server and a video management server, known as the *re-server or video re-server*. The database server uses Oracle 10g and is designed to integrate the data from all the various collection points and make it usable by the visualization application (not discussed in this paper). One limitation of the Oracle platform was that, while it is able to handle meta-data associated with the captured video, it is not able to manage the video files themselves. This prompted the introduction of the re-server, running Debian GNU/Linux, which is powered by an Intel 2.13 GHz Dual Core processor with 2GB Ram and two high performance 80GB hard-drives in a RAID 0 configuration for storage of the video.

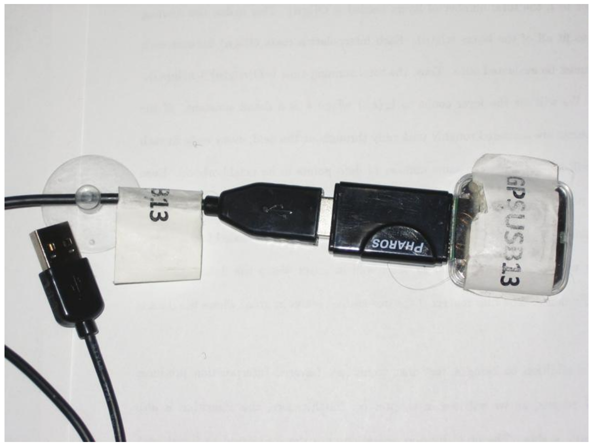

Figure 5. Pharos i500 USB GPS receiver

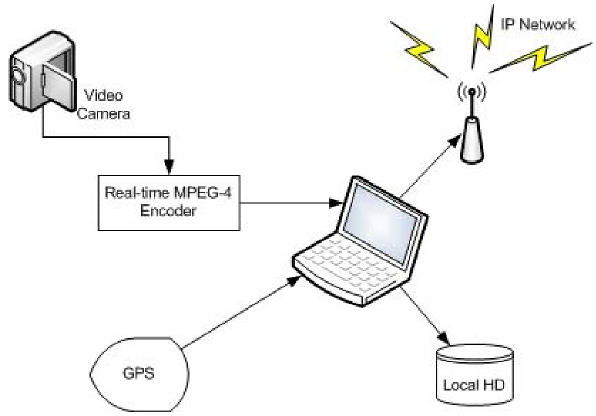

Figure 6. Wearable system components

4. SYSTEM INTEGRATION

The video system can be segmented into two functional parts, video management and video acquisition. Video acquisition is accomplished by the wearable system and the stationary system (Figure 1). The wearable system handles the capturing, encoding, storing, and broadcasting of the video feeds along with GPS tracking information from participants in the exercise, whereas the stationary system provides the same functionality from a fixed frame of reference in the exercise environment. The video management component is tasked with remote management of the wearable systems and serves as the video data store for video information.

4.1. Wearable System

Figure 6 shows a diagram of the wearable system. This system enables the command center to have live video and tracking information for each trainee in the exercise. Each wearable system consists of a small portable Linux based computer coupled with a GPS radio, a rugged mountable camera, and a wearable harness to hold the equipment. Each camera is mounted in a location that provides a first-person perspective of what the trainees see as they move through the exercise. Video and tracking data captured by the system is broadcast wirelessly back to the command center for viewing. The wearable system requires only one touch to be brought online and requires no other input for the duration of the exercise and has a battery life of approximately 3 hours. All devices are networked with one another. In the case of the wearable system, connectivity is realized via a deployed 802.11a wireless network. This network covers the entire test site and provides data rates of up to 54Mb/s [12].

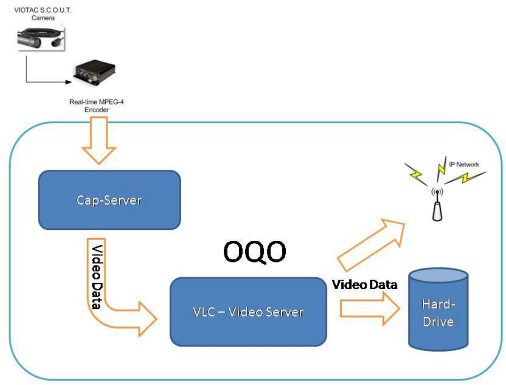

Figure 7. Video data flow within the OQO

The wearable system is separated into four main hardware components, the GPS receiver, camera, encoder, and the OQO. These components are managed by two control systems, one for the video and the other for the GPS. Initial development of the wearable system was performed in a Win32 environment. A number of issues emerged during development concerning control and reliability of the system under Windows XP. There existed instances where the video system would work and spontaneously cease to function. In addition the system monopolized a lot of resources on the device. Due to these issues, we decided to rebuild the system using Linux. This transition allowed us to customize the operating system in a way that was not possible on Windows XP.

In Linux both control systems were implemented using shell scripts which leveraged other applications on the device. In the case of the video subsystem, the control script leverages *VLC* [13], an open source video server package, and *cap-server*, an application bundled with the encoder for receiving video streams from the encoder. The video system was designed so that during an exercise the video is available to be streamed live while also being written to the local hard-drive of the OQOs. The video saved on the OQO is transferred to the re-server at the conclusion of the exercise. The re-server then acts as the central video store for the video files for viewing post exercise. Live streaming of the video is accomplished through a direct connection to the OQO. The only input required from the user is to press the power button on the re-server and the OQOs to bring the whole system online.

Video data moves from the camera to the encoder that encodes the video into MPEG-2 at a resolution of 352 x 240, 30fps, with an average data rate of 512kb/s. *Cap-server* receives the encoded video stream and writes it to a pipe that is read by VLC. VLC takes this data and saves it to the local hard-drive while simultaneously making it available for broadcast (Figure 7). The video is saved as MPEG-2 and packetized into an MPEG Transport Stream (MPEG-TS).

Most Wi-Fi networks using 801.11b/g operate in the 2.4GHz band [12]. This spectrum is also shared with other wireless devices such as cordless phones. To reduce interference from other RF devices, an 802.11a wireless network which operates in the 5GHz band is used for the video. Transmission of the video over Wi-Fi is handled by VLC using HTTP on port 8080. VLC is also used as a player to open the stream on the player/client side.

The GPS system was initially controlled via a standalone script. Poor performance was observed during testing and the GPS system was recoded in C. A pseudo terminal device is used to communicate with the GPS receiver which enables reading of the NEMA GPS streams as they are generated by the GPS receiver. This code extracts lines that begin with a *GPGGA* string. These strings contain location and fix information of the wearable system. Lines without the *GPGGA* string are discarded. The entire NEMA string is paired with device ID information and then sent to the database over the IP network.

When the OQO is powered up, *cap-server* is started followed by the VLC server. During this time, a unique file name for storing the video is generated for VLC. Once the video system is up and running, the file name, along with other meta data and connection info, is pushed into the database for use by the visualization application.

Device information is also pushed to the re-server for management purposes. At the end of the exercise a shutdown command is sent to all of the OQOs from the re-server. The OQO stops the video system, updates the video meta-data in the database, and copies the recorded video file from its local hard drive to the re-server. Once the copy operation is complete the OQO shuts itself off.

4.2. Stationary System

During development, we realized that deploying fixed stationary cameras in the environment would add an additional perspective that would be helpful during the post-exercise review. The addition will enable one to observe the actions of the participants beyond the perspective of the wearable camera system. The stationary camera systems consist of a small Linux based portable computer and mountable camera, but does not make use of a GPS system. The stationary cameras use the same configuration as the mobile camera with the exception that they broadcast their video over a wired 100Mb/s Ethernet network.

4.3. Database Connectivity

The database provides an efficient method for indexing the videos and associated information. This information is used by the visualization application to provide simultaneous synchronized video playback across multiple streams. Both the GPS and video sub-systems rely on a database connection to make sure that the visualization application can interact with the device. This connectivity is achieved by installing the Oracle 10g client on the OQOs. When information is pushed to the database the scripts use SQL to place the information into the correct tables.

4.4. Video Management Server

All of the remote devices can be manipulated directly from the management server. This server, also known as the re-server, allows remote management of the OQO's during and at the end of the exercise. During startup each OQO sends the re-server their respective connection information. This is the last step in the startup process and also signifies that the systems on the OQOs have finished starting up. When the exercise is complete, a stop command is issued from the re-server that instructs the OQO to stop broadcasting, upload it's video, and update the database before shutting down. This allows the complete wearable system to be controlled by one person remotely from the re-server. After the exercise, once the videos have been transferred, the re-server prepares the video for use in after-action playback by the visualization application.

5. TESTING AND DEPLOYMENT

To deploy the system in the field, the system is transported in a standard SWAT belt that accommodates the computer and various other devices and components. The OQO, encoder, and battery pack for the camera are fitted into the main pouch of the belt (Figure 8). The cables connecting the video camera and the GPS are the only wires leaving the pouch. Once an user is ready for the system, all that is needed is to press one button to power up the OQO, turn on the battery for the camera, and the wireless system will be online recording video within 5 minutes.

Initial testing was conducted at Purdue University to evaluate the functionality of video system. Once functionality was validated, the system was moved to the test site for integration with the larger system.

5.1. Approach

An incremental approach was used during testing. First the GPS, wireless, and wired video were tested followed by testing of the system as a whole. For the full system test we equipped participants with the SWAT belt (Figure 9), mounted the camera on the bill of a baseball cap, and a back pack containing other sensors developed by another group. After evaluating a number of different mounting options, the lid of the baseball cap provided the most stable platform for the camera and other sensors (Figure 10).

Each test examined the following:

- Video output

Figure 8. The Complete Wearable System

- Automatic startup of video and GPS systems
- Pushing of video meta-data to the database
- Pushing of GPS data to the database
- Recording of video locally on the OQO
- Streaming of video over the wireless network
- Pushing end of session data to the database
- Remote shutdown from the re-server
- Uploading of video content to re-server

6. RESULTS

6.1. Video

Initially the performance of the video left much to be desired. The wireless video experienced numerous video dropout and freeze events. Both the wired and wireless video systems were also experiencing issues with video encoding which resulted in gaps of video data missing during playback. These issues negatively affected the usefulness of the video system and caused video synchronization issues. We realized that the default rate control algorithm used by the wireless network driver on the OQO was not optimal for streaming of live video. The rate control algorithm for the wireless driver was changed to a more aggressive type to adjust for the inherent errors in wireless channels. This improved the performance of the live stream. Non-necessary services were disabled and a newer version of the Linux kernel (2.6.23.1) was installed. The new kernel resolved the missing

Figure 9. SWAT belt(black) on actor along with back pack for other sensors

video problem and drastically improved live and after action performance on the wired camera. When the new version was tested on the wireless cameras, the video performance was improved but wireless connectivity was not functional. As a result the original configuration was used on the wireless systems.

During live action, the wired video showes an average delay from real-time of less than 3-4 seconds, while the wireless system has an average delay of 3-5 seconds. For after action playback the recorded video was able to be replayed back in sync with the radio traffic, and tracking information.

6.2. GPS

The GPS system worked well and was able to insert location data into the database. The initial issue with system resources usage, which adversely affected video performance, was addressed by re-coding the GPS control system in C, rather than as a shell script. This change drastically reduced the strain on the system while still providing the capability to push location information to the database.

Figure 10. Camera on lid of baseball cap

7. CONCLUSION AND FUTURE WORK

We demonstrate an affordable wearable video system that was integrated in a larger system used for training. This solution has the core set of features that are found in higher end systems and uses a modular design for its components.

The system provides the basic core functionality of capturing location and video data. In the future we hope to leverage the capability of the portable computer to incorporate object tracking into the video system. This would allow us to track people and objects in real-time video for surveillance applications.

REFERENCES

1. S. Mann, "Smart clothing: The wearable computer and wearcam," *Personal Technologies*, vol. 1, no. 1, March 1997.
2. C. Dickie, R. Vertegaal, J. S. Shell, C. Sohn, D. Cheng, and O. Aoudeh, "Eye contact sensing glasses for attention-sensitive wearable video blogging," *Extended abstracts on Human factors in computing systems*, pp. 769–770, April 2004.
3. Y. Sawahata and K. Aizawa, "Wearable imaging system for summarizing personal experiences," *Proceedings of International Conference on Multimedia and Expo*, vol. 1, pp. 45–48, July 2003.
4. H. W. Ng, Y. Sawahata, and K. Aizawa, "Summarization of wearable videos using support vector machine," *Proceedings of International Conference on Multimedia and Expo*, vol. 1, pp. 325–328, August 2002.
5. K. Aizawa, K. Ishijima, and M. Shiina, "Summarizing wearable video," *Proceedings of International Conference on Image Processing*, vol. 3, October 2001, pp. 398–401.
6. J. Siegel, R. E. Kraut, M. D. Miller, D. J. Kaplan, and M. Bauer, "Collaborative wearable systems research and evaluation(video program)," *Proceedings of the ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM, 1996, pp. 9–140.
7. Y. Sumi, S. Ito, T. Matsuguchi, S. Fels, S. Iwasawa, K. Mase, K. Kogure, and N. Hagita, "Collaborative capturing, interpreting, and sharing of experiences," *Personal Ubiquitous Computing*, vol. 11, no. 4, pp. 265–271, 2007.
8. G. Kortuem, M. Bauer, and Z. Segall, "Netman: The design of a collaborative wearable computer system," *Mobile Network Applications*, vol. 4, no. 1, pp. 49–58, 1999.
9. R. E. Kraut, M. D. Miller, and J. Siegel, "Collaboration in performance of physical tasks: effects on outcomes and communication," *Proceedings of ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM, November 1996, pp. 57–66.
10. A. C. Teeters, "Use of a wearable camera system in the conversation: Toward a companion tool for social-emotional learning in autism," Master of Science Thesis, Massachusetts Institute of Technology, Boston, Massachusetts, United States, 2007.

11. R. A. el Kaliouby, "Mind reading machines: Automated inference of complex mental states," Ph.D. dissertation, University Of Cambridge, Computer Laboratory, Cambridge, UK, March 2005.
12. *IEEE 802.11 Working Group: Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements–Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, The Institute of Electrical and Electronics Engineers Std., July 2003.
13. VideoLAN, "Videolan - vlc media player," 2007, software Package - http://www.videolan.org/vlc/.