

Wireless 3D CAD Viewing on a PDA Device

Charles Woodward, Seppo Valli, Petri Honkamaa, Mika Hakkarainen
Technical Research Centre of Finland, VTT Information Technology
Tekniikantie 4 B, Espoo, P.O.Box 1203, FIN-02044 VTT, Finland
E-mail: charles.woodward@vtt.fi

Abstract:

This paper presents a method and its software implementation which enables engineers and other people to view and interact with 3D CAD files using wireless phone connection with a PDA device. The method is based on transmitting compressed video stream from the server workstation to the client device, with a two-way data link for user interaction and system messages. The software is optimized for real time performance even on low data transmission speeds, setting practically no limits on the CAD file size or complexity.

1. Introduction

Today's industrial production relies on (3D) CAD models, which various people including designers, engineers, marketing and sales need to view and comment at different stages of the production cycle. The CAD files are customarily transferred between different companies and departments. Often, information on CAD models is needed at customer calls or other sites where no wired data connections or even electricity exist, such as in a car or train, or just out in town. Mobile solutions for viewing of CAD files independently of location and work hours would bring significant time and cost savings to the production process by cutting short critical communication delays.

Besides laptop PC's with mobile phone data connections, PDA's (Personal Digital Assistant) such as the Compaq iPAQ, and multimedia phones like the Nokia 9210 Communicator, are becoming increasingly popular as portable communication devices. However, a major restriction to all the portable communication solutions is the limited bandwidth of the wireless communication line. Today, the fastest GSM phone/modem connections in operational use are a few tens of kbits/sec, while more typical speeds are close to 10 kbits/sec. Even in the future, the generally supported bandwidth will probably grow only by a reasonable magnitude as the building up of the operator networks is likely to be delayed and/or reduced from the original full scale plans.

In this paper we present a solution to remote 3D CAD viewing. The XVQ method applies live video compression to transfer the continuously changing CAD view at the server workstation to the client PDA device, with a two-way interaction and data message link between. The client software lets the user rotate, pan and zoom into the CAD model, as well as ask for distances, angles and radii between points in the model. Other CAD interaction commands would be straightforward to implement, as well. The client end of the video codec is very light, enabling software-only implementation. The method offers reasonable interaction speed already with data transmission rates as low as 5-10 kbits/sec, while the CAD file size is limited only by the computing resources of the server machine.



Fig. 1. XVQ Wireless CAD Viewer

An alternative portable CAD solution offered by PocketCAD, Inc. [7] enables local viewing and editing of 2D AutoCAD files on a PDA device. Some 3D solutions such as the VRML viewer by ParallelGraphics, Inc. [8] have recently been presented, too. However, the “remote” or “wireless” use of these products is based on downloading the complete CAD model onto the PDA device, which is obviously quite restrictive regarding both download time and local 3D processing speed. Also, security problems are another issue to consider when downloading sensitive CAD files.

Among other related products, Citrix Systems Inc. has introduced its server-client software for handheld PDA devices [9]. However, the Citrix solution is not tuned for displaying rapidly changing image content, thus being too bulky for CAD viewing and interaction purposes. Other remote PC control software such as those by Tarantella, Inc. [10] and by AT&T Laboratories, Cambridge [11], share equal problems when running on low speed data connections.

2. Background

MVQ (Motion Vector Quantization) is a video coding technology developed at the Technical Research Centre of Finland, VTT [12]. In the co-operation between VTT and DeskArtes Oy, the MVQ technology is applied to remote viewing of CAD image content on a PC screen, running DeskArtes’s View Expert software. Due to its strong resemblance to the X Windows of the UNIX world, the MVQ application in remote PC control is dubbed XVQ (X Windows VQ).

DeskArtes Oy develops 3D CAD software for international distribution. The company’s Expert Series product family covers various applications from free-form 3D CAD modeling and visualization to utility software for Rapid Prototyping, Simulation and CAE/CAM applications [13]. For CAD viewing purposes, the View Expert software deals with files in the international standard formats STL, IGES and VDA-FS, as well as some native CAD system formats such as CATIA [3].

3. The MVQ Method

MVQ is essentially a combination of different compression methods (see [4] for a textbook), exploiting various redundancy types with the video data – INTRA within an image, INTER between images, and entropy of the encoded bit stream. The best applicable method for each part of the image is chosen dynamically during encoding.

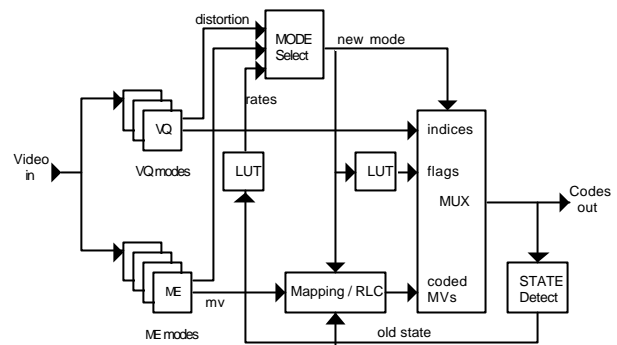


Fig. 2. Block diagram of the basic MVQ encoder.

The MVQ method is very light and fast at the receiving end, as most of the decoding is done with just look-up retrievals. It is thus especially well suited for implementation on low performance receiving devices. Several variants and implementations (incl. Java) have been developed for MVQ. In the described application, a basic version of MVQ [1, 2] has been used – see Fig. 2.

Basic MVQ uses two variations of VQ with different codebook sizes for INTRA coding of 4x4 pixel blocks. Totally three INTRA modes with fixed bit rates are utilized. Motion compensated blocks are used in INTER coding without refining, i.e. prediction error is not coded. The resulting lack of accuracy is compensated for by using 8x8 block size and half pixel accuracy in motion compensation. The resulting motion vector field is segmented and quantized in order to reduce motion information. Background prediction is used to store and predict stationary image areas being uncovered. Extended motion estimation and median prediction for motion vectors is used to predict the location of moving objects.

Entropy coding is accomplished by run-length coding (RLC) in order to capture also large scale correlation. Variable length coding (VLC) is not used so that fixed length code words of eight bits result from the coding process. Bit rate is controlled by switching between different coding modes.

The efficiency of MVQ stems from three main features:

- ✍ use of Hierarchical Classified Vector Quantization (HCVQ) method for the INTRA update information (- US Pat. 5692,012)
- ✍ good utilization of interblock dependencies by the quantization and segmentation of motion vectors (- European Pat. WO0101692 pending)
- ✍ coding mode selection using the actual rate-distortion performances of different options

The main differences compared e.g. to the standard H.263 are in the way motion information is obtained, segmented and presented in MVQ. The above mentioned background prediction for stationary image areas is very useful and is not included in the standard. The maximum noticeable speed of motion between successive image frames (i.e. the maximum search area in motion estimation) is considerably higher in MVQ than in H.263. Finally, and perhaps most importantly, the implementation of MVQ is far more simple than that of H.263.

4. The Wireless CAD Viewer

The XVQ Wireless CAD Viewer consists of three software modules, the XVQClient on the client PDA device, and the XVQServer plus DeskArtes View Expert on the server workstation. The following gives an overall description of the CAD viewing process:

1. The user creates a phone connection through a modem or Internet portal from his/her PDA device to the server workstation.
2. The user starts XVQClient on the PDA device. The XVQClient user interface pops up, sending its screen size (width and height of the image area) to XVQServer.
3. XVQServer launches View Expert on the workstation. The most recently used CAD model is automatically loaded to View Expert.
4. The user gives viewing commands using the command buttons in the XVQClient user interface. Some commands ask for graphical interaction, which are executed by clicking at the CAD image with the pen. A separate command is available for loading new CAD files into the workspace.
5. XVQClient sends the commands in ASCII form to XVQServer. The commands are passed to View Expert as Windows messages.
6. XVQServer continuously grabs and encodes the image on the View Expert display and streams the compressed image to XVQClient. Possibly, also distance query information is sent in ASCII form to XVQClient.
7. The steps 4-6 are repeated until the user closes XVQClient (and optionally the phone connection as well).

The interaction commands with their respective ASCII names are shown in Table I. There, $\langle x y \rangle$ denotes two coordinate values relative to the XVQ client display area, i.e., the pixel location where the user pointed with the pen. View Expert converts the location to its internal coordinate system based on the known XVQClient screen resolution.

Table 1. List of XVQ / View Expert Commands

LOAD <filename> - displays a file browser to load new models to View Expert
FIT - fits the model in the middle of the screen
VIEW X / Y / Z / 3D - sets the viewpoint to one of the axes, or back to 3D
SPIN - spins the model continuously around the vertical axis
ZOOM IN - zooms closer to the viewing center
ZOOM OUT - zooms away from the viewing center
ZOOM <x y> TO <x y> - zooms to the area between two shown points
PAN <x y> TO <x y> - pans from point to point, as shown on the screen
ROTATE <x y> TO <x y> - rotates the model by an angle determined by two points
DIST <x y> TO <x y> - reports the 3D distance between two points on the model
ANGLE <x y> TO <x y> TO <x y> - reports the angle of three points on the model
RADIUS <x y> TO <x y> TO <x y> - reports the circle radius of three points on the model
STILL - displays a still JPEG image on the screen
END - closes the View Expert session

All the commands are implemented as icons – cf. Fig. 1. Some of the commands have pop-up dialogues associated to them, for example for controlling the speed of spinning the model. File access is implemented with a Windows-like browser, with file and directory names locally uploaded to the client device for fast graphical interaction.

5. Software Architecture

XVQ has been written in C (libraries) and in C++ (user interfaces). XVQServer runs under Windows and it can be used on any common PC workstation. XVQClient is implemented in the Windows CE environment and it is currently compiled for the Compaq iPAQ and Casio Cassiopeia devices. In addition to XVQ applications, also DeskArtes View Expert required some programming to allow the usage of external control commands.

Both XVQServer and XVQClient rely heavily on dynamic libraries. All constantly needed services like the MVQ compression codec and picture handling utilities are located in statically loaded dynamic libraries. The application specific parts, like communication with the application and the client user interface, are also separated in external libraries but this time as dynamically loadable drivers.

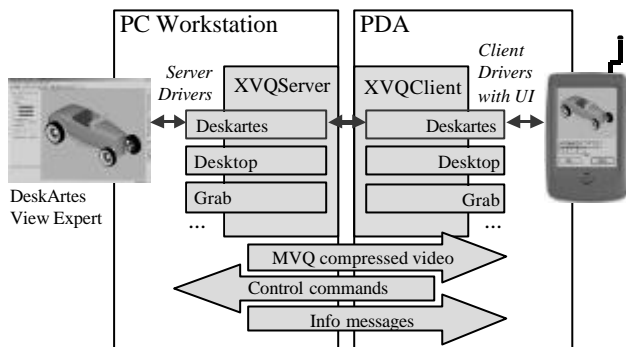


Fig. 3. XVQ software architecture.

The usage of dynamic drivers makes it possible to communicate not only with CAD software but with any software that allows external control commands. The change of the external application can be made on the run, just by changing the drivers. And as the client user interface is also contained within the driver, the PDA view can always be optimized for the application in use. The dynamic change of the drivers was tested by making two additional drivers: the first for viewing the desktop area of the PC workstation, and the second for controlling and capturing the view from an external video camera – see Fig. 3.

The communication between the XVQServer and the XVQClient is made through a TCP/IP connection. Two sockets are used: one for streaming the MVQ-compressed application view from server to client, and the other for application specific control commands and info messages in ASCII form. This way only the application specific client driver needs to be updated if the set of commands is changed. The server driver won't need any update, as it delivers the commands unaltered to the application.

The CAD model view is captured by copying the bitmap from the View Expert window. XVQServer resizes the view and uses the MVQ codec to compress it for low bitrate video stream. The control communication between the XVQServer and View Expert is made using Windows messages. The ASCII strings used for this (listed previously in Table I) are the same ones used also for the communication between the XVQServer and XVQClient. With this approach the XVQ software is highly transparent: in effect, the client user interface communicates straight with the CAD viewer on the PC workstation.

6. Image Quality and Speed

The current XVQClient implementation on the Compaq iPAQ H3635 handheld uses 4086 colors and 224x168 pixels (full width of the screen, less margins) to

display the CAD image. XVQServer running parallel with View Expert on a 800 MHz PC workstation grabs and compresses the video stream at an average rate of 10 frames/sec.

The frame rate that can be obtained at the receiving end depends linearly on the data channel bandwidth. According to our tests, a wireless phone connection with 14.4 kbits/sec yields the speed of five frames per second. Even much slower connections provide sufficient speed to be regarded, for the purpose of the application, as “real time.”

Good image quality has been reached by training the XVQ codebook with screen shots of different CAD models. As an inherent feature of MVQ, the image sharpness is gradually refined when there's little movement in the image. This happens, in particular, when the user just views the PDA screen without interaction.

After an interaction pause over three seconds, the client screen is automatically updated with a JPEG compressed still image, and image transmission is halted until the next user interaction takes place. The user is also given the option to request for the JPEG image more immediately if required.

With rapidly changing image content, e.g., when spinning the model, XVQServer is requested to provide high frame rate with less quality. This, too, is a readily available feature of the basic MVQ method.

Future implementations of XVQ will be built using an advanced version of MVQ, which provides a more modular and flexible coding structure for application specific optimization. While panning, for example, simple motion estimation should be favored among the different coding methods. Also, specialized motion compensation methods for the model's rotation and zooming [5] may be considered for future implementation.

7. CAD Conferencing

XVQServer actually launches DeskArtes View Expert on the server workstation screen, and there's nothing to prevent View Expert from accepting commands from the keyboard and mouse. Thus, two people can work on the same model simultaneously, one of them having started View Expert with the PDA and the other one sitting at the workstation. User interaction by either one takes place on the same View Expert workspace. For example, design changes performed by the workstation user are immediately seen by the PDA user, who can then choose any view angle to look at the changes.

CAD conferencing between multiple users could be implemented with a Internet server modem pool, transmitting the CAD image to each participant's screen. Interaction control would be delivered to one participant at a time upon request. The participants would be equipped with different hardware and data connections according to their location, e.g. laptop PC, PDA, Java-enabled phone, Internet, etc.

8. Conclusions

According to our knowledge, XVQ is the first implementation in the world dedicated to wireless 3D CAD viewing on a PDA device [6]. The method is independent of the CAD file size, providing good interaction speed and image quality already with low data transmission speeds. Besides PDA, the XVQ method is equally useful on other portable computer equipment, also enabling CAD conferencing between multiple people.

The current implementation of XVQ incorporates the basic functionality for 3D CAD viewing. Future versions will be enhanced with various features, such as new coding modes, color selection, new dimensioning commands, clipping planes to view inside to the model, chat, audio and so forth. Besides just viewing, the XVQ technology can further be utilized in CAD modeling, i.e., interactive design of curves, surfaces and solids, and many other PC desktop applications as well.

References

[1] S.T. Valli, "Very low bitrate coding using hierarchical classified VQ and cluster based segmentation of motion information", paper No. 7.3, *Proc. VLBV94*, Univ. of Essex, UK, 1994, 4 p.

- [2] J.T. Virtamo and S.T. Valli, "Vector quantization with hierarchical classification of sub-blocks", *Signal Processing: Image Communication* 8 (1996), pp. 475-479
- [3] K. Lee, *Principles of CAD/CAM/CAE Systems*, Addison-Wesley, 1999
- [4] Y.Q. Shi, H. Sun, *Image and Video Compression for Multimedia Engineering – Fundamentals, Algorithms and Standards*, CRC Press, 2000
- [5] N. Rodrigues, V. Silva, S. Faria, "Hierarchical Motion Compensation with Spatial and Luminance Transformations", in *Proc ICIP 2001*, Vol. III, pp. 518-521.
- [6] G. Smith-Moriz, "Mobile STL Viewing", *Rapid Prototyping Report* 10 (6), June 2001, p. 11

WWW Links

- [7] <http://www.pocketcad.com>
- [8] <http://www.parallelgraphics.com>
- [9] <http://www.citrix.com>
- [10] <http://www.tarantella.com>
- [11] <http://www.uk.research.att.com/vnc>
- [12] <http://www.vtt.fi>
- [13] <http://www.avstechnologies.com>
- [14] <http://www.onair.dk>
- [15] <http://www.deskartes.com>