Understanding Multimedia

Multimedia is using more than one sense to convey information. Television is multimedia. Images of the latest Amtrak accident reinforce and make vivid the verbal description read by the reporter. Universal Studio's *Back To The Future* ride is multimedia. It pulls the rider into a world where that doesn't exist by loading the body with vibration to indicate he is moving, video to confirm and provide direction to the motion, and sound to give the vibration and images realism. That the rider is going nowhere is lost in the heavy load of coordinated data filling the input channels.

Personal computers are not multimedia. Their data are visual and, within this set, usually text. Color and graphics tend to be limited to helping the user navigate among options. Color separates the title bar from the menu bar. Non text data is still visual, usually limited to graphics on icons, relatively simple images to supplement text, and, of course, to illustrate numeric data via the famous bar and pie charts.

In the traditional multimedia of television and movies data are "uncompressed" analog data. Actually, the data are compressed to fit within a specified bandwidth. The "compression" occurs as data are discarded to encode the audio and video in a broadcast format, NTSC for the U.S., Canada, and Japan, for example, and PAL for much of Europe. For one second of user experience one second is required to transmit the data. Copies degrade quality. Editing is clumsy. Bandwidth limits the number and flexibility of distribution channels. By using computers to digitize data some magic things happen:

- Ordinary personal computers can be used to store, retrieve, edit, copy, and display data other than text just as they can text data.
- Limitless copies can be made with no loss of quality.
- Existing distribution channels can carry several times the analog content.
- New distribution channels become available, including telephone lines, CD-ROMs, corporate networks.
- New uses of computers appear, such as video conferencing, training, high quality video presentations and entertainment.

An entire industry is emerging devoted to using digital techniques, software, and hardware to provide end users the sensations of multiple, simultaneous, coordinated, data input. This industry produces television shows, commercial movies, training materials, games, tools for application development and content editing, and special hardware for data acquisition, storage, and rendering.

Because the focus is on content and the importance of playing that content on as many devices as possible this industry is intensely focused on standards. Some of these standards:

- **JPEG** - Joint Photography Experts Group standard for still image compression. Part of the compression of JPEG images is the discarding of color information the human eye is not likely to notice is missing. Both software and hardware support for JPEG are available for Windows.
- **MPEG** - Motion Pictures Experts Group standard for full motion video. MPEG includes compression of near CD quality audio. There are several types of MPEG, of which two are currently important to Windows PCs:
    - **MPEG-1** (or plain MPEG) - current television quality video with near CD-quality audio at standard CD-ROM data rates of 150 kilobytes/second. MPEG is supportable in software under Windows but is limited to 6 to 8 frames per second. Except when Windows is run on certain high performance RISC processors television quality MPEG video requires hardware support to achieve 30 frames per second.

    - **MPEG-2** - higher data rate MPEG that is capable of supplying HDTV quality video and can also drop a CD-ROM data rate of 150 kbytes/second. MPEG-2 is a likely default codec for digital HDTV.

- **Px64** - a family of video and audio codecs specifically defined for compressing and decompressing video conferencing data. Playback of MPEG data holds quality constant in order to preserve the viewing experience. Px64 allows variability in the quality of video and audio in order to preserve intelligible communication over variable bandwidths, as might be found in enterprise networks.
- **QuickTime Movie File** - a file format containing Apple QuickTime format multimedia data objects. The format is popular with titles developers because, with some restrictions, a QuickTime Movie File can be played on a Windows PC with Apple's QuickTime for Windows.
- **Video for Windows .AVI File** - the most popular movie file format under Windows and OS/2. Can contain a number of multimedia objects. VfW movie files consisting of uncompressed audio and Cinepak or Video 1 compressed movies can be played on the Macintosh under Apple's QuickTime.

This industry uses multimedia-capable operating systems from Microsoft, Sun, Silicon Graphics, IBM, and Apple. Within the spectrum Microsoft Windows is the most common OS for multimedia applications for personal computers. It leads Apple's System 7 in the number of content preparation and editing tools and in quantity and variety of special hardware. But despite parity with Apple in capability the Macintosh leads as the development environment of choice for titles developers. There are two reasons for this. The first is historical. Content developers have been successfully courted by Apple for many years. The second is that there are two distinct multimedia playback markets, the Macintosh and Windows. Apple has promoted the Macintosh as the development environment that produces content that runs on both Windows and Macintosh PCs. There is some truth to this in that some QuickTime data types will play on both the Macintosh and Windows, and that a few tools, most notably Macromedia's *Director*, have content run-times for both the Macintosh and Windows. The Windows to Macintosh cross platform story is neither as clear nor as well promoted.

To pull ahead of Apple Microsoft must have more than parity. We must have compelling technology that exceeds that offered by Apple and we must provide either easy cross-platform access to the Macintosh or have a Windows market story so strong the Macintosh becomes irrelevant.

The intention of this paper is to convey an understanding of multimedia, the demands it puts on an operating system, OS implementations of multimedia by Microsoft and our principal competitors, an analysis of our competitive position, a recommendation for areas of ownership under our 32-bit operating systems, and recommended next steps.

Summary of Conclusions:

The main points and conclusions of this paper are:

- Multimedia data types are unlike other data types Windows handles. This places special requirements on the operating system, the PC, and peripherals. Windows 3.1 does not handle multimedia particularly well. Windows 95 and NT offer opportunities for improvement.
- Lots of "standards" exist for data storage, containers, compression, and streams. Each targets some particular use of multimedia or some specific target market. Windows as an authoring environment needs to support several of these. Windows as a playback environment gives Microsoft an opportunity to specify some of these.
- The architecture of Microsoft's multimedia is similar, but not identical, to that of Apple's QuickTime and IBM's MMPM. Apple does a superior job of presenting the architecture in terms developers can relate to the problems of creating applications and titles. IBM addresses synchronization more explicitly. All three have just about reached the end of their

capabilities. An opportunity exists for Microsoft to leap beyond QuickTime by adoption of OLE 2. This is planned for post Windows 95.

- Windows leads every other platform for multimedia tools and hardware options in all but broadcast quality video editing and high-end graphics. These will be addressed with NT and Windows 95.
- Displacing the Macintosh as the development platform of choice for titles developers requires the following:
    - Large and growing installed base of Windows multimedia-ready PCs.
    - Compelling technical reasons to choose Windows over the Macintosh.
    - Compelling story well and thoroughly distributed to ISVs, IHVs, and OEMs.

Understanding the multimedia data types

Microsoft's first multimedia release, the Multimedia Extensions 1.0, focused on audio data. Within audio were "waveform", an actual data representation of sound waves, and synthetic sound, created by special purpose audio integrated circuits. The first release of the Multimedia Extensions for Windows 1.0 contained support for these, some tools for manipulating and displaying still images, a simple, extensible, language for manipulation of devices that are sources of audio and video information, and some utilities for streaming data from a CD-ROM. The belief was that third parties would want to add audio and image data to CD-ROM titles.

This belief has, in fact, turned out to be accurate, as the hundreds of multimedia titles and games available today for Windows show. But it did not crisply anticipate digital video, nor, for that matter, the potential advantages of anticipating unspecified of multimedia datas.

The emerging industry standard for a motion video data stream, MPEG, is an example of such a new data type. While MPEG contains video and audio but is not purely one or the other. An MPEG stream has two characteristics that are different from those assumed in the original multimedia data model:

1. An MPEG stream consists of interleaved video and audio data. The rendering application has the responsibility of separating, rendering, and synchronizing the individual streams.

   The original Microsoft extensions assumed a data stream was purely of one data type. Handling audio and video interleaved streams has been added to Microsoft multimedia with Video for Windows.

2. The process of compressing the video in MPEG is not unidirectional. Until MPEG Microsoft multimedia assumed data were compressed beginning with the first bit and working through to the last. MPEG, however, provides compression options to optimize image quality within a restricted data rate. For video this means the compression of a particular frame can be influenced not only by the frames that came before but also by the frames that come after.

   Neither QuickTime nor Video for Windows explicitly support this model, nor does either recognize the MPEG data stream. But Video for Windows is designed to have these added easily by third party or ourselves. Several third parties are demonstrating MPEG compression for Microsoft multimedia and will be shipping product this coming year.

The purpose of this mention of MPEG is to illustrate a complete set of multimedia data types is difficult to specify. In addition to MPEG there are many other data type possibilities, including captioning mixed with animation, video teleconferencing, and even instrumentation data. A general definition of a multimedia data type is required and, with it, an architecture that allows for it's implementation.

Characteristics of a Multimedia Data Type

Time dependent - Failure to render each bit of data at precisely the right time results in loss of meaning to the user.

Ordinality - Arrangement of data within a stream is unidirectional, with the first bit first and the last bit last. It need not, however, be cardinal. That is, a seek to the middle of a data stream is not necessarily a seek to the time middle of the data.

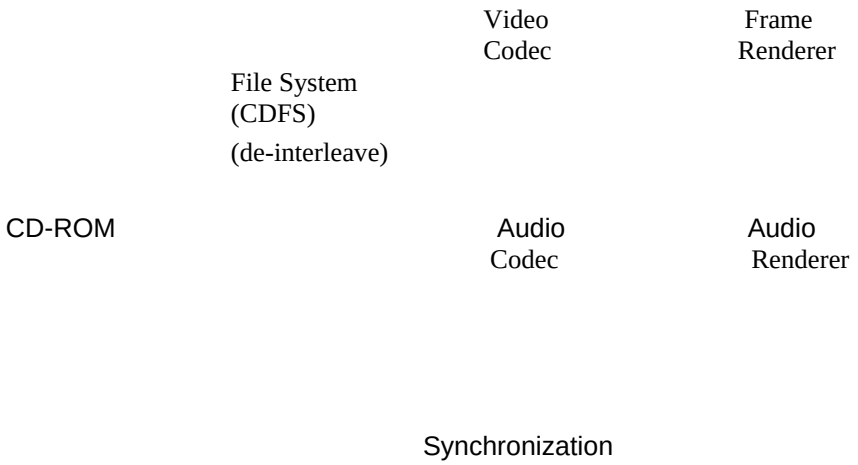
Synchronization - To yield its meaning the rendering of each bit of data from one multimedia data stream may have to be precisely synchronized with the rendering of the bits of data from another.

Data without end - In some situations a data object must be acquired and interpreted without reference to a beginning and an end, as with a video conferencing data object.

Multimedia hardware requirements

The characteristics of multimedia data have implications for every part of the multimedia rendering process. For illustration consider the task of playing a movie on CD-ROM with television quality video and CD quality stereo audio. Figure 1 diagrams the components of this system as it is now being built for personal computers and for some consumer products.

Figure 1 - Playing Frame-based Movie with Synchronized Audio



Storage:

In Figure 1 the storage media is CD-ROM. This is only one of several storage media proposed for commercial digital movies. Others are Sony MiniDisc, digital tape, network server, and video on demand (VOD) server. VOD gives the user all the start, stop, reverse, and scan options offered on a video tape recorder. Microsoft's *Tiger* technology is an implementation of VOD on standard PCs running NT.

CD-ROM advantages:

- **High data density on an inexpensive medium**. At 150 kbytes/second transfer rate an MPEG CD-ROM will hold about one hour of content. Dual or double speed drives promise better image quality at lower play time capacity. Higher density drives are on the horizon but are not

yet available, even for evaluation. For movie playback on CD-I Philips is experimenting with a drive that holds two discs.

- **Inexpensive to manufacture**. The companies that own content (Sony, Disney, Warner) also own audio CD fabrication plants. There is virtually no investment in facilities required to make CD movies.
- **Reliable and inexpensive drives**. Early in the development of the multimedia extensions drives and drivers of sufficient efficiency to leave enough CPU cycles to render the content were rare. Now nearly every CD-ROM drive exceeds our multimedia performance standards.

CD-ROM disadvantages:

- **Slow seek time**. At up to one second . This introduces the following complications:
    - Uninterrupted video or audio play requires uninterrupted data reading. This consumes CPU resource or requires special hardware. The former is the option adopted by Microsoft multimedia, the latter by special hardware used to read the Philips/Sony format CD-ROM XA.
    - Applications that require frequent random access, such as a data base, perform poorly. To minimize this problem database applications, such as the Microsoft MM Pubs titles *Encarta* and *Cinemania* place data location information on the hard disk or load it into memory.
- **No write capability**. This means data like top scores for a game or notes on an encyclopedia entry are stored separately from the application. A CD-ROM then moved to another machine goes without this additional data.

It should be noted here that Microsoft multimedia assumes the ISO 9660 format for CD-ROM data, and does not support CD-ROM XA or other so-called “mode 2” CD data formats. If Windows is to create or read discs in the Philips **Video CD** format this will have to change with Windows 95. The magnitude of the change depends on how efficiently the data are to be rendered. We are happy to talk with anyone in Systems about the options.

De-interleaving:

“De-interleaving” is a useful mental concept. It implies a computer function that, through some sophisticated software, produces continuous streams of audio and video data from whatever was read from the file system. In fact, however, this is not what happens. Data are streamed from the source. Each homogenous chunk is routed to its process, or “handler”, a piece of code that knows how to process it. The CPU utilization for of this is very low. The purpose of the interleaving in the first place is to make certain the next chunk of data arrives before the last chunk of the same data is finished being rendered.

The diagram shows audio and video streams. Candidates for other streams are MIDI, captioning, animation, other video streams, and foreign languages. The data streams are not in synchronization at this point. The purpose of interleaving is to make later synchronization possible.

How data are interleaved depends on the requirements of the media. Because seeking on a CD-ROM is undesirable audio data for each video frame is closely interleaved between each frame. Video to be read from a hard disk on a server, however, has data loosely interleaved. A “chunk” of audio will be read and, while it is playing, the video frames are retrieved. This approach has advantages should network bandwidth drop. Under low bandwidth not all video data can be rendered. The audio plays without breaking because a large piece of data was read. Video frames that can’t be transferred in time are “seeked” past, meaning network bandwidth is invested only in those frames that can be delivered in time to be in synchronization with the audio, making the most efficient use of the network.

Each data stream is then made available to its handler. Video for Windows, QuickTime, and MMPM treat data streams similarly. A data stream is processed by a piece of code specifically written for that type data. Video for Windows ships with “handlers“ for waveform audio and frame-based video.

Frame-based means the motion the viewer sees is created by rapidly drawing a series of still images. This is different from animation, where motion is produced by executing graphics drawing instructions. Third parties can add handlers for their other data types and we may be shipping a handler for MPEG in Windows 95.

Decompression:

The data handler can do just about anything with the data it desires. The steps shown in Figure 1 are decompression and rendering. Microsoft multimedia supports compression of both audio and video. We can provide details of audio compression to those that are interested. What inhibits extremely high quality video on PCs is bandwidth. And video data is by far the largest of the multimedia data streams. So special attention is paid to video compression. To understand video decompression it is helpful to understand a few terms.

Compression Terms:

Key frame - a video frame that, if uncompressed is complete and viewable. And when compressed it is compressed entirely based on the data it contains. That is, it can be decompressed and displayed without any information from a preceding or following frame.

Delta frame - a video frame that contains only the information that has changed from the frame before.

Intra-frame compression - compression of video without knowledge of other frames. Intra-frame compression is used to compress a key frame.

Inter-frame compression - compression of data differences between video frames.

There are several compression technologies in common use. These appear in the video codecs used with Video for Windows.

Compression Technologies:

RLE - run-length encoding. Well known in Microsoft for compressing bitmaps this is a technique of describing data by a bit pattern followed or preceded by a number of times to repeat it. PSS has offered for years an application note on how this works for Windows bitmap images.

DCT - The DCT is the basis of the JPEG and MPEG compression standards. Using the discrete cosine transform, an image is broken up into low and high frequency components. These frequency components are based on how quickly such things as luminance and chrominance vary from pixel to pixel. The eye is not as sensitive to high frequencies, so these components can be removed, giving some compression. If a particular frequency component has some range of values, subranges of values may be represented by fewer bits. For example, a range may use 256 values normally. If we split up that 256 value range into 4 subranges, where each subrange would be represented by some value found in that range, we have gone from 8 bits to 2 bits for 4 to 1 compression. This technique has been made highly efficient. Compression in MPEG, however, can be very slow since delta frames are expensive to create (see **Motion Compensation**).

Motion Compensation - MPEG uses a technique called motion compensation to create its delta frames. The current image is split up into blocks. The compressor then looks for these blocks appearing in certain previous or future frames. If a sufficiently close block can be found, directions on how to get to that block are stored, rather than the block itself. An error block can also be stored which gives the difference between the matched block and the current block. This search for a matching block can take a large amount of time, which is why MPEG compression can be quite slow.

Wavelet - Wavelets are somewhat like a DCT. An image is broken up into components sort of like the frequency components of the DCT, but they aren't really frequencies. A large number of

DCT frequencies are needed to capture a sharp edge, so removing frequencies in the DCT can cause visual artifacts. Wavelets are much better at capturing edge information with fewer "frequencies", so can throw away more "frequencies" while not causing as many artifacts. This way wavelets can get higher compression rates than DCT-based algorithms with fewer artifacts. This comes at a higher computational cost than the DCT, and more efficient ways of calculating the wavelet transform are being found.

VQ - Vector Quantization. In VQ, a small number of patterns are found which represent the patterns of pixels found in the image. Usually these patterns are all blocks of the same size. A dictionary table of these patterns is created. The image is broken down into a set of blocks and each block is replaced by the table index of the pattern most closely resembling that block. This compresses the image because the index is represented in a smaller number of bits than the original pixel data for the block. Decompression is very quick since it is merely a table lookup. Compression can take a very long time since it is usually very hard to come up with an appropriate set of representative patterns.

Fractal - Fractal compression is based on the assumption that a lot of images contain the same sub-image over and over again. The sub-image may be rotated, or stretched, or have any of a small class of transformations applied to it when it appears again in a different place, but it is still more or less the same image. So fractal compression locates these repeated sub-images, and stores where the sub-image resides and what transformations are necessary to get it from the original sub-image. It is assumed that the location and transformation information takes up less room than the original image, and this usually holds true.

Video for Windows ships with four video codecs. Other codecs are shipped by third parties.

Codecs Shipped with VfW:

RLE - intended for compressing clean graphic images such as animation of bar charts. Has low CPU overhead. Does not handle rapid, complex scene changes well.

Video 1 - developed by Microsoft, patent applied for. This codec is low CPU overhead, good for full motion moderate quality video. Compresses quickly.

Cinepak - licensed from Supermac. This codec provides the best looking and the best video playback performance, typically 320 by 240 images at 15 frames per second or better. Compression times are very long, typically 12 to 16 hours for 10 minutes of finished video. Cinepak is a common video codec for CD-ROM titles for both Windows and the Macintosh. It is the codec used on Microsoft *Dinosaurs* and *Cinemania*. Cinepak runs on Windows, Windows NT, and soon on Windows NT for MIPS and for Alpha.

Cinepak is also available on the Macintosh. Supermac has licensed the technology to video hardware vendors for inclusion in future video subsystems.

Indeo - developed by Intel. Indeo has undergone significant change since its introduction a year ago. What started out as a mediocre software codec (160 by 120 at 15 fps, fuzzy images) now looks about as good as Cinepak. Compression is faster and Intel sells an inexpensive video capture adapter with built-in hardware compression for real-time capture and compression. Indeo is gaining in popularity.

Indeo will run on Windows NT for x86 machines. Intel is writing a 'C' version of Indeo to allow Microsoft to port it to other NT processors, if we desire.

Rendering and Synchronization:

The break between decompression and rendering introduces a performance problem for most PCs. Until the data are decompressed the bandwidth limitations of the ISA bus have not been particularly important. But the data rates required for large, full color images at motion picture frame rates exceed the bus' capabilities.

Local bus video helps performance greatly in that massive changes to video can be made quickly. Local bus is inexpensive and, even with hardware-assisted decompression, is essential for high performance video playback. There are other rendering improvements, including hardware acceleration of routine graphics operations. The ATI Mach 32 accelerates stretching, for example. Another is to combine hardware decompression with rendering. Cirrus is planning a PC video subsystem that combines SVGA graphics with hardware support of Cinepak decompression.

Rendering is also where synchronization takes place. In this example there are two rendering engines, one for audio and one for video. The data each receives is uncompressed. As a result each knows its own "local" time. At 24 frames per second, for example, if the video renderer is at frame 405 the local video time is 16.875 seconds. The audio renderer also has a local time. If the sample rate is 22 khz and the renderer is at sample 371,250 the local audio time is also 16.875 seconds. In this case audio and video are in synchronization.

Synchronization Problems Caused by Audio Clocks

Audio breaks are unacceptable. For this reason video is almost always synchronized to audio. Audio local time is the clock used to synchronize the video. Audio time is determined by keeping track of how much audio data has been passed to the audio subsystem and where in the last packet the rendering is. Overall playback accuracy, then, is a function of the accuracy of the audio clock. Typical audio cards have clock variations between samples of up to 10%. This has three unfortunate implications:

- Audio samples are captured with a clock error, meaning playback time doesn't automatically equal capture time. Since video is captured through a different channel this introduces synchronization error at capture time.
- Audio samples played back on different audio adapters will play at different speeds, reflecting differences in clock accuracy, meaning the entire video will play at different speeds.
- Synchronizing audio with some other process that also can have no breaks, for example audio played through a second audio adapter, is nearly impossible.

There are methods proposed for dealing with this. One, which has been discussed with both Media Vision and Creative Labs, is to provide a clock adjustment buffer. This would allow the operating system to set some other clock as master and adjust the playback of audio to fit that clock.

If a display subsystem is unable to keep up with drawing demands a codec must have some way to keep what video is displayed in synchronization with audio. The most usual is dropping frames, by refusing to display them, by refusing to decompress them, or by skipping reading them from the source file all together. In this example each time the video renderer receives CPU attention it asks the audio subsystem for the local audio time. It then translates that time into which frame should be displayed. If that frame is the next one queued it is displayed. If the renderer is behind it skips frames to catch up. If the time has not arrived to change the frame it does nothing.

An important understanding here is that degradation can involve coordination all the way back through the data chain to which data are read from the file. For this to function efficiently the codec, the renderer, and the handler need to cooperate. Each of these is installable under Video for Windows, meaning third parties can provide their own.

Other Multimedia Hardware Components:

In this example we have discussed the CD-ROM, audio adapters, and hardware-assisted video decompression and rendering. There are a number of other multimedia components to complete the hardware set.

Video Capture Adapters and Hardware Compression:

Video data has to be acquired before it can be digitized and compressed. Video capture adapters to this. Typically an adapter takes analog video input and converts it to digital images. This allows the user to plug in a number of familiar video devices, including laser video disk players, video tape players, and video cameras.

Captured raw data takes up huge amounts of disk space. To minimize disk requirements data is often compressed as captured.

Video capture occurs one of two ways, real time or step frame.

Real time capture:

Real time capture means video data are digitized and written to disk as fast as the data are received by the adapter. This is useful for video conferencing and for capturing video data from devices incapable of rendering high quality still images of each video frame. Unfortunately, that's nearly all consumer video tape recorders and video cameras.

Captured data is usually written directly to disk. If it is uncompressed the data transfer capabilities of the system bus and disk system are stressed, meaning that after all buffers are filled data are lost. Some capture adapters, however, compress in real time as well. The most common is Intel's *Smart Video Recorder*, which captures and compresses with Intel's *Indeo* codec.

Step frame capture:

High quality compression with codecs like Indeo and Cinepak require intense examination of each frame and its relationship to the frame before it. Doing this in real time requires special compression hardware. If it is not done in real time then there are two options. Either each raw frame is already in a disk file, which requires lots of disk space, or it is read into the capture adapter one frame at a time. The latter is step frame capture and it works only with devices that are capable of displaying single frames and can be controlled by computer. VfW supports two such devices, laser video disk players from Sony and Pioneer and video tape recorders that can be controlled by Sony's command language VISCA.

CD-R and Other Writable Optical Media:

Most multimedia titles are published on CD-ROM. Philips Corporation has a number of "standards" it promotes, usually in cooperation with other consumer electronics companies, particularly Sony. These are not independent standards, in that each is independent of the other. The White Book, for example, implements technology specified in the Yellow Book for support of MPEG movies.

CD-R stands for Compact Disc - Recordable. It is a CD-ROM that will allow PCs to record their own CD-ROMs. Currently these are available from JVC, Sony, Philips, and others and sell for between $4,000 and $10,000. In two years the price is expected to drop to about $500.

A CD-R drive will record data on the CD in any of the standards, meaning a titles publisher can prototype a title on his own PC. But CD-R also opens the possibility a user might want to add to the data already recorded on a recordable CD-ROM. This is called "multi-session" recording and how this is to be done is specified in the Orange Book standard.

There are other storage devices that have the characteristics of high volume and reasonable performance necessary to make them candidates for multimedia data. Sony's MiniDisc is capable of recording multiple times, something like a computer's hard disk. Magneto-optical disks have similar capability but with better random access performance. And some third parties are proposing putting digital movies on digital tape cassettes.

Microsoft multimedia is capable of working with each of these devices but has no services specific to the characteristics of any. This may change for CD-R as it is used to support Kodak's PhotoCD.

Laser Video Disk:

Laser Video Disk players produce high quality analog television images, PAL or NTSC usually. They are connectable to video capture adapters.

There are two data formats for laser disk, CLV (constant linear velocity), and CAV (constant angular velocity). CLV stores nearly twice as much data on a disk as does CAV. But CAV has the advantage that one revolution of the disk equals one full image frame. As a result, CAV can be used as a step frame video capture source. Many titles developers have one-off laser video disks made of their content solely for the purpose of step capturing it.

Video Tape:

Any video tape source capable of providing analog video out can be used as a capture data source. But only those capable of providing single frames can be used for step frame capture. These are available from several vendors for VHS tapes and Sony has a family of decks for their high quality Hi-8 format.

Interactive Controls:

These include joy sticks, power "gloves", etc. Currently Microsoft support of these is minimal.

Special-purpose Video Display Subsystems:

This is a very special issue. On PCs targeted for Windows 95 and NT nearly all the video subsystems are VGA or derivatives such as SVGA. Many are capable of high quality color but the family is not optimized for multimedia.

VGA is not optimized for video images: The VGA standard is based on the assumption that all the colors people care about can be made by mixing red, green, and blue light. And for most purposes this assumption is accurate. It is, in fact, how color is rendered on a CRT and a television, as holding a magnifying glass close to the screen will show.

Where inefficiencies appear is in how accurate colors are represented in memory for the VGA. Briefly, there are two color modes, choosing and mixing colors from a palette of 16 for VGA and 256 colors for SVGA, and representing the intensity of a color by allocating a certain number of bits for the intensity of red, green, and blue for each pixel. The latter scheme is called "true color" and produces by far the better image.

Within true color are two schemes. The first is 16 bits per pixel, where 5 each are used to specify the level of red or blue and either 5 or 6 is used to specify the level of green. The second is 24 bit color, where a full byte is used to specify each level of red, green, and blue. This is called 24-bit RGB and produces color gradations that are indistinguishable from real life. 16-bit RGB, on the other hand, produces extremely good color but experts can often detect artifacts caused by truncation of the data.

The good news about true color is its accuracy. The bad news is that it is not closely aligned with how video data are compressed. This is because video compression is lossy. The art in compression is to pick that data the eye can't detect or won't miss and discard it. RGB gives a full byte to each primary color but the eye distinguishes brightness (luminence) from color (hue). Between these it is much more sensitive to variations in brightness than in color and, within color, is much more sensitive to red and green than to blue. A common compression technique is to preprocess video data by taking RGB, performing a matrix operation to produce a luminance vector and two color vectors, and throw away much of the color detail. There are two common transformations, YUV, used for computer images, and YIQ, used for commercial NTSC color television. Color television has less than one half the color resolution as black and white resolution.

This means a video adapter tuned for video presentation could produce images as good as 24-bit RGB but with less memory than 3 bytes/pixel if it only stored data in YUV format instead of RGB. Nathan Myrvhold has a paper on this subject, *Full Color at Half Price*.

VGA and its access through GDI does not include means for rapidly updating image pixels: In order to spend as little time drawing images as possible only those pixels that change from one frame to the next are updated. VGA has hardware support for pixel update but the overhead through GDI is high. GDI was not designed to do rapid pixel changes. This is the major stimulus for the VDI initiative from Intel. VDI, now jointly owned by Intel and Microsoft and renamed DCI, provides much more rapid access to individual pixels for high speed low overhead pixel update.

VGA does not usually include support for hardware decompression: An ideal place to put hardware support for video decompression is right where the data are to be rendered. At this time there are few VGA adapters that support major codec schemes in hardware. Cirrus is planning such support for Cinepak and MSKK has told us most major Japanese computer manufacturers are asking for direction in how to build in MPEG decompression.

VGA does not support hardware sprites: The performance of hardware sprites would enable two important groups of applications for Windows, high action level games and sprite-based video codecs. The former is well understood but the latter is a new technology being developed at Microsoft and several other companies.

They work this way - source content, either a movie or a video tape, has each frame in a set examined. Objects are identified and tracked from frame to frame. In *It's a Wonderful Life*, for example, we have worked with the scene where Jimmy Stewart and Donna Reed are walking together in front of the white picket fence. Jimmy, Donna, and a mail box they pass behind are each identified as objects. Everything left over is also treated as an object. Each object is traced through each frame in the series. A computer application then pulls each from the rest, examines their behavior, creates a single sprite, and, by performing mathematical operations on it, animates it through the same time period.

There are several interesting results:

The data required to recreate the screen is greatly reduced. All that is required are the initial sprites and the drawing and morphing instructions, plus an error signal to correct mistakes the morphing may make.

The rendering is independent of frames. This means motion is smoother and has no frame strobing introduced artifacts. In his memo on gsprites Nathan uses an example of having no wheels turning backward as a benefit of removing strobing artifacts.

By introducing drawing commands other than those needed to recreate the image we can make Jimmy and Donna do things they didn't actually do in the movie. This opens the door to interaction with movies and to new methods of movie making.

Video Conferencing Subsystems:

These are adapters that combine multimedia functions and add a few of their own. In addition to containing video capture and compression they also have bi-directional audio and communications interfaces to TAPI and networks. Several companies have shown us their video conferencing hardware and applications. We are now working on generalized support within Win32.

Multimedia Architectures:

Windows competes with MMPM from IBM on OS/2 and with Apple's QuickTime on System 7.

Microsoft Multimedia:

Under Windows 3.1 Microsoft multimedia is not as clean as is desirable. It has two components, the multimedia extensions documented in the Windows 3.1 SDK, and Video for Windows. These are often presented as separate architectures. The have complementary functions but because a single architecture is not presented there is room for some confusion by ISVs in how best to access certain services. Integration

of these into one 32-bit story will be accomplished in the multimedia content of Windows 95. Taking each of these in turn:

Multimedia Extensions:

Figure 2 - Multimedia Extensions

This information is taken from the *Microsoft Windows Multimedia Programmer's Workbook.*

Through **MMSystem** the extensions provide **multimedia applications** a set of **low level** APIs for access to waveform, MIDI, file I/O, and timer services. Upon these APIs is built a high level access, the **Media Control Interface**. MCI has adds the following:

- Both a string and a message-based interface.
- Access to multimedia services.
- The concept of an **MCI device driver**, a way to extend and customize multimedia services. An MCI driver can be for a physical device, such as a video tape recorder, or a logical grouping of services that have in common a common set of behaviors, such as digital video.

MMSystem also accesses physical devices through a device driver interface, principally used for accessing audio adapters.

Video for Windows 1.1:

Video for Windows architecture is based on the idea that there exists a **container** of "video" information. Video information is actually any set of multimedia data types and need not, in fact, include any actual video. A custom container can be defined by the developer to suit special purposes. The default format is Microsoft's **AVI** file format.

Within the container can be any number of multimedia **streams**. A developer can create his own multimedia stream type. VfW comes with several predefined, including frame-based video, waveform audio, and captioning.

An application has access to creating and manipulating these services through a set of APIs. These provide for:

- Manipulation of container files.
- Access to audio and video codecs and renderers.
- Communication with MCI devices.

VfW includes a set of predefined window classes and Visual Basic custom controls for creating capture and playback applications. How the parts fit is shown in Figure 3.

Figure 3 - Video for Windows Block Diagram

Briefly, the components are:

AVICAP.DLL - provides services for capturing multimedia data. The Visual Basic control CAPWNDX.VBX provides a subset of these features.

MSVIDEO.DLL - provides the video channel between AVICap and the video capture device driver, accesses video compression services, provides window for playing multimedia content, and high performance DIB drawing services.

MCIAVI.DRV - the MCI command interpreter for video.

AVIFILE.DLL - provides easy access to container files, including stream manipulation, locating key frames, queuing. Also provides base for custom file and stream formats.

Image Compression Manager - provides compression services by accessing installable compressors.

Audio Compression Manager - provides audio-specific compression services.

Apple QuickTime:

Apple positions all multimedia as QuickTime multimedia. They support one container, the QuickTime movie file. All multimedia data objects are QuickTime objects and are in a movie file. Like Video for Windows a movie file need not contain video data. Apple's presentation of their architecture is different from ours in that they diagram and document according to the data flow as an application developer needs to think about it. This is one of the fundamental changes we need made to our multimedia documentation for Windows 95. Figure 4 is a block diagram of QuickTime. It is taken from *Inside Macintosh: QuickTime 1.5 Developer Kit.*

Figure 4 - Apple QuickTime Block Diagram

The design is similar to that of Video for Windows. Like VfW QuickTime has a container format, the QuickTime file. It is more flexible that the VfW container in that it provides for referencing components outside the container itself. Unlike VfW QuickTime does not provide for substituting other file formats. The major components are:

Image Compression Manager - provides image compression services. Similar to VfW's ICM. Developers can create, install, and access their own compressors.

Movie Toolbox - APIs for storing, retrieving, and manipulating QuickTime movies. These are the APIs developers use to create editing applications. Tools are provided for the definition of and access to custom multimedia data types.

Video Media Handler - predefined data handler for the video multimedia data type.

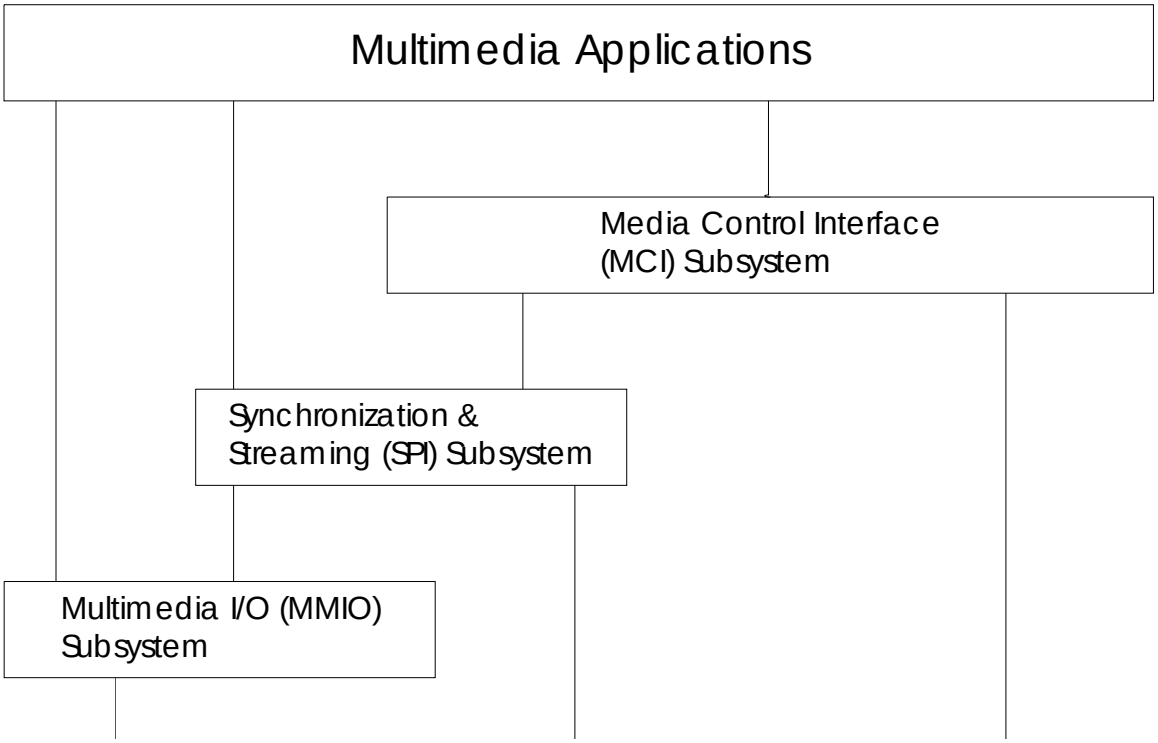
Sound Media Handler - predefined data handler for the audio data type.

Quickdraw - device independent drawing services used to render video data.

IBM MMPM (Multimedia Presentation Manager/2):

IBM's multimedia architecture is similar to that of Microsoft's (Figure5).

Figure 5 - MMPM Block Diagram



The surface similarity to the Microsoft multimedia extensions is obvious with the reference to MCI. There are, however, two differences. The first is the concentration of what Microsoft calls low level APIs in a separate subsystem, the MMIO. The second is the existence of a module specifically intended for streaming synchronization. By component:

Media Control Interface (MCI) - same as for the Microsoft multimedia extensions. IBM gives credit to Microsoft as joint developer of the MCI interface. The philosophy is a little different because of the existence of the SPI. Opening up a Wave device means that the wave data can be gotten. It doesn't mean the audio card is initialized. That happens by using the amp-mixer device, which is **connected** to the wave handler. Also, they have cue point support, which means that messages can be generated when specific events (such as time or position) take place.

Synchronization & Streaming (SPI) - breaks into two parts, APIs to provide an uninterrupted flow of data from storage, and synchronization.

Streaming - Microsoft has no services like these. Under Windows 3.1 the application must use file I/O services to request, or "pull" the next piece of data to be processed.

Synchronization - allows the grouping of data streams with one stream acting as the timing master with the others as slaves. Each stream handler has its own thread. The Sync Stream Manager also has its own thread. Streams will work with any data handler,

as long as it supports the streaming messages. If it supports the syncronization messages, it can be used as a master or a slave, depending on the set of messages it handles. Streams can come from any container, and they doesn't have to come from the same container.

The designated master stream is monitored by a Sync/Stream Manager thread that then tells slave streams when they are falling behind.

VfW 1.1 is restricted by the architecture of Windows 3.1 from providing synchronization pulses. Rather, stream renderers are provided the opportunity to query the time in another stream.

Multimedia I/O Programming Interface - a standardized method for applications to read and write different data types. MMIO is an extension to the OS/2 file services and is similar to the file services provided with the Microsoft multimedia extensions.

IBM has other multimedia offerings:

CD-ROM XA - described above in the video playback scenario describing multimedia data types and their requirements this is a specific CD-ROM format that has hardware-specific interleaving of data. IBM claims OS/2 to be specifically capable of supporting XA because of threads. Microsoft owns similar technology and we are considering adding it to the Windows 95 CDFS.

Networked video via DVI - hardware-assisted compression and decompression using an Intel technology. This is based on the Action Media II board set sold for some time by Intel. It is difficult to install and is waning in popularity.

Ultimedia - IBM's digital video system. Ultimedia is accessible through MCI, as is VfW, and provides an API set similar to that of Video for Windows. The container format is Microsoft's AVI file. Codecs supported are Indeo and IBM's own codec, Ultimotion. Ultimotion's performance is about equal to Cinepak.

Competitive position:

Success for Microsoft Multimedia means consumers and businesses buying PCs for multimedia choose Windows PCs over all others. They will do this when:

- The selection of titles, tools, special hardware, and applications is greatest.
- Windows PCs are the easiest to set up and use.

To achieve these we must have pervasive plug and play, complete technology, hardware vendors developing first for the Windows platform, and software vendors developing first for the Windows platform. In each area we currently trail Apple's QuickTime.

Hardware vendors may be categorized into manufacturers of PCs and of multimedia add-in components. Software vendors may be broken into those that provide titles (finished multimedia content products), tools for development of multimedia titles, and applications.

The following subsections contain a number of tables and statistics. Those without specific attribution come from *NewMedia 1994 Multimedia Tool Guide*, a special issue of *NewMedia* magazine, published by BPA International.

Hardware vendors:

Systems manufacturers:

This comparison is a little odd. Apple, of course, makes a variety of very fine multimedia-capable Macintosh computers. Their most recent offerings are the Centris 660AV and the Quadra 840AV. These machines incorporate the AT&T 3210 DSP, used for telecommunications, audio, and speech processing.

Apple has a standard, called ARTA for Apple Real Time Architecture. Through this the DSP is available to applications developers. Adobe is using it with their *Photoshop* application to accelerate functions and filters.

For PCs, however, the situation is more open. 26 manufacturers are shipping MPC2 compatible machines, including five that are portables. In addition, there are 31 MPC2 upgrade kits for Windows available from 17 manufacturers.

MPC2 Specification:

16-bit audio
double speed CD-ROM drive with 64kb on-board buffering
80486sx processor 25 mhz
8 mbytes RAM
160 mbyte hard disk
SVGA capable of 640 x 480 by 65,536 colors (16-bit true color)

Add-in components:

Video Capture Cards:

Video capture cards digitize video data. Source material is usually analog, either PAL or NTSC. Some cards also compress the data. Common algorithms are motion JPEG, DVI, and Indeo. Image sizes range from 160 by 120 pixels through broadcast quality of 704 by 485 (U.S. standard). Frame rates vary from zero through about 70, with commercial film at 24 and television at 30.

Video Capture Cards by Platform

Macintosh	15(1)
PC	29(2)
Other	1(3)
Total	45

Notes: (1) 15 adapters from 10 companies.

(2) 29 adapters from 20 companies. Of these adapters 7 ship bundled with Video for Windows. No other PC video software product is shipped with more than one or two adapters.

(3) 1 adapter, from Fast Forward Video, is claimed to work with any SCSI adapter, independent of PC or operating system.

Sound Cards:

16-bit cards are now the standard and the vast majority are for Windows, 42 different adapters from 33 different companies. Because the Macintosh ship with 8-bit audio built in the selection options are far fewer, about 4, one each from Digidesign, Macromedia, Spectral Innovations, and Media Vision.

Software vendors:

Tools:

Image Capture and Conversion:

These are utilities that produce the bitmaps multimedia presentation and animation graphics work with. These tools are used to capture screen activity or images, and to convert existing files to other formats. As acquirers of raw material they are not specific to a "brand" of multimedia. But they give some idea of the tool range between the Macintosh and Windows PC.

Number of Image Capture and Conversion Tools

Macintosh only	5
Windows PC only	6
Mac and Windows	2
Other*	3
Total:	16

Other = 1 each MS-DOS, OS/2, and NeXTEP.

2D Presentation and Animation:

These are the tools used to create presentations that contain movies, audio, and animation. Included are Asymetrix *Compel*, Gold Disk *Animation Works Interactive*, and Software Publishing *Harvard Graphics*.

Number of 2D Presentation and Animation Tools for Macintosh and Windows

Macintosh - no video	0
Windows - no video	4
Macintosh QuickTime	10
QT for Windows	0
Video for Windows	9
Both VfW and QT for Windows	2
Other (all MS-DOS)	6
Total:	31

Authoring:

These are tools used to create interactive multimedia applications. Included are AimTech *IconAuthor*, Asymetrix *Multimedia Toolbook*, and Macromedia *Director 3.1*.

Number of Authoring Applications for Macintosh and Windows

Macintosh - no video	1
Windows - no video	2
Macintosh QuickTime	9
QT for Windows	1
Video for Windows	13
Both VfW and QT for Windows	3
Other*	19
Total:	48

Other = 13 MS-DOS, 4 OS/2, 1 Unix, 2 Amiga. These total more than 19 because some were for more than one "other" operating system.

3D Modeling, Rendering, and Animation:

These are compute intensive tools that render animation in real time. Included are Adobe *Dimensions*, Autodesk *3D Studio*, and Softimage *Creative Environment*.

3D Animation Tools for Personal Computers and Workstations

Macintosh	19
Windows	9
Amiga	8
DOS	8
SGI	12
SUN	3
Other	6
Total	65

Sound Editors:

These are tools for cut, copy, paste, process, merging, and otherwise editing waveform audio. They usually present the audio data in points plotted over time. On the Macintosh the audio data format is AIFF or snd; on the PC either waveform or ADPCM.

Sound Editors by Operating System

Macintosh	5
Windows	9
Other*	3
Total	17

Other = 1 NeXTStep and 2 Amiga

Special Effects Video Software:

Special effects include morphing, mapping, and other elaborate manipulation of video transitions. Tools include *Digital Morph* from HSC Software, *MetaFlo* from The Valis Group, and *Morph* from Gryphon Software.

Special Effects Video Software

Macintosh	6
Windows	3
Mac and Windows	3
Other*	5
Total	17

Other = 2 for SGI and 3 for Amiga

Video Editing Tools:

These are the most visible of multimedia content manipulation tools, and include *After Effects* from CoSA, *MediaMerge* from ATI, and *Premiere* from Adobe. Windows trails both in the number of tools and their revision levels. Most video tools come out for the Macintosh before Windows.

Video Editing Tools

Macintosh	8
Windows VfW	1
Windows QT & VfW	2
Other (OS/2)	1
Total	13

Analog Video Editing Systems:

These are tools for editing video that is not destined for digital storage or distribution. Source material is analog of any source. Destination is almost always video tape - VHS, S-VHS, or Hi8. The computer is used to assemble source segments into a final tape, inserting transitions, etc. The content is

then copied from source machines to a transcribing tape machine. Control is accomplished through a device control language, something like Control-M. Video for Windows ships with a driver for the Sony family of video products. Content is located on the source material via SMPTE time code.

Analog Video Editing Systems by Platform

Macintosh	13
Windows	17
MS-DOS	8
Amiga	8
Other*	2
Total	48

Other = 1 each OS/2 and SGI.

Nonlinear Video Editing Systems:

These are editing systems that work with digitized source material, producing a final, digitized movie. This movie may be played on a digital platform or, with the proper hardware, converted to an analog signal and transcribed to tape. The more famous high-end video editing tools are in this group, including the *Media Suite Pro* from Avid Technology and *Montage MP3* from Montage Group.

Nonlinear Video Editing Systems by Platform

Macintosh	7
Windows	7
MS-DOS	3
Other (SGI)	1
Total	18

Multimedia Databases:

A multimedia database provides storage for multimedia data objects, everything from waveform audio and MIDI through movies. They provide access to information about each object (subject, size, format, play time, etc.) and a variety of ways to view it, including thumbnails, text lists, and previews.

Multimedia Databases by Operating System

Windows only	11
Macintosh only	9
Windows and Mac	3
Other*	3
Total	26

Other = 1 each for Sun, SGI, and OS/2

Titles:

Strategic Recommendations:

<to be worked on>