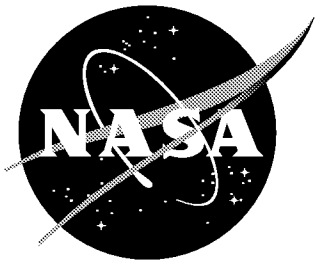
https://ntrs.nasa.gov/search.jsp?R=19990005547 2018-05-11T16:55:39+00:00Z

NASA/TM-1998-208748



GIF Animation of Mode Shapes and Other Data on the Internet

Richard S. Pappa
Langley Research Center, Hampton, Virginia

December 1998

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

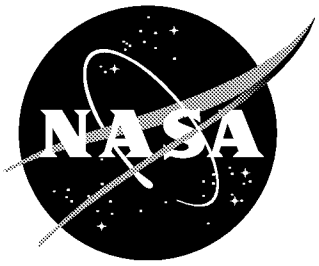
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-1998-208748



GIF Animation of Mode Shapes and Other Data on the Internet

Richard S. Pappa
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

December 1998

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

GIF Animation of Mode Shapes and Other Data on the Internet

Richard S. Pappa
Structural Dynamics Branch
NASA Langley Research Center
Hampton, Virginia 23681
E-mail: r.s.pappa@larc.nasa.gov

ABSTRACT

The World Wide Web abounds with animated cartoons and advertisements competing for our attention. Most of these figures are animated Graphics Interchange Format (GIF) files. These files contain a series of ordinary GIF images plus control information, and they provide an exceptionally simple, effective way to animate on the Internet. To date, however, this format has rarely been used for technical data, although there is no inherent reason not to do so. This paper describes a procedure for creating high-resolution animated GIFs of mode shapes and other types of structural dynamics data with readily available software. The paper shows three example applications using recent modal test data and video footage of a high-speed sled run. A fairly detailed summary of the GIF file format is provided in the appendix. All of the animations discussed in the paper are posted on the Internet at the following address: http://sdb-www.larc.nasa.gov/SDB/Research/data/pappa_data/imac17_figs.

INTRODUCTION

World Wide Web (WWW) designers use many types of graphics to enhance their pages. The two most common formats of WWW graphics files are the Graphics Interchange Format (GIF), developed by CompuServe in 1987, and the Joint Photographic Experts Group (JPEG) format, developed by the International Standards Organization (ISO) in 1991. GIF uses a maximum of 8 bits per pixel for data storage, allowing up to 256 colors, and is best suited for line drawings, simple cartoons, or photographs with overlaid text. JPEG uses 24 bits per pixel for data storage, allowing up to 16.8 million colors, and is better suited for scanned photographs, continuous-tone artwork, and similar material. References 1 through 3 elaborate on the principal Web graphics formats. The remainder of this paper will discuss only the GIF file format.

The World Wide Web today displays an abundance of cartoons, simple line drawings, and advertisement banners that move, wiggle, flash, or morph. Most of these figures are "animated GIFs" (Refs. 4,5), although there are other ways to create WWW animations including Java, Shockwave, and Common Gateway Interface (CGI) scripts. GIF format offers significant reductions in bandwidth and programming requirements compared to these other approaches. Animated GIF files contain a series of ordinary GIF images. Web browsers show each frame in rapid succession, providing a simple but very effective animation method. Control blocks in the file give various instructions to the browser, such as the duration of each frame, specified in hundredths of a second, and the number of times to cycle the animation. (Most animations play continuously.) Animation begins while the file downloads from the Web site. After the entire file is received, the animation continues to run without any additional data being sent over the Internet line.

The author recently investigated the possibility of creating animated GIF files containing test and/or analysis mode shapes. Using inexpensive software (mostly freeware and shareware), a simple method was developed to create detailed animated mode shapes (typically 640 x 480 pixels) with several colors and about 10 frames, yet they were usually under 100 kbytes in size. Such small files transfer over local networks in about 3 seconds. This new approach allowed modal test results to be shared with other people working on the same project and also with management in a simpler way than before. GIF format was then investigated for two other types of data, namely, for a series of data functions covering different frequency ranges and for video footage. These files were also easily created and were found to be an efficient, effective way to share this information on the Internet.

The paper first describes the software used to create the animated GIFs mentioned above. The process involves

three principal steps: 1) Generate a series of PostScript plots using the Device Independent Graphics Library (DIGLIB), developed at Lawrence Livermore National Laboratory, 2) Convert the PostScript plots to an unoptimized animated GIF file using the ImageMagick software, developed at E. I. du Pont de Nemours and Company, Inc., and 3) Optimize the animated GIF file using the GIFBatcher software, developed at Totally Hip Software, Inc. The paper then covers three example applications: an animated mode shape of a Beechcraft Starship fuselage; a series of plots showing identified natural frequencies versus assumed number of modes for the Starship fuselage; and video footage of a high-speed sled run at the Aircraft Landing Dynamics Facility (ALDF). The final section of the paper mentions some additional uses of animated GIFs in the structural dynamics field.

While it is not necessary to understand the internal structure of GIF files in order to create them (there is plentiful software available), knowing the main features and terminology of the format helps to produce optimum results. To aid the reader in developing other animated GIFs, the appendix of the paper contains a fairly detailed summary of the GIF file format.

SOFTWARE

Figure 1 shows a flowchart of the software steps. Each block in Fig. 1 is discussed separately in this section. **NOTE**: A large variety of software currently exists for performing each of the functions shown in the flowchart, and the particular software products mentioned in this paper are only representative choices. The use of particular brands of software in this work is not an official endorsement or promotion of any product by NASA or the United States Government.

The upper path in Fig. 1 with three steps creates animated mode shapes and other types of animated data plots using DIGLIB graphics output. The public-domain DIGLIB Fortran library has been the principal plotting software used by the author in his research for over a decade. It is a convenient way to create general-purpose graphics for either X-window display or as PostScript files. Example Nos. 1 and 2 in the paper follow this upper route.

The lower path in Fig. 1 with two steps is a special, additional path used to convert a QuickTime movie to animated GIF format. The movie had 261 frames of data at 30 frames per second, with a frame size of 160 x 120 pixels. Direct conversion of the entire movie created a GIF file that was substantially larger in size than the original QuickTime movie. Clearly, the QuickTime format excels for this particular purpose. However, it was found that by extracting every fifth frame of the movie, and then optimizing the file, the resulting GIF file was 28% smaller than the original movie, with only a slight decrease in animation quality. The GIF Animator software, developed at Ulead Systems, Inc., was used to create this animation. Example No. 3 in the paper follows this lower route.

DIGLIB Software

The Device Independent Graphics Library (DIGLIB) is a general-purpose Fortran graphics code developed in the 1980s by Hal Brand at the Lawrence Livermore National Laboratory. It is public-domain software, created under the sponsorship of the U.S. Government. The author uses DIGLIB to plot test data and modal identification results in the Fortran version of the Eigensystem Realization Algorithm (ERA) (Refs. 6,7). DIGLIB creates either X-window or PostScript output.

The ERA software includes a full-featured mode shape animation program named ERAG50, which uses the DIGLIB routines. As ERAG50 begins to execute, it pauses and displays the following question on the computer screen:

```
Graphics device number ?  (1)
  1 = x windows
  2 = b & w postscript tall
  3 = b & w postscript wide
 -2 = color postscript tall
 -3 = color postscript wide
```

Entering a "1" (the default) creates a color mode shape animation on the computer screen. Any of the viewing parameters can be easily changed during the animation by pressing appropriate keyboard buttons. After seeing the display, the user can make a hard copy by simply rerunning the program (using the up arrow to recall the previous command) and selecting a PostScript graphics device instead of X-windows. Entering a "-3," for example, generates a color printout in landscape format. By default, the PostScript file routes to a laser printer and then deletes. A simple software switch allows the user to save the output as a PostScript file rather than making a hard copy. Example No. 1 discussed in the paper uses a PostScript file created in this manner with ERAG50.

Another routine in the ERA software package, named ERAG15, creates a color-coded plot of ERA-identified natural frequencies versus the assumed number of modes. This program also uses DIGLIB and generates either X-window or PostScript output, as described above. A special feature in ERAG15 lets the user plot the results using a sliding frequency window in order to expand the display in narrow frequency bands. This allows the results to be viewed in greater detail. Example No. 2 discussed in the paper uses a PostScript file created in this manner with ERAG15.

ImageMagick Software

ImageMagick™ is the key piece of software used in this project. It is an extensive image processing package, developed by John Cristy at E. I. du Pont de Nemours and Company, Inc., for the display and interactive manipulation of images on the X-window system. Among many other features, it can transform images in almost any format to any other format. In particular, Version 4 of the ImageMagick software creates animated GIF files from sequences of images stored in over 70 different formats, including PostScript. (Ref. 8).

ImageMagick runs on most Unix systems, including Linux, and also on Windows NT, Windows 95, Macintosh, and VMS. Pre-compiled binaries are available for many popular operating systems, including SGI Irix, HP-UX, SunOS and Sun Solaris, IBM AIX, Windows 95, and Windows NT. The software is copyrighted, but it is available for free and can be redistributed without fee.

The author runs ImageMagick on an SGI workstation and uses it to create animated GIFs from ERA-generated PostScript files, mentioned above. Each PostScript file can contain any number of plots, and the entire file is converted to GIF format with a single ImageMagick instruction. To simplify things a little, an alias command named "ps2gif" performs the actual file conversion. For example, typing "ps2gif infile.ps outfile.gif" converts the entire contents of the PostScript file named infile.ps to an animated GIF file named outfile.gif.

The ps2gif command is equivalent to the following Unix shell string:

```
convert -delay 40 -loop 0 +dither
-dispose 1 -rotate -90
-density 56.47x58.18
-verbose $1 $2
```

where:*

- "convert" is the ImageMagick command.
- "-delay 40" specifies that each frame in the animation displays for 40/100 seconds. (This value is used for animated mode shapes with 10 frames.)
- "-loop 0" causes the animation to loop continuously.
- "+dither" deactivates dithering to preserve the original set of colors created in DIGLIB.
- "-dispose 1" sets the frame disposal method to "Do Not Dispose."
- "-rotate -90" rotates the plot 90 degrees counter-clockwise, required for DIGLIB "PostScript Wide" plots.
- "-density 56.47x58.18" creates GIF images that are 640 x 480 pixels in size. Changing these numbers will increase or decrease the image size proportionately.
- "-verbose" generates detailed information during the conversion process.
- "$1" represents the input PostScript file name.
- "$2" represents the output GIF file name.

As indicated in Fig. 1, animated GIF files created by ImageMagick are unoptimized, meaning that they do not use any of the special features of the GIF file format (see appendix) to reduce file size. GIFBatcher, discussed next, is one of several recently developed programs for optimizing animated GIFs using these special features. Optimized files download faster than unoptimized ones, with no change whatsoever in the resulting animation.

GIFBatcher Software

GIFBatcher™ is a Macintosh program developed by Totally Hip Software, Inc. (www.totallyhip.com) for optimizing animated GIFs. A distinguishing feature of the program is that it allows an entire folder of files to be optimized as a batch operation simply by dragging it onto the GIFBatcher icon. The program launches, minimizes the size of every file contained in the folder, and then turns itself off.

Reference 9 provides an excellent overview of the various aspects of GIF file optimization. Due to the rapidly increasing popularity of the WWW and animated GIFs, many new software products for GIF optimization have appeared recently on the market. A list of candidates is

* The appendix explains the meaning of the various GIF file attributes listed here.

available in Ref. 9. The author has tried several different optimization programs using the data discussed in this paper (most companies allow free trial periods) and has found only minor differences in the resulting size of the optimized files. There can be a noticeable increase in the required computation time, however, to obtain the highest optimization level. The products differ mostly in the number of features that are offered and in the user interfaces.

GIF Animator Software

GIF Animator™ is a PC Windows application developed by Ulead Systems, Inc. (www.ulead.com). It features a well-designed, professional appearance with easily understood menus and tool bars. As shown in Fig. 1, GIF Animator was used in this project to convert a QuickTime movie to GIF format. The software excelled for this purpose. It also allowed every fifth frame to be extracted easily, and a new GIF file was created using this subset of frames.

THREE EXAMPLE APPLICATIONS

The remainder of the paper covers three example applications. Example Nos. 1 and 2 use PostScript plots created with DIGLIB. Of course, there are many other ways to create a series of PostScript plots, and similar results should be expected using other sources. Example No. 3 shows how to make an animated GIF file from a QuickTime movie. The QuickTime movie was produced previously from a video recording. Only six frames of each animation are shown in the paper for brevity. All of the animations are available in their entirety on the Internet at the following address: http://sdb-www.larc.nasa.gov/SDB/Research/data/pappa_data/imac17_figs.

Table 1 summarizes the three examples, giving the principal file attributes and corresponding file sizes. These file sizes provide good estimates of the sizes to be expected for other similar applications. GIFBatcher was used for all file optimizations, and the optimization technique is discussed briefly in the first example application (mode shape animation). File size variation with changes in principal file parameters is also addressed in the first example. Although the third application (video animation) creates a much larger file than the other two, this animation is still quite efficient because all GIF animations begin to operate while the file downloads from the Web site, minimizing wait time.

Example No. 1: Animated Mode Shape

Figure 2 shows the test article used in this example. It is a Beechcraft Starship fuselage, manufactured about 10 years ago during the development phase of this vehicle. The structure serves as a testbed at the NASA Langley Research Center for interior noise reduction studies. The Starship is an all-composite business aircraft with aft-mounted pusher turboprops, variable-sweep canards, and large winglets that serve as vertical stabilizers. It is the first all-composite aircraft certified by the FAA. The photograph shows the structure mounted on rigid stands during a photogrammetry survey. It was then moved to the structural dynamics laboratory and remounted on air bags. A modal test was conducted in the free-free configuration.

Figure 3 shows the first vertical bending mode at 36.4 Hz, identified with ERA using frequency response functions. The ERAG50 program created ten color PostScript frames, and Fig. 3 shows six of these frames. The rotating clock at the bottom of each plot indicates the corresponding phase angle in the animation cycle. The modal test used 245 accelerometers, consisting of 64 radial measurements, 77 biaxial measurements, and 9 triaxial measurements. Each of the three types of measurements is plotted in a different color in Fig. 3 to simplify interpretation of the results (Blue: radial measurements; red: biaxial measurements; and green: triaxial measurements). Note that this low-order mode has a relatively complex shape with significant expansion and contraction of the cabin in conjunction with the overall vertical bending. The door (and side windows) were not installed during the test, so that the front of the aircraft also twists in the opposite direction from the back of the aircraft at the door location. Animation helps considerably to see such complicated motions more clearly.

The baseline mode shape animation in Fig. 3 is 640 x 480 pixels in size, with 10 frames and 5 colors. Tables 2 and 3 show the variation of GIF file size using other frame sizes, number of frames, and number of colors. In most cases, at least 10 frames per cycle are necessary to create a smooth animation, but at the larger frame sizes, reducing the number of frames down to five might be acceptable to keep file sizes below 150 kbytes while still giving a reasonable presentation. Variation of file size with frame size (i.e., total number of pixels) and number of frames in Table 2 is approximately linear, except at the smallest frame size. Table 3 gives additional results using various numbers of colors. A 20% file size reduction

occurs by decreasing the number of colors from 5 to 2, albeit at a considerable loss in visual content. In every case shown in Tables 2 and 3, optimized files are about 15%-25% smaller than corresponding unoptimized ones. Quantitative understanding of these file size variations requires a good working knowledge of the GIF file format, summarized in the appendix.

When viewed on a Web browser, optimized GIF animations created by GIFBatcher look exactly the same as the corresponding unoptimized ones. There is no change whatsoever in the visual appearance, although the optimized file can be considerably smaller than the unoptimized one. To show how this size reduction occurs, Table 4 compares the unoptimized and optimized files for Fig. 3 on a frame-by-frame basis. Two changes have occurred in the optimized file: 1) Frames 2 through 10 have smaller frame sizes than the original 640 x 480 pixels, with accompanying changes in the frame origin, and 2) Four of the frames in the optimized file have added a transparent color (see appendix for explanation of transparency and other GIF parameters). The optimized file is taking advantage of these special features of the GIF file format.

Example No. 2: Animated Sliding-Frequency Window

Animated GIFs are also useful for creating slide shows of technical data on the Internet, as demonstrated in this example. The overall data for the slide show appears in Fig. 4 in its original form. This complicated plot shows ERA-identified natural frequencies of the Starship fuselage from 25 to 115 Hz versus the assumed number of modes ranging from 1 to 100. The height of each vertical dash in the plot is proportional to the Consistent-Mode Indicator (CMI), the principal accuracy indicator of ERA (Ref. 7). Modes with CMI values greater than approximately 80% are identified with high confidence. The CMI information is also represented using three different colors for various ranges as noted at the top of the figure (Blue: CMI at least 90%; green: CMI between 80% and 90%; black: CMI less than 80%). Color coding makes interpretation of the results much easier. The triangles at the top of the plot indicate the final set of identified natural frequencies obtained using an automatic mode condensation algorithm (Ref. 10).

To see the results in Fig. 4 more clearly, Fig. 5 shows the same data replotted in six 15-Hz-wide frequency ranges. It is now obvious, for example, that there are two modes at 31 Hz, and that the results near 57 Hz break up unacceptably in the range of 90 to 100 assumed modes. Neither of these details was clear in Fig. 4. Figure 5 also shows the degree of frequency fluctuation for each mode versus the assumed number of modes much better than Fig. 4.

A slide-show animated GIF with a frame size of 640 x 480 pixels is created from the six plots in Fig. 5. The "ps2gif" command, described earlier, converts a PostScript file generated by ERAG15 to GIF format, with the exception that the interframe delay is changed from 40 to 500. This modification causes each frame to display for 5 seconds, rather than for 40/100 seconds. If five seconds per frame are insufficient to see the data adequately during the slide show, the animation can be paused at any time for closer viewing by hitting the stop button on the Web browser. (Unfortunately, however, the slide show cannot be resumed if it is stopped, but it must be restarted from the beginning using the reload button. Perhaps future versions of Web browsers will allow paused animations to continue.)

Table 1 shows the files sizes for this sliding-frequency application. The optimized file in this case is only 24 kbytes in size, about 3-1/2 times smaller than the animated mode shape. The smaller file size of Example 2 compared to Example 1 results primarily from two factors: there are about half as many frames in the second animation, and there is more open white space. The GIF format is particularly efficient for storing images with large areas of the same color.

Example No. 3: Animated Video Recording

The third example illustrates the creation of an animated GIF file from video footage. Figure 6 shows six of the frames contained in the video. It displays a high-speed sled run at the Aircraft Landing Dynamics Facility (ALDF) located at NASA Langley Research Center. ALDF is a test track used primarily for landing gear and tire research. Pressurized water applies a force exceeding 2,000,000 lbs to a 54-ton test carriage, creating a peak acceleration on the order of 20 g's. The carriage accelerates to 200 knots within 400 ft., coasts through an 1800-ft test section, and is then arrested by five cables. High-speed photography and other instrumentation record the performance of landing gear and tires during the run.

The lower path in Fig. 1 shows the process used to create an animated GIF of this sled run from a QuickTime movie. The QuickTime movie with 261 frames was

produced previously from a video recording. (Various desktop computers and workstations with video input capability can create QuickTime movies.) The GIF Animator software reads digital movies and converts them automatically to animated GIF format. That is all that was necessary to create an unoptimized GIF file of the entire film. However, the resulting GIF file was found to be several times larger in size than the original file.

As a simple experiment to see if a good GIF animation could be created nevertheless at a lower frame rate, with a correspondingly smaller file size, every fifth frame of the movie was extracted in GIF Animator. The interframe delay was changed from 1/30 seconds to 17/100 seconds to approximate the original time scale. This new GIF file with 53 frames was then optimized using GIFBatcher. The resulting GIF file is now 28% smaller than the original movie--619 kbytes vs 856 kbytes (see Table 1).

Although this new file is considerably smaller than the original, the QuickTime version uses 30 frames per second while the GIF animation uses only 6 frames per second, so the original movie plays smoother. The QuickTime version would be the preferable choice in most cases. However, the animated GIF version also provides a very good representation of the original video and downloads quicker. Furthermore, as mentioned previously, animated GIFs begin to play while the file downloads from the Web site, minimizing the wait time. Both the QuickTime and animated GIF versions of the movie are available for viewing on the Internet at the site mentioned at the beginning of this section.

OTHER APPLICATIONS

Three additional, potential uses of animated GIFs for sharing structural dynamics data on the Internet are:

1. Comparison of test and analysis mode shapes
   Pairs of test and analysis mode shapes can be displayed side-by-side in an animated GIF as follows. Use the ERAG50 program to create separate PostScript files of corresponding test and analysis modes. Interleave the two files into a single file named infile.ps, then type the command, "ps2montage infile.ps outfile.gif." The "ps2montage" command is equivalent to the following Unix shell string:

```
montage +frame +shadow +label
+dither -bordercolor red
-borderwidth 1 -density 47.06x47.06
-tile 2x1 -geometry 400x518
-verbose $1 temp.gif;
convert -delay 40 -loop 0
-verbose temp.gif $2
```

   where:

   - "montage" and "convert" are ImageMagick commands.
   - "$1" represents the input PostScript file name.
   - "$2" represents the output GIF file name.

2. Transient or forced response animation
   Using an approach similar to that discussed in Example No. 1, the motion of a structure resulting from arbitrary excitation forces versus time can be displayed with an animated GIF file. The data can be either test or analysis results.

3. Frequency response or mode indicator functions
   These commonly created test functions can be displayed using the approach discussed in Example No. 2. Animated GIF slide shows can contain either sliding-frequency displays or a series of different functions covering the same frequency interval. The slide show can step quickly from frame to frame for simple plots or step slowly for complex plots.

CONCLUSIONS

This paper showed how to create high-resolution GIF animations of technical data using inexpensive software. Three example applications were covered. The first example showed how to create animated mode shapes using DIGLIB Fortran graphics and ImageMagick. This approach provides a very efficient, new way to share modal analysis results. The second example showed how to create slide shows. A complex plot covering a wide frequency range was split into six intervals that were displayed in slide-show fashion. The animation runs continuously, or it can be stopped at any time for closer examination. The third example showed how to create an animated GIF from a QuickTime movie. Although the QuickTime format is more efficient than GIF for video data, a good animation was produced using every fifth frame of the original movie. In all three examples, the GIFBatcher program performed interframe optimization. The paper closed by mentioning some additional uses of animated GIFs. In the months ahead, other good uses will likely arise as well.

APPENDIX

GRAPHICS INTERCHANGE FORMAT

The appendix summarizes the main features of animated GIF files. For the interested reader, the complete specifications are posted on the Internet (Ref. 11). CompuServe created the initial specifications in 1987 (known as GIF87a) and then expanded them in 1989 (known as GIF89a). **NOTE**: All GIF images are compressed using the Lempel-Ziv-Welch (LZW) technique, which is patented by the Unisys Corporation (Ref. 12). Ramifications of this patent are discussed at the end of the appendix.

Each image in an animated GIF is a raster array of color indices with H pixels in the horizontal direction and V pixels in the vertical direction. There are no restrictions on the sizes of H or V. Every image is compressed row-by-row using the LZW method. There is no option to skip compression, nor are other compression methods available. In addition to the image arrays, each file also contains various control parameters that apply to either the entire file or to individual frames, as follows:

Global Parameters

The parameters in this category apply to the entire GIF file.

1. Global Color Map - GIF uses variable color depth, which allows images with fewer colors to use fewer bits for data storage. Anywhere from 1 to 8 bits are available, corresponding to a maximum number of colors, C, ranging from 2 to 256. If C is not a power of two, the next higher number of bits is used; e.g., images with 5 colors require 3 bits per pixel for storage. Each color in the image is assigned a color index ranging from 0 to C-1. The color map associates each color index with its corresponding red, green, and blue color intensities. The global color map applies to the entire file unless superseded by a local color map for an individual frame.

2. Number of Loops - The number of loops is the number of times that a GIF animation repeats itself, ranging from 1 to infinity, where zero represents infinity.

3. Background Color - GIF format includes designation of a background color. The background color appears outside the boundaries of reduced-size images, and it will also show through transparent pixels (discussed below). However, when viewed on Web browsers, GIF images usually inherit the background properties of the corresponding Web page.

Frame Parameters

The parameters in this category apply to each frame individually.

1. Local Color Map (optional) - Each frame can have its own color map, which supersedes the global color map. Local color maps are usually avoided, however, because flickering can occur.

2. Frame Size - Frame size is the number of pixels in the image in the horizontal and vertical directions. The simplest way to create animated GIF files uses the same frame size for every image. However, significant reductions in file size are possible by cropping each picture to contain only those pixels that change from a previous image. This is one aspect of interframe optimization. Note that the physical size of a GIF image on a computer monitor depends on the monitor resolution (pixels per inch).

3. Frame Origin - Frame origin specifies the coordinates of the upper-left corner of the image, where (0,0) indicates an undisplaced image. Reduced-size images in optimized GIF files translate by varying amounts so that each image covers only those pixels that have changed from a previous image.

4. Interframe Delay - Each frame of an animated GIF displays for M hundredths of a second, where integer M is the interframe delay. On slower computers (e.g., less than 200 MHz), the actual clock time of each animation cycle may be somewhat greater than the sum of the interframe delays specified in the file because of insufficient processing speed. On faster computers (above 200 MHz), the cycle time will be the sum of the interframe delays.

5. Transparency - GIF format allows one color index in the image to be designated as transparent, allowing the background color or a previous image to show through these pixels. Optimization software can change certain pixels in an image to transparent to reduce file size. (In the LZW data compression method, a horizontal string of transparent pixels

requires fewer bytes of file storage than a horizontal string of varying colored pixels.)

6. Disposal Method - The disposal method of a frame tells the Web browser what to do with the frame after being displayed. This feature is used in concert with frame size, frame origin, and transparency in GIF optimization software to reduce file size. There are four possible disposal methods, as follows:

   Do Not Dispose - causes the pixels that are not covered up by the next frame to continue to display. This option is used in optimized GIFs having reduced-size frames, where each image covers only those pixels that have changed from a previous image. It requires the least work from the Web browser, permitting higher-speed animations on slower machines.

   Restore to Previous - restores to a previous, undisposed image (not necessarily the previous one). This option can be used to overlay a series of varying subimages onto a repeating, background image. The background image shows through any transparent pixels in the overlaid images.

   Restore to Background - allows the background color or background image, rather than a previous frame, to show through the transparent pixels. This option can be used to allow a series of objects with transparent backdrops to float around on the background.

   Unspecified - lets the Web browser decide what to do with the frame after being displayed. This option is usually equivalent to "Do Not Dispose."

   Note: At the beginning of every animation cycle, the first image fully replaces whatever is displayed during the last frame of the animation; i.e., the display refreshes completely at the beginning of each animation cycle.

LZW Compression & Patent Controversy

All images stored in GIF files are compressed and decompressed using the Lempel-Ziv-Welch (LZW) algorithm developed by Lempel and Ziv in the late 1970s and refined by Welch in the early 1980s (Refs. 13-15). The LZW method achieves compression by replacing commonly used patterns of color indices appearing in the image with special codes. The image is compressed row-by-row, so that horizontal patterns compress well, whereas vertical patterns do not. The compression codes are stored in a "dictionary," which grows as the image is encoded. A significant feature of the algorithm is that it is unnecessary to save the dictionary in the file. The same dictionary is rebuilt automatically when the image is decoded.

A significant controversy arose in 1995 concerning the GIF file format. When CompuServe developed the specifications in 1987 and 1989, they did not realize that the LZW method was patented by the Unisys Corporation (Ref. 12). The GIF specifications granted royalty-free use of the format, and CompuServe encouraged its widest possible dissemination. When the WWW began to grow in the early 1990s, GIF became the principal graphics format. For many years, Unisys did not realize that the GIF file format included LZW compression.

In 1995, Unisys began enforcing the patent, which generated considerable uproar in the Internet community. The current status of the situation is that developers of commercial software that uses LZW compression or decompression must pay a license fee. End users, non-profit organizations, and online services are not required to pay the fee. A direct result of this controversy is that the Web community is developing a replacement graphics format known as PNG (Ref. 3). The PNG format is now supported on most major Web browsers. However, at this writing, the PNG format does not permit animation, so that animated GIFs will continue to appear abundantly on the WWW into the foreseeable future.

REFERENCES

1. Kentie, P., Web Graphics Tools and Techniques, Peachpit Press, Berkeley, CA, 1997.

2. Fernandez, J. N., GIFs, JPEGs, and BMPs: Handling Internet Graphics, MIS Press, New York, NY, 1997.

3. Webster, T., Web Designer's Guide to Graphics: PNG, GIF, and JPEG, Hayden Books, Indianapolis, IN, 1997.

4. Frazier, R., "GIF Animation on the WWW," Website: http://members.aol.com/royalef/gifanim.htm.

5. Koman, R., GIF Animation Studio: Animating Your Web Site, Songline Studios, Inc. and O'Reilly and Associates, Inc., Sebastopol, CA, 1996.

6. Pappa, R. S., "ERA Bibliography," Website: http://sdb-www.larc.nasa.gov/SDB/Research/data/ERA_biblio.html, September 1998.

7. Pappa, R. S., "Eigensystem Realization Algorithm User's Guide for VAX/VMS Computers," NASA TM-109066, May 1994.

8. Cristy, J., "ImageMagick - X11 Image Processing and Display Package," Website: http://www.wizards.dupont.com/cristy/ImageMagick.html.

9. King, A., "Optimizing Animated GIFs," Website: http://www.webreference.com/dev/gifanim, July 1998.

10. Pappa, R. S., Woodard, S. E., and Juang, J.-N., "The Development of Autonomous Structural Modal Identification," *Sound and Vibration*, August 1997, pp. 18-23.

11. CompuServe, Inc., "GIF Specifications," Websites: http://members.aol.com/royalef/gif87a.txt (GIF87a), May 1987, and http://members.aol.com/royalef/gif89a.txt (GIF89a), July 1989.

12. Welch, T. A., "High Speed Data Compression and Decompression Apparatus and Method," U. S. Patent No. 4,558,302, Unisys Corporation, December 1983.

13. Ziv, J. and Lempel, A., "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, Vol. IT-23, No. 3, May 1977, pp. 337-343.

14. Welch, T. A., "A Technique for High-Performance Data Compression," *Computer*, Vol. 17, No. 6, June 1984, pp. 8-19.

15. Blackstock, S., "LZW and GIF Explained," Website: http://www.dc.ee/Files/Programm.Packing, Download file lzwexp.arj, September 1992.

Table 1 - File Attributes and Sizes of Three Example GIF Animations

	File Attributes			File Size, kbytes		
	Frame Size, H x V pixels	No. of Frames	No. of Colors	PostScript (P) or QuickTime (Q) file	Unoptimized GIF file (* = reduced to 53 frames)	Optimized GIF file (* = reduced to 53 frames)
Mode Shape Animation (Fig. 3)	640 x 480	10	5	536 (P)	106	85
Sliding-Frequency Animation (Fig. 5)	640 x 480	6	4	165 (P)	33	24
Video Animation (Fig. 6)	160 x 120	261	256	856 (Q)	829 (*)	619 (*)

Table 2 - File Size of Animated Mode Shape (Fig. 3) vs Frame Size and Number of Frames

Frame Size, H x V pixels	Total pixels, thousands	Unoptimized GIF file size, kbytes		Optimized GIF file size, kbytes	
		5 Frames	10 Frames	5 Frames	10 Frames
320 x 240	77	21.1	41.6	17.3	31.6
640 x 480	307	53.3	106.0	45.7	84.8
800 x 600	480	70.7	140.7	61.4	114.3
1024 x 768	786	106.5	212.5	92.5	172.6
1280 x 1024	1311	155.5	310.5	135.2	252.3

Table 3 - File Size of Animated Mode Shape (Fig. 3) vs Number of Colors

No. of Colors	Unoptimized GIF file size, kbytes	Optimized GIF file size, kbytes
2	82.6	67.7
4	102.0	83.8
5	106.0	84.8

Table 4 - Comparison of Unoptimized and Optimized GIF Files of Animated Mode Shape (Fig. 3)

Frame No.	Unoptimized GIF file				Optimized GIF file			
	Frame size	Frame origin	Transparent color?	File size, kbytes	Frame size	Frame origin	Transparent color?	File size, kbytes
1	640 x 480	0, 0	No	11.0	640 x 480	0,0	No	11.0
2	640 x 480	0, 0	No	10.2	589 x 274	20,153	No	8.0
3	640 x 480	0, 0	No	10.2	592 x 313	21,121	No	8.2
4	640 x 480	0, 0	No	10.4	594 x 349	24,96	Yes	8.6
5	640 x 480	0, 0	No	10.5	593 x 360	27,90	Yes	7.6
6	640 x 480	0, 0	No	10.4	591 x 360	29,90	Yes	8.0
7	640 x 480	0, 0	No	10.0	592 x 340	28,105	No	8.4
8	640 x 480	0, 0	No	10.5	592 x 299	25,135	No	8.3
9	640 x 480	0, 0	No	11.0	591 x 274	22,153	No	8.8
10	640 x 480	0, 0	No	11.2	589 x 278	20,150	Yes	7.9
TOTAL:				105.4				84.8

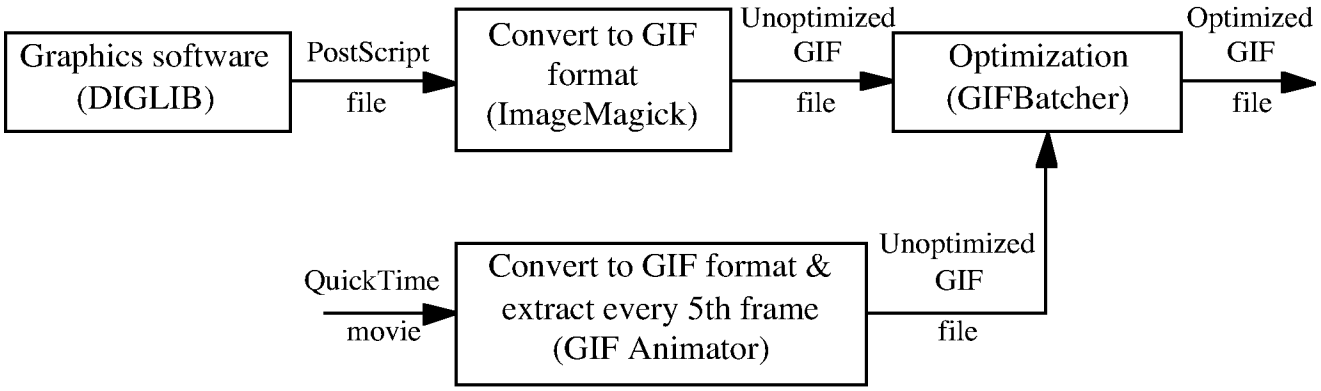


Fig. 1 - Software Flowchart (Examples 1 and 2 use upper path; Example 3 uses lower path)

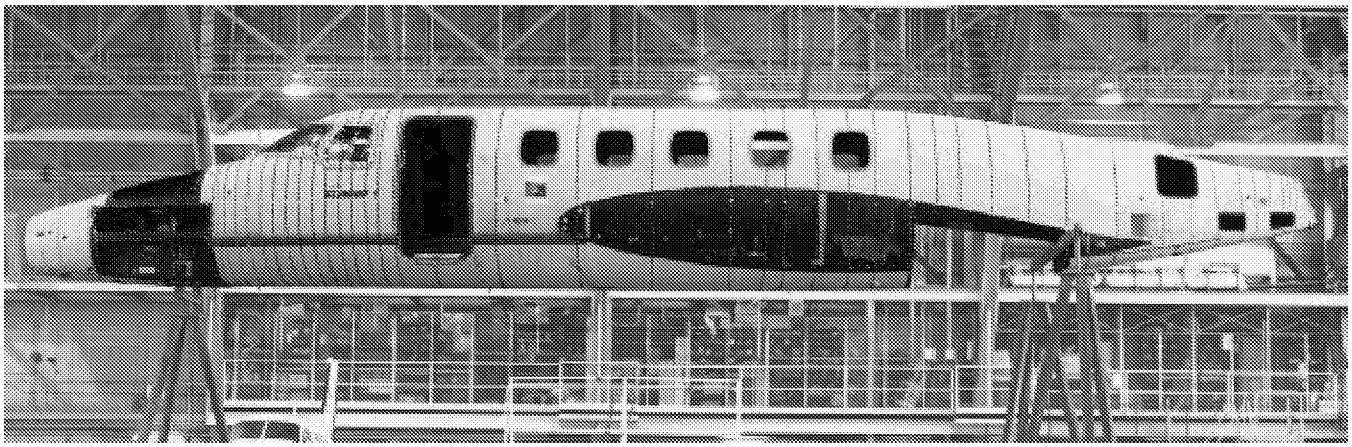
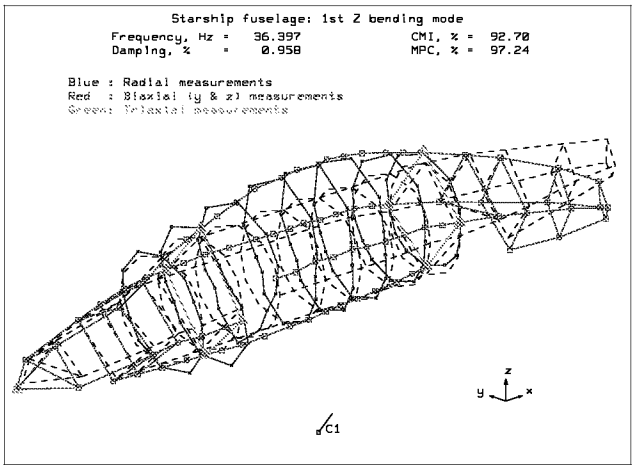
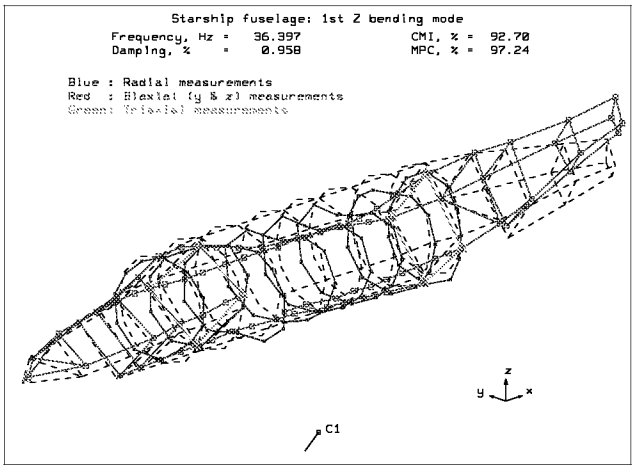


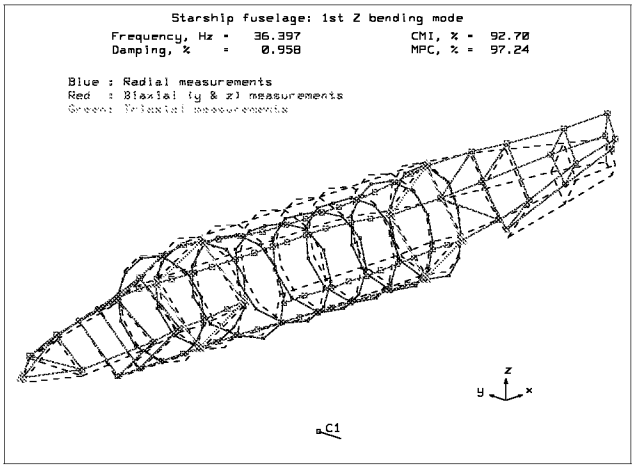
Fig. 2 - Starship Fuselage Test Article



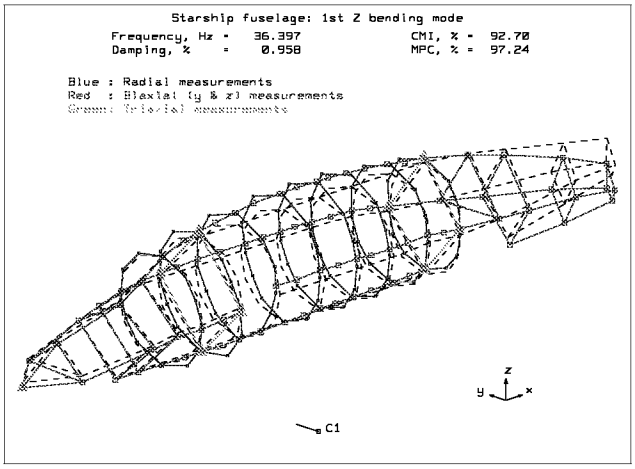
(a) Frame 1



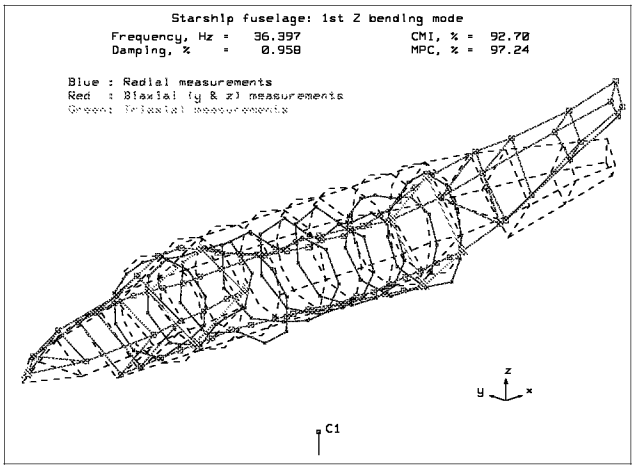
(d) Frame 6



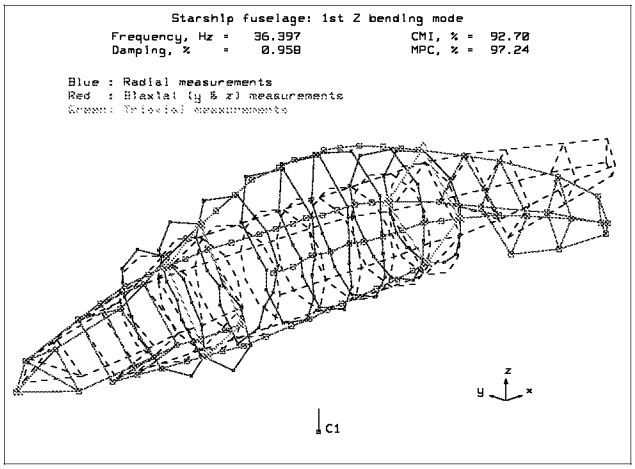
(b) Frame 3



(e) Frame 8



(c) Frame 5



(f) Frame 10

Fig. 3 - Six Frames of an Animated Mode Shape (10 Frames in Total)

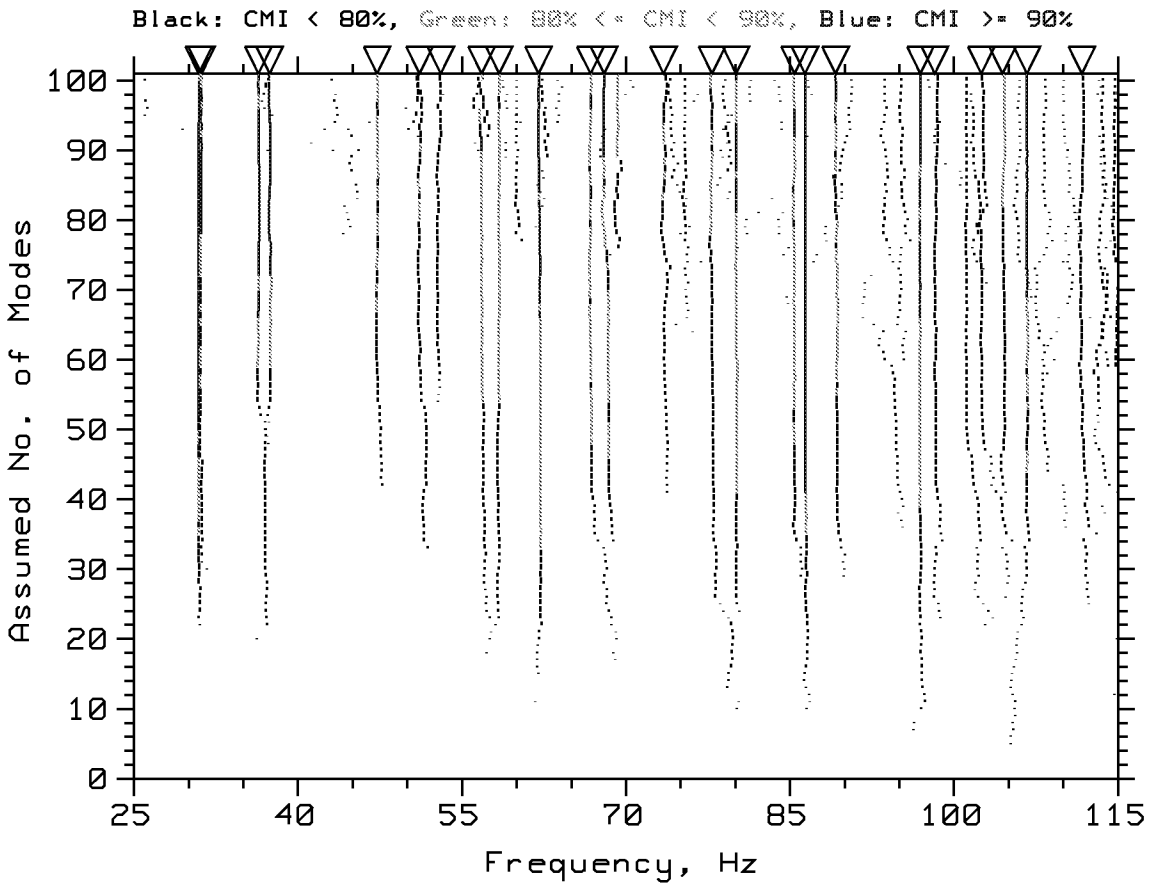
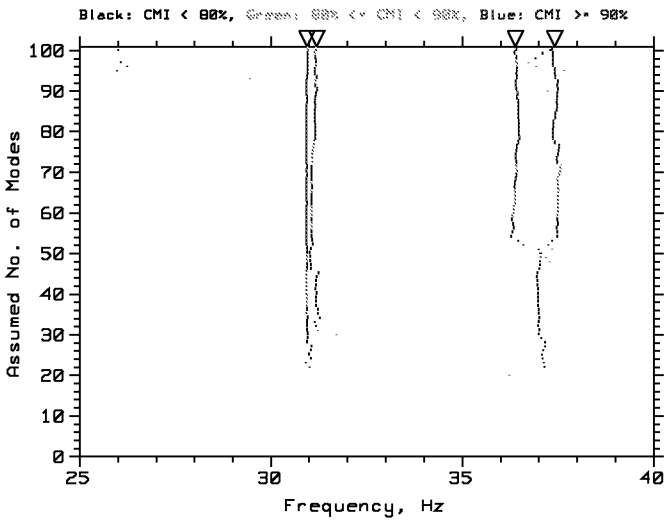
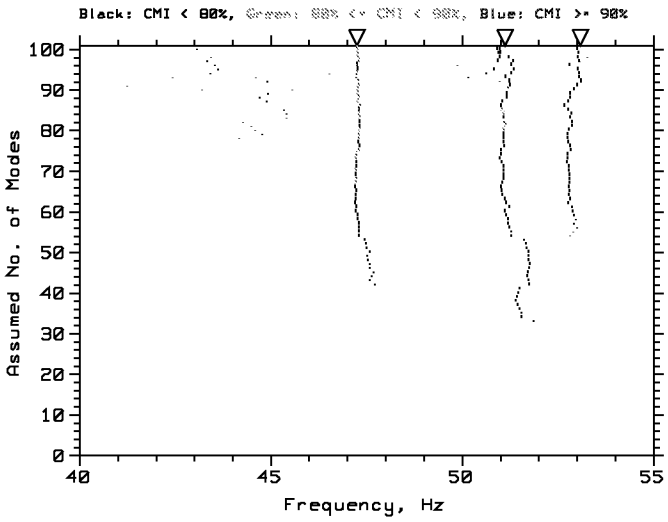


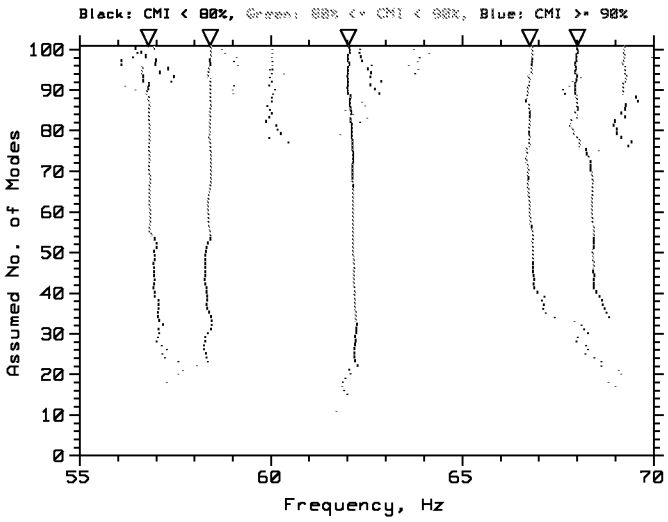
Fig. 4 - Identified Natural Frequencies vs Assumed No. of Modes
(Difficult to See Details in this Complex Plot)



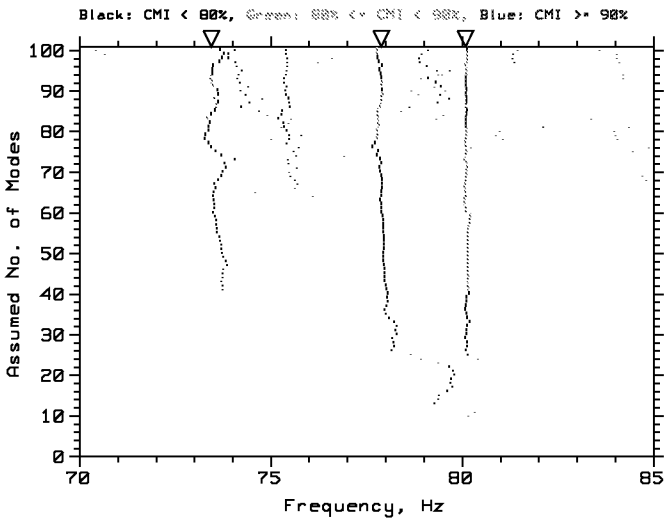
(a) Frame 1



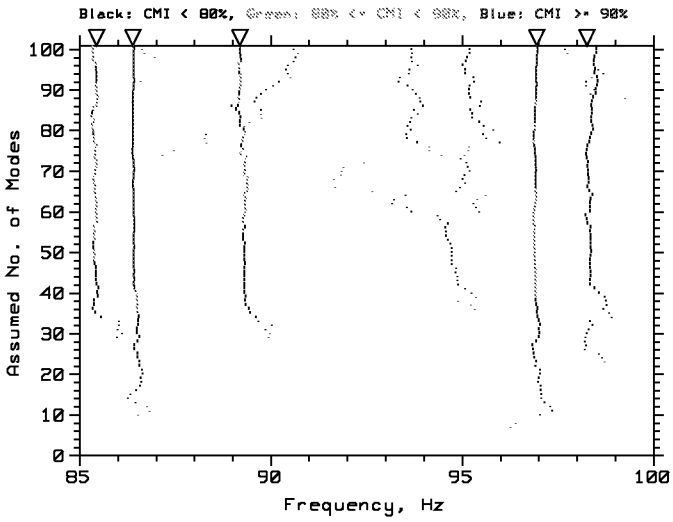
(b) Frame 2



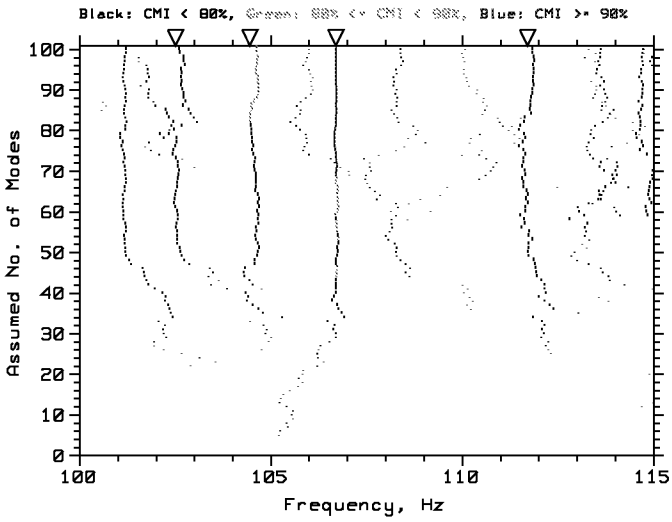
(c) Frame 3



(d) Frame 4

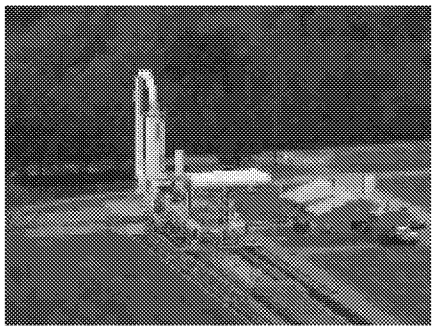


(e) Frame 5

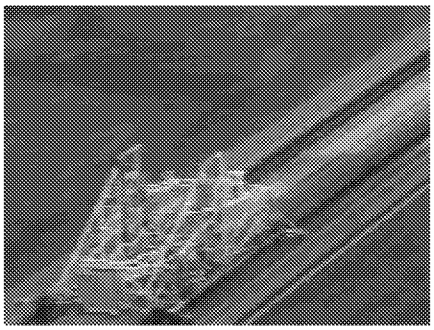


(f) Frame 6

Fig. 5 - Six Frames of an Animated Sliding-Frequency Display of Fig. 4 (6 Frames in Total)



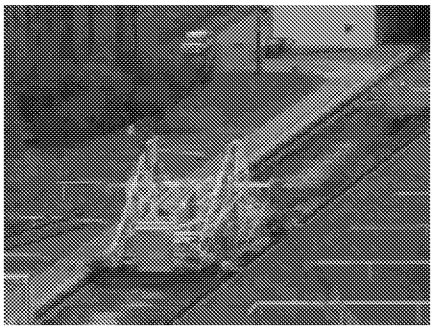
(a) Frame 1



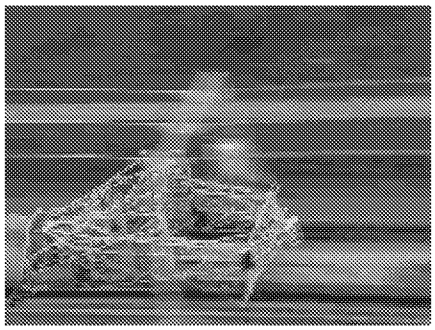
(d) Frame 31



(b) Frame 11



(e) Frame 41



(c) Frame 21



(f) Frame 51

Fig. 6 - Six Frames of an Animated Video Recording (53 Frames in Total)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Technical Memorandum

4. TITLE AND SUBTITLE

GIF Animation of Mode Shapes and Other Data on the Internet

5. FUNDING NUMBERS

WU 632-10-14-04

6. AUTHOR(S)

Richard S. Pappa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-17789

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA/TM-1998-208748

11. SUPPLEMENTARY NOTES

Pappa: Langley Research Center, Hampton, VA. Presented at the 17th International Modal Analysis Conference, Kissimmee, Florida, February 8-11, 1999.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified-Unlimited
Subject Category 39 Distribution: Standard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The World Wide Web abounds with animated cartoons and advertisements competing for our attention. Most of these figures are animated Graphics Interchange Format (GIF) files. These files contain a series of ordinary GIF images plus control information, and they provide an exceptionally simple, effective way to animate on the Internet. To date, however, this format has rarely been used for technical data, although there is no inherent reason not to do so. This paper describes a procedure for creating high-resolution animated GIFs of mode shapes and other types of structural dynamics data with readily available software. The paper shows three example applications using recent modal test data and video footage of a high-speed sled run. A fairly detailed summary of the GIF file format is provided in the appendix. All of the animations discussed in the paper are posted on the Internet at the following address: http://sdb-www.larc.nasa.gov/SDB/Research/data/pappa_data/imac17_figs.

14. SUBJECT TERMS

Animated Mode Shapes, GIF Animation, Animated GIF, Graphics Interchange Format, Structural Dynamics, Modal Testing, Modal Parameters, World Wide Web, Internet

15. NUMBER OF PAGES

20

16. PRICE CODE

A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL