

A Medical Workstation for the Display and Analysis of Multi-Frame Digital Cineradiographic Data

K. Mueller², E.L. LaPresto¹, K.A. Powell¹, J.L. Hirsch³, J.F. Cornhill^{1,2}

¹Department of Biomedical Engineering, The Cleveland Clinic Foundation, Cleveland, OH

²Biomedical Engineering Center, The Ohio State University, Columbus, OH

³Shiley Heart Valve Research Center, Irvine, CA

Abstract-- A medical workstation has been developed for the efficient display and analysis of large sets of digital cineradiographic images. Various features aid the clinician in quickly identifying and extracting the image information relevant for diagnosis: animated viewing of image frames, a digital magnifying glass for local image enlargement and enhancement, a special review queue for critical image frames, and task-oriented image processing. Double frame buffering and direct memory addressing ensure fast, artifact-free image display and transfer. Much freedom is provided for adapting the system to one's own preferences. A statistical analysis of extensive tests conducted by eight clinical expert reviewers is given.

Keywords: Graphical User Interface design, medical workstation, digital cineradiography, digital cineangiography, digital image management, PACS, computer assisted radiology, computer aided diagnosis, human factors.

1. INTRODUCTION

Angiographic time-motion studies, such as the visualization of the beating heart or the circulatory system, tend to generate vast amounts of image data. An even larger load of image data is produced if separate sets of images are acquired from a number of different viewing angles or from different modalities to ensure that all potential pathologies are being captured. As a consequence, the magnitude of images that have to be reviewed by the clinician can be overwhelming, a circumstance that may compromise his or her attentiveness in the diagnostic process. Thus, due to human fatigue, subtle yet pertinent image information relating to pathologic features may be overlooked which could lead to a wrong diagnosis. To prevent this potential risk, the image data review process must be made as efficient and streamlined as possible, enabling the clinician to quickly grasp, enhance, and extract the relevant portion of the image data and discard unimportant image information. By employing such a procedure, highly trained clinical personnel may be better utilized, ultimately leading to a reduction in health care cost.

Traditionally, angiographic time-motion studies have been captured on rolls of cine film subsequently examined on specially designed film viewing machines (e.g., the XR-350 cine viewer,

Contact: Klaus Mueller, The Ohio State University, Biomedical Engineering Center,
270 Bevis Hall, 1080 Carmack Road, Columbus, OH 43210
Tel : (614) 292-1555, Fax: (614) 292-7301, E-mail: klaus@chaos.bme.ohio-state.edu

Vanguard Instrument Corporation, Long Island, NY). In a typical review session, the radiologist, cardiologist or cardiac surgeon places himself in front of the machine's TV-sized back projection screen, controlling the magnified film's speed, direction and brightness via a set of sliders or knobs. To change the image magnification, a set of optical lenses can be inserted between the illuminated film and the projection screen by flipping a selector switch. Three basic limitations of this analog system are immediately apparent: 1) The images can only be viewed sequentially; if the reviewer desires to compare two critical frames located far apart on the film roll, he is required to scan back and forth across many uninteresting images. Clearly, this tedious navigation task prolongs the reviewing process significantly. 2) Different views are either on separate film rolls or placed back to back on a single roll. This makes it impossible for the reviewer to compare side by side frames of different views taken at equivalent time instances within the heart cycle. 3) Image enhancement other than brightness control and magnification can only be achieved by restrictive and often expensive optical arrangements.

Presently, modern imaging equipment allows the output of image data in native, digital form. This opens the gate to a wealth of digital tools and techniques much more effective than the simple magnification and illumination manipulations of the analog cine viewers. Examples of such digital tools include operators for image enhancement, filtering, and automated segmentation, which all can be used to aid the clinician in the diagnostic process. A digital environment generally also facilitates image data management, communication, and navigation, both locally during a review session and globally across department and hospital boundaries. Significant recent advances in the conceptual development of Picture Archiving and Communication Systems (PACS) [1]-[4], the introduction of the DICOM standard [5][6] designed to facilitate image data interchange among clinical institutions, and the growing number of commercial PACS (e.g., Shared Vision from ImageLabs, Burlington, MA; VRS and Allegro by ISG, Toronto, Canada; and Bscan from GLP Rochester, NY), and public domain PACS (such as OSIRIS [7][8]) are clear evidence that digital image review technologies are in the process of replacing traditional film-based procedures. However, when designing clinical software for image management, display, and enhancement, one has to take into account that the clinician may be inexperienced in the use of modern computer workstations. To avoid frustrations that could ultimately lead to the rejection of the software, the digital system must be designed in such a way that it maintains a functionality similar to the traditional analog system but is augmented by the new digital imaging features. It is important to realize that technicalities that may be acceptable to a computer scientist or programmer must be hidden from the medical professional. Tight collaboration with the clinical users to identify their needs and preferences is crucial in the development phase of the system, and this paper is a result of such a team effort.

In the next section, we describe the design goals for a medical workstation prototype for the rapid clinical review and diagnosis of digital cineradiographic images. (Apart from its potential practical use in daily clinical routine, the workstation has also functioned as a testbed for a collection of features to determine their utility and acceptance by the clinician.) In Section 3, we focus on some application-specific implementation issues. Then, in Section 4, we supply a detailed analysis of the clinical acceptance and utility of the various workstation features. This analysis is based on a series of detailed log files produced by the system's software which were acquired during a number of extensive clinical review sessions conducted by a group of eight expert clinical reviewers. We conclude with our plans for the second generation of the system.

2. DESIGN GOALS AND PURPOSE OF OUR MEDICAL WORKSTATION

In this section, we enumerate the design goals of our medical workstation and describe its particular clinical application. We then compare these design goals with the offerings of present commercial and public domain PACS and other image display and manipulation packages. It was the outcome of this comparison that led us to develop our own system instead of using and modifying one of the existing packages.

2.1 Design Goals

To accommodate clinicians with varying levels of experience in the use of computer windowing systems in a workstation environment, our main design goal was to create a functional but highly comprehensible Graphical User Interface (GUI). Great emphasis was placed on the user's ability to manage the large amount of image data efficiently. (For example, our clinical application specified that the total review time for each image set was not to last more than four minutes.) The image processing tools needed to be interactive and reversible in their effect on the image. It was emphasized that these tools should aid, and not confuse, the clinician in the diagnostic process. Finally, the design of the interface was to allow the user to learn all of the features quickly, while providing enough flexibility for personal preference. The design specifications were as follows:

- implement all functionality of the analog cine-viewer (such as animated viewing at variable frame rates and image magnification) using similar controls;
- implement lightbox style side-by-side viewing of images of different views;
- provide a special folder (the notebook queue) for future review of interesting images;
- provide a task-oriented set of image processing capabilities for interactive image enhancement;
- provide facilities for the acquisition, retrieval, loading, and reformatting of image sets;
- ensure fast image set retrieval and display;
- keep the GUI design as simple as possible by minimizing the number of push-buttons, pull-down menus, and pop-up windows;
- minimize the number of actions required by the user to achieve a certain task;
- convey a uniform appearance for all main and pop-up windows;
- separate image and file control operations from the image display;
- minimize the presentation of textual information;
- ensure readability and reduce operational errors with large buttons and button-based text;
- provide keyboard shortcuts and accelerators ("hot keys") for frequently used menu buttons;
- implement all features into an integrated user environment;
- use a modularized software design to facilitate program extension and expansion in a standard X-Windows/Motif environment;
- use low-cost, readily available computer equipment;
- incorporate automatic logging of user actions to enable statistical evaluation of the usage of the various workstation features.

2.2 Evaluation of Existing Implementations of Medical Workstations

Our first step was to evaluate currently available image review systems to see if one could be used or adapted to fit our design specifications. This search yielded a number of systems of varying complexity and sophistication. The public domain package xv (by J. Bradley, University of Pennsylvania) offers a wealth of image processing and composition tools, but does not provide an integrated framework for efficient multi-image management, such as image stack animation. NIH

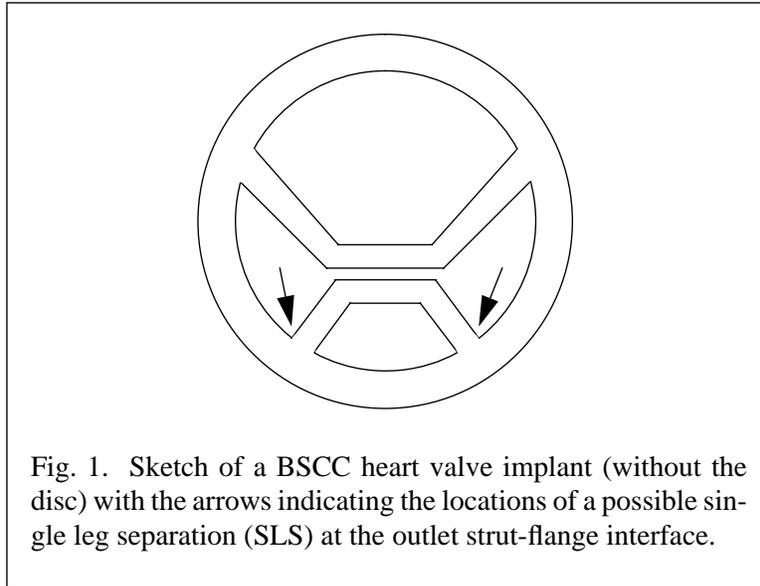
Image (by W. Rasband, National Institutes of Health) does provide both image processing and animated viewing, but an implementation of our image notebook queue cannot be done easily. It was also seen as a performance-limiting factor that NIH Image's use is confined to a Macintosh computer environment. Most of the commercially available PACS packages (such as VRS Allegro, Bscan and Shared Vision, as mentioned in the Introduction) provide a nicely integrated system with animated viewing and image processing utilities. However, they generally do not support an open architecture which hampers the integration of in-house developed routines or functionality into the system. One of the closest matches to our specifications was a public domain package called OSIRIS [7][8] (from the Digital Imaging Unit of the University Hospital of Geneva). OSIRIS provides both lightbox-style and animated viewing as well as Region Of Interest (ROI) based image processing. It is available on Unix workstations and is implemented as an open architecture in an object-oriented design (OOD). In OOD, a software system is decomposed into a hierarchy of self-contained objects that possess certain functionality. The OOD paradigm is increasingly used in software design in an effort to make software reusable and easily expandable into a common framework. The MedStation PACS [9] (from the University of Tübingen, Germany) is another example for such a system. By using OOD, new modules, e.g. for image processing or image file transfers, can be easily incorporated. With respect to our application, no major problems were anticipated with importing our task-oriented image processing and file handling routines into either OSIRIS or MedStation. We were also quite certain that we could tune the systems to match our specification of workstation-"look and feel" within an acceptable extent. However, given the structure imposed by these systems, there was no reasonably easy way to implement our notebook queue without compromising our functionality specifications significantly. It was therefore decided to develop a new system that met all our requirements exactly.

2.3 Clinical Application

The workstation was primarily designed for the clinical study of cineradiographic images of Björk-Shiley Convexo-Concave (BSCC) heart valve implants. This study was motivated by a number of occurrences of, so-called, Single Leg Separation (SLS) in these heart valves [10]. The SLS condition describes a valve where one of the two legs of the outlet strut, which, along with the inlet strut, holds the disc in place, breaks away from the flange. The second leg remains intact, and the valve continues to function normally. If the second leg subsequently also separates, disc escape occurs, a condition known as outlet strut fracture (OSF). OSF leads to sudden death of the patient in approximately two-thirds of the clinical cases. Fig. 1 depicts a sketch of a BSCC valve (without the disc) illustrating the locations of a possible SLS. In Fig. 3, an X-ray image of a BSCC valve acquired in a direct en-face viewing angle (also called "tunnel" view [11]) is shown.

Due to the high incidence of patient morbidity and mortality associated with reoperative valve replacement, prophylactic valve replacement was generally not recommended [12]. Instead, it was deemed more advisable to routinely monitor the patients by acoustic [13] or radiographic [14] procedures. In one possible scenario for patient monitoring by radiographic means, a BSCC heart valve patient would report annually or semi-annually to one of the international imaging sites to have a set of X-ray images taken. This image set would then be examined by a resident or remote expert reviewer using the here-described review workstation.

In a study involving ten BSCC heart valve patients with SLS, it was found that the typical size of a single leg separation at the outlet strut-flange intercept in an explanted valve under static, no-load conditions ranges from about 30 to 230 μm [15]-[17]. This rather small width of the SLS



necessitates the use of high resolution imaging techniques.

The first study (subject to a future publication, see [18]) utilizing the medical workstation compared two different imaging modalities using either a modified X-ray cineangiographic system (based on the Siemens DIMA Hicor) or a newly developed prototype of a high magnification X-ray system (manufactured by Feinfocus, Germany). The modified system produced cine film images, while the newly developed equipment generated digital images only. In the study, 17 sheep were implanted with BSCC valves in the mitral position, in which some of the valves were intact while others had SLS.

During a three day sequestered review session, a group of eight expert reviewers consisting of radiologists, cardiologists and one cardiac surgeon examined 112 image sets. Each image set consisted of three imaging views (left oblique, right oblique, and tunnel), and each imaging view was made up of 21 frames of 1024 x 1024 pixel 8-bit gray-level images for a total of 63 images per image set. Three main goals were identified for this study: 1) Compare the image sets generated by the two image acquisition systems, i.e. the modified and the newly developed cineangiographic equipment, in terms of their usefulness for SLS diagnosis; 2) Compare the digital review system against the conventional film-based cine viewer system for diagnostic accuracy. 3) Obtain insight about the clinician's acceptance, preferences, and interactions with the software and its features. In this paper, we will solely focus on results related to the third item on this list, while the first two topics will be subject of a future publication [18]. The outcome of the study dealing with the clinician's interaction with the software was then to be utilized to define the final implementation of the clinical workstation.

Although examples are presented from only one particular application, the concept and structure of the workstation are quite general: cineradiographic image data sets portraying various kinds of anatomical structures and physiological systems of the human body may be displayed and analyzed using our workstation as a generic platform.

3. IMPLEMENTATION OF OUR MEDICAL WORKSTATION

We will now describe the general implementation of our medical workstation according to the design goals specified in the previous section of this paper. Since a rather massive amount of image data (over 90 MByte per case) had to be managed in an efficient manner, some unique problems had to be addressed, with regards to both hardware and software. Finally, we will give a description of the task-oriented image processing routines used in the image processing toolbox.

3.1 General Description

The chosen platform was a Sun Microsystems SPARCstation™ 20 workstation running Sun's Solaris™ 2.3 Unix operating system. The workstation was equipped with 256 MB of DRAM, Fast SCSI-2 hard disks, and a 1280x1024x8 bit display on a 20-inch color monitor. Our system was also implemented on a Silicon Graphics (SGI) IRIS Indigo² XZ workstation running IRIX 5.3 with similar memory, disk, and monitor configurations. The display software was written in ANSI C and C++ programming language and OSF/Motif 1.2.3 windows environment.

For our prototype study, the program did not use file names explicitly. Instead, the image sets were organized in 28 blocks of 16 image sets each. Each image set was comprised of 21 frames of image triples, where each such triple was composed of a left oblique, a right oblique, and a tunnel view of the BSCC heart valve. Ideally, each such triple would correspond to the same time instant within the heart cycle. In our study, however, ECG gating was not used to trigger image acquisition, and thus images from these different views were not synchronized with each other.

The workstation was designed to operate in three modes of operation:

- **Preview Mode** - in which all three views are displayed on the same screen simultaneously. In this mode, spatially reduced images that fit in a 480 x 480 window are shown.
- **Single View Mode** - in which only one view at full resolution is displayed in a 980 x 980 window.
- **Selected Frame Mode** - in which the user can review and re-analyze critical image frames that were selected in Single View mode. This is our notebook folder.

For the following description please refer to Fig. 2, which shows the workstation in Preview Mode. To view a particular image set, the user first chooses an image set block. Depressing "Select Block" on the menu bar in the left portion of the interface screen in Fig. 2 pops up a menu which presents the blocks currently available on disk. After a block is chosen, the user steps sequentially through the image sets within this block by depressing the "Next Case" button repeatedly. However, image sets are not actually read into memory and displayed until the "Load" button is depressed. While the system loads the 63 images of the set (plus other image data, as will be discussed later) from optical or magnetic disk into memory, spatially compressed versions of the images are presented to the user in the three preview windows (shown to the right of the menu bar in Fig. 2) at a fixed update rate of 3 seconds per image triple. By previewing the images one triple at a time, the user is given the opportunity to gain a quick overview while the concurrent loading of the full image data set is in progress.

Upon completion of the loading phase, the preview windows reset to show the first image triple in the image set. At this point the user switches to Single View Mode by selecting any of the three buttons labeled "Tunnel", "Left Oblique", or "Right Oblique". Fig. 3 shows the system displaying the ninth frame of image set 1 within block 1 in Tunnel view. Note that the frame number and view

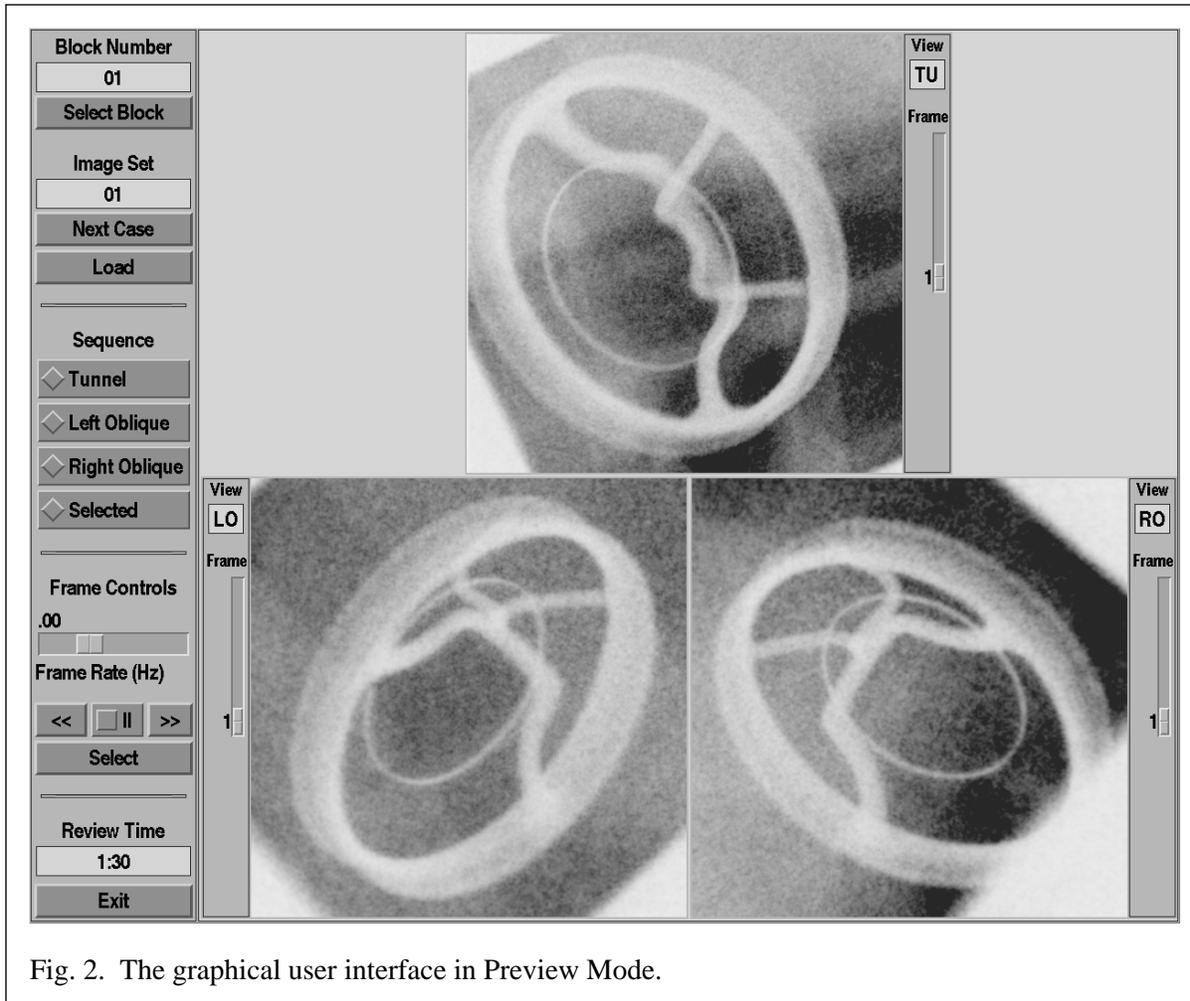


Fig. 2. The graphical user interface in Preview Mode.

are indicated by a label and a slider bar, respectively, on the right side of the interface screen, while the block number and image set number are displayed in two labels within the top section of the main menu bar. A display at the bottom of the main menu panel contains a digital clock indicating the time expired since image set load time.

The menu section labeled “Frame Controls” offers the reviewer several ways to step through the image sequence. By using the “<<” and “>>” buttons, respectively, the frames are advanced manually in the backward and forward direction, a method functionally analogous to reverse and forward in a cine-film viewing workstation. Since this is tedious for a large number of frames, the images can also be displayed in an animated, cine-type mode. By means of the slider bar, the frame rate can be varied interactively within a bound of -3 to 7 frames/sec. (The negative frame rate corresponds to playing the frames backwards.) Animation can be temporarily paused by depressing the “||” button. Views may be switched at any time, and during paused animation the frames may be advanced manually. This combination of reverse, forward, pause, and animation allows the user to step quickly through the image data, only stopping at interesting frames. To signal the user that he has reached the end or the beginning of an image set, a bell is sounded when moving between the first and the last frame of the sequence.

Since there is a large number of images in a set, and there may only be a few images that require

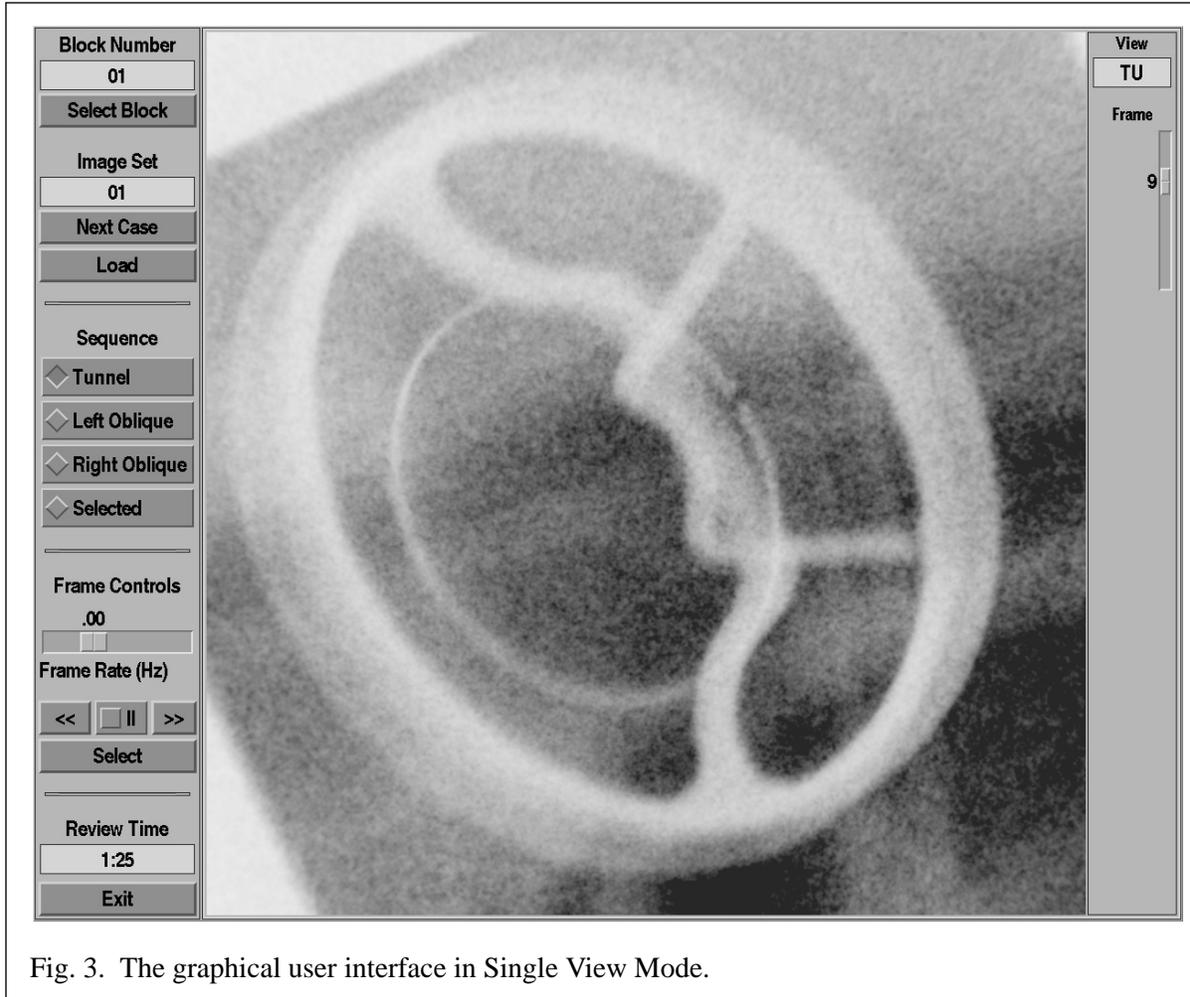


Fig. 3. The graphical user interface in Single View Mode.

a second review, these critical frames can be tagged by depressing the “Select” button when displayed in the window during Single View mode. The marked frames are saved into a separate notebook queue, ordered by view and sequence number, and can be viewed by depressing the “Selected” button. All frame control features apply in this mode, called the Selected Frame Mode, as well.

Often the magnification of a suspicious area in an image is helpful in deciding if a pathologic condition, such as a tumor, an arterial stenosis, or an SLS in one of the outlet strut legs of the BSCC heart valve, is present. Our system supports a magnifying glass feature bearing a fixed magnification factor of 2. Here, the user moves the mouse pointer to the location of interest and clicks the left mouse button. The system then superimposes a magnified subimage of a 256x256 pixel neighborhood centered around the point of interest onto the displayed image. Fig. 4a shows an image ROI, here the welding area of a BSCC heart valve strut, and Fig. 4b shows the same ROI magnified by application of the magnifying glass. When the mouse button is clicked outside the ROI, the magnification is removed, and the image is displayed in its unprocessed form.

In some instances it is desirable to enhance certain image characteristics, such as feature edges, or to attenuate undesired image characteristics, such as noise or excessive level of detail. The X-ray images in our study contained a considerable amount of quantum noise. Our system provides

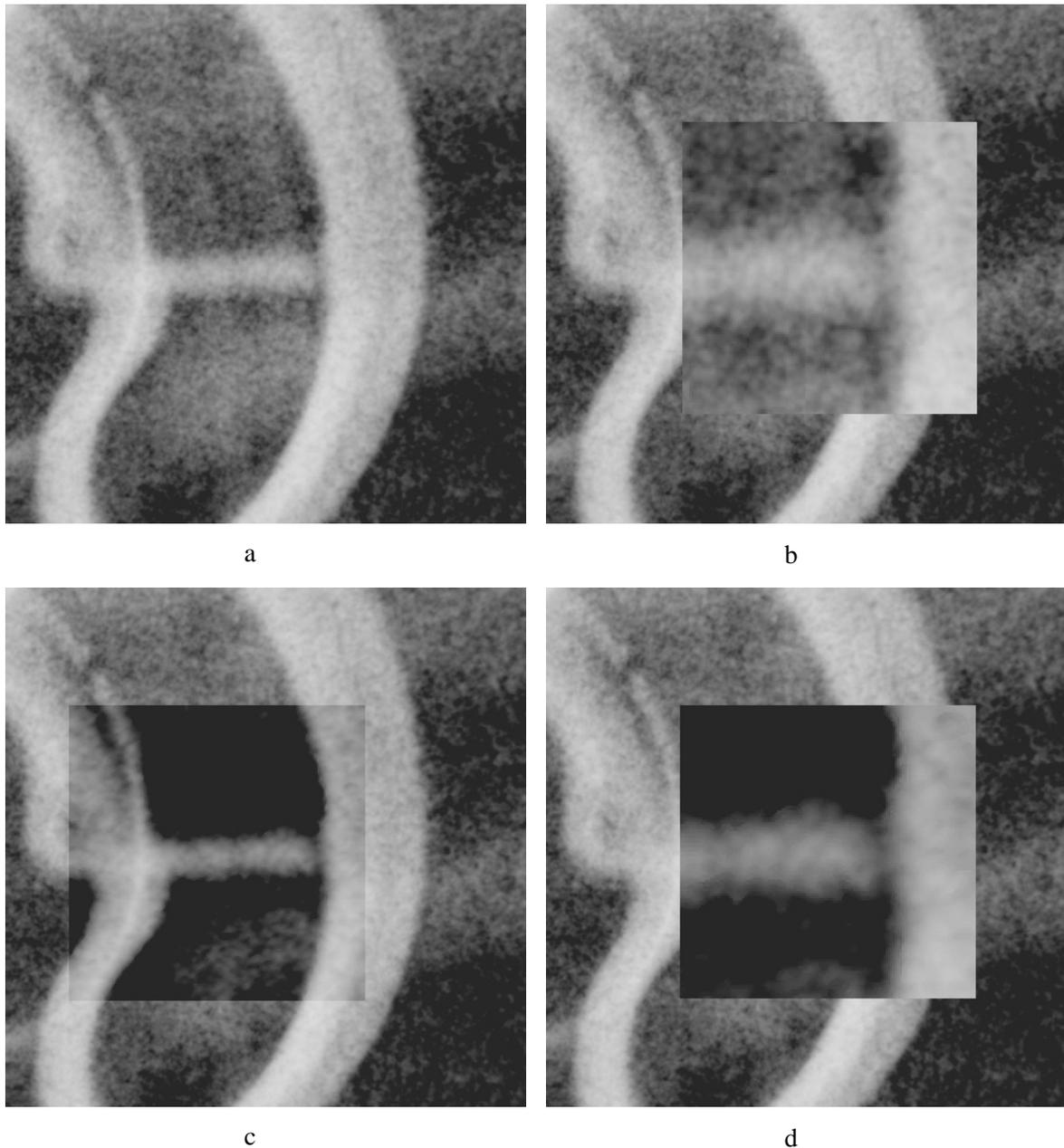


Fig. 4. An image region-of-interest (here the welding point of the valve strut to the strut flange): a) in unprocessed form, b) magnified by a factor of 2, c) processed by an adaptive trimmed mean filter, d) processed as in c) but magnified by a factor of 2.

an adaptive trimmed mean filter [26] that is particularly designed to preserve edges while removing background noise. The filter is locally applied in the same fashion as the magnification feature described earlier, but triggered by the center mouse button. While this filter works best for our application, other filters are under consideration, such as a Laplacian edge detector, a median low-pass filter, an adaptive contrast enhancement, or a thresholding operator (see, e.g., [27]). The system also provides a magnified version of the processed area by depressing the right mouse button.

Fig. 4c and Fig. 4d illustrate the result of applying the filter, unzoomed and zoomed, respectively, to the ROI.

When an image is selected for the notebook queue, the processing state is saved along with it. Therefore, the reviewer, when in Selected Frame mode, does not have to remember and replicate the processing steps that worked best for the image at hand. This speeds up the evaluation process, and avoids duplication of effort. Note that at any time, the user may return to the original image and perform alternative manipulations.

In some cases, the amount of mouse travel across the screen can be extensive, for example if the user advances the frames manually or frequently selects frames into the notebook queue, and, at the same time, frequently picks ROI's for processing or zooming. This can become very tiresome and inefficient. The problem was overcome by associating *hot keys* with some of the menu functions: in particular, the "<<", "||", ">>", and "Select" buttons were duplicated by the 'Z', 'X', 'C', and 'S' keys on the terminal's keyboard, respectively. The locations of these keys on an American keyboard map directly into the arrangement of the corresponding menu buttons on the menu panel. Using hot keys, a right-handed user is now able to perform the menu functions on a keyboard with his left hand, and operate the mouse with his right hand. For left-handed users, the hot key arrangement is easily changed via a program start-up file.

3.2 Measures to Enable Rapid Image Display and Processing

We stated before that it was one of the prime goals of our workstation to enable the clinician to review an image set in a rapid (but nonetheless accurate) fashion. However, the time required by the workstation for loading and processing (e.g. magnifying and filtering) of the images compromised this goal. The following two measures were introduced to minimize the time the reviewer spends idle in front of the computer, waiting for images to be loaded or processed.

3.2.1 Precomputed processed subimages

Magnification and adaptive filter operations introduced delays that were deemed interruptive by some reviewers. While magnification took on the order of 5 secs, the adaptive filter generally needed more than 15 secs to complete. While the former delay would be tolerable, the latter represents a time span that is hardly in the interactive range. Therefore, in our prototype system, the magnification and filtering operations were not performed on the fly but were implemented by superimposing a pre-computed subimage onto the appropriate location in the full image. Thus, whenever the mouse was clicked within an image area (i.e. an ROI) for which a precomputed subimage was available, this subimage was displayed at its fixed image location. This is in contrast to on-the-fly calculation where the computed subimage is dependent on the location of the mouse click. There are a number of advantages to the approach of pre-computing the ROI based subimages: 1) The results of magnification and adaptive filtering are instantly visible to the user; 2) Spatially sensitive operations such as our adaptive filtering operation are not dependent on the exact location at which the user has clicked the mouse. 3) The possible user operations are limited to pre-defined areas, a constraint that may prevent errors for diagnosis. The downside of this approach, however, is the incurred effort of pre-computing, storing, loading, and managing the inserted images. Since in our application the main focus was the strut-flange interface of the valves, we required three subimages per visible valve strut leg: one each for magnification, filtering, and magnified filtering. (Recall that tunnel views have two such legs visible while oblique views only show one.) With a subimage size of 256x256 pixels and 42 oblique views and 21 tunnel views, we

therefore computed 252 subimages per image set which is equivalent to about 16MB storage (about 20% of the regular image data load).

Apart from the computational savings during the review, a second reason for pre-computing the inserts was the accuracy of the adaptive filter: In order to obtain maximum contrast at the strut-flange interface it is very important to apply the adaptive filter on a region that has the strut-flange interface in its center. An automated procedure [19] can be applied to identify the exact location of the strut-flange interface, which then is used as the ROI's center point around which the filtered and magnified subimages were computed.

We admit that this automatic procedure to determine the appropriate position of the ROI incorporates a great amount of knowledge about the nature of the image data and is therefore certainly not for general use. In this respect, we would rather like our method to be seen as an example how domain knowledge and image processing methods can be integrated to streamline and focus the image review process. As clinical workstations are in many cases applied in a high volume routine setting, it may often be worth the effort to develop methods similar to the ones we have used. The many advances made in today's medical knowledge-based systems (see e.g. [20]) will undoubtedly facilitate this task.

This pre-processing step, which also included the computation of the spatially compressed preview images, required less than 1 hour per image set and was usually done as a batch process over night. Clearly, the availability of more powerful computer equipment may eliminate the need for computing the subimages beforehand. For our particular application, one solution that would reduce the memory overhead but maintain the accuracy of the adaptive filter was to store only the location of the strut-flange interface for each image (determined in a pre-processing step) and to compute the filtered image on-line. In order to save repeated calculations of the same subimage, one could cache it into memory and store the pointer for later use. Similar compromise solutions may be appropriate in other applications.

3.2.2 Direct memory mapping

Another user-oriented design specification for the project required all images in an image set to be loaded and displayed within 50 seconds. As one image set contained over 90MB of data distributed throughout 378 files (divided in 63 spatially-compressed preview images of size 480x480, 63 full-resolution images of size 980x980, and 252 pre-processed subimages), this specification required an effective sustained transfer rate of 1.8 MB/sec.

All images are stored in an array of X Window pixmap resources. In X Windows (see e.g [21]), pixmaps are treated like "invisible windows" that are resident on the display workstation. Their contents can be rapidly copied to the screen. By storing a series of images in pixmaps one is able to animate even large images at a very high rate.

There is one pixmap allocated for each image in the set. To fill an image pixmap, the image data must first be read into an *XImage* data structure which is then transferred to the pixmap. Therefore, when a study is loaded, two memory transfers are necessary per image: first, the image data is read from disk using the Unix buffered file *read()* library call, then the data is transferred to the pixmap using the X Window routine *XPutImage()*. The buffered file *read()* incurs an overhead from the operating system's page replacement algorithm, which, after a sufficiently large number of image files have been read, compromises fast image set loading by producing a large number of page faults in virtual memory. This circumstance oftentimes doubled the time required to read an image

set into the system. However, we can bypass the buffered *read()* by directly mapping the disk file to a virtual address space in main memory and reading the mapped data straight into the pixmap [22][23]. This is done by using the *mmap()* facility available in most UNIX implementations (such as SGI's IRIX and Sun's SOLARIS). Using *mmap()* and advising it of the fact that only sequential data read's will be performed results in a significant improvement of I/O performance. A simplified code segment for this mechanism is given in Fig. 5. Since the file systems on medical workstations may often be NFS-mounted, it is important to note that *mmap()* also functions over these links.

```

imageFileNameList: array of the names of the images in the image set
pixmapArray: array of X Window pixmap resources
xImage: instance of an XWindows XImage resource, initialized for the common image properties
xDisplay: ID of the workstation's X Windows display

LoadImageSetIntoPixmaps(imageFileNameList)
    imageIndex = 0;

    /* all images are composed of a header and the data and have the same size */
    dataSpan = size(imageHeader)+size(imageData);

    for each imageFileName in imageFileNameList

        /* open an image file descriptor for read-only access */
        fileID = open(imageFileName, O_RDONLY);

        /* establish a private virtual mapping of the image file on disk to the data field of xImage */
        xImage.data =
            mmap(..., dataSpan, PROT_READ, MAP_PRIVATE, fileID,...) + size(imageHeader);

        /* copy the image data into the pixmap */
        XPutImage(xDisplay, pixmapArray[imageIndex], ..., xImage, ...);

        /* release the virtual memory */
        munmap(xImage.data,dataSpan);

        /* close the image file */
        close(fileID);

```

Fig. 5. Pseudocode to illustrate fast loading of an array of X Windows pixmap resources by mapping a virtual address space to the image files on disk.

3.2.3 Double Frame Buffering

As mentioned before, updating an animated image display in X Windows is performed most quickly by continuously copying image pixmaps into the display window in a synchronized fashion using *XCopyArea()*. With *XCopyArea()*, the new image is written to the display window line by line from top to bottom. This is quite noticeable on large display windows and leads to flickering at high frame rates. One solution to this problem is double frame buffering: Here, the workstation framebuffer is divided into two portions: the front frame buffer and the back frame buffer. The currently displayed image is held in the front frame buffer, while the next image in the sequence is written to the (undisplayed) background buffer. Then, at frame update time, front and back buffers

are switched for screen display of the new image. In this scheme, the writing of the new image is never visible to the user since it occurs in the background while the previous image is displayed. The new image is only unveiled to the user after writing to the back frame buffer is completed. As standard X Windows (X11, Release 6) does not provide true double frame buffering, we have implemented double frame buffering with OpenGL [24] on the SGI version of our software. OpenGL is the portable version of the graphics library GL that was originally developed by SGI. This standard has recently been adopted by a variety of other workstation manufacturers implementing OpenGL either in software or with add-on hardware. In order to use OpenGL within an X/Motif interface, one simply replaces the Motif DrawingArea widget by the GlxMDraw widget which accepts OpenGL calls. Images are no longer stored in pixmaps, but in pixel arrays that are copied to the GlxMDraw widget during animation. Fig. 6 shows a small code segment to illustrate the idea of double-buffering with OpenGL. We do not want to leave this section without mentioning that efforts have recently been made to also extend X Windows for double and multi-buffering. The current document with regards to establishing a standard in this issue is [25]. We, however, have not tried these extensions.

```

DisplayNewImageFrame(imageData)

    /* write the image to the back frame buffer */
    glDrawPixels(imageWidth, imageHeight, ..., imageData);

    /* swap back frame buffer with front frame buffer */
    glXSwapbuffers(xDisplay,xWindow);

AnimateImages(frameRate, startIndex)

    /* initialize frame pointer */
    imageIndex=startIndex;

    /* while the user does not change the frame rate or stops animation */
    while no interrupt

        /* display the next image in the sequence */
        DisplayNewImageFrame(imageData[imageIndex]);

        /* pause according to the specified frame rate */
        Pause(1/frameRate);

        /* point to next image in the sequence */
        imageIndex=(imageIndex+1) mod numberOfImages;

```

Fig. 6. Pseudocode illustrating the use of OpenGL double buffering for image animation. The routine **DisplayNewImageFrame**() is also used when advancing frames manually.

3.3 Implementation of the Notebook Queue

The insertion of frames into the notebook queue must be performed quickly, especially at large animation frame rates. We also wish to tag a frame with additional information if the user has

zoomed or processed some region in the image at selection time. We implemented the notebook queue as a table with each row in the table corresponding to an image in the original sequence. When an image is selected, the pointer to its pixmap is entered at the appropriate row in the notebook table and a selection-bit is set. An additional series of bits is provided for each row indicating the type of image processing (i.e., zoom and filter) performed on the image at the time of insertion. When the images in the queue are animated, the table is scanned row by row, and if a row's selection-bit is set, the image is displayed with the corresponding processed region superimposed on it. If the subimages were not precomputed, it would be wasteful to perform zooming or filtering every time an image in the notebook queue is displayed. In this case, one would cache the image processed region into memory at the time an image is inserted and store the respective memory pointer in the frame's row of the notebook queue.

Note that when an enhanced image is linked to the notebook queue, we do not just store a picture of the enhanced image. Instead, the raw image and image processing actions still co-exist. It is only at display time that the two pieces of information are merged into one image. Therefore, when browsing through the notebook queue, all image enhancements are easily reversible and changeable.

4. ANALYSIS OF WORKSTATION FEATURE USAGE

The analysis of the review log files that were acquired throughout the review sessions is summarized in Table 1. Here we see that the average review time was 222 seconds, which meets the 240 second design goal. We also observe that each user interacted with the features available in the interface differently. This is reflected in the relatively large overall standard deviations in almost all logged categories. For example, while some reviewers simply single-stepped through the frame sequences, others chose to animate the frames (usually at a frame rate of 1 Hz) to step through the image sequences automatically. These reviewers would then only pause for critical frames which required a closer evaluation. Another implementation feature which resulted in varying preference was the use of hot keys versus mouse interaction. It was observed that there was a slight preference for hot keys over the use of mouse interaction (42 vs. 39 hits per image set). The Selected Frame Mode feature and the capability of displaying selected image regions in zoomed or image processed form was used by all reviewers throughout the review session. It was observed that most reviewers initially experimented with all alternative strategies provided to perform a certain task, e.g. animated vs. manual viewing, but then, after a relatively short trial period, only used one, the most preferred, method for each such task.

5. FUTURE WORK

As mentioned before, one purpose of our medical workstation prototype was to gain insight about user preferences and acceptance of the system's features. This knowledge was then to be utilized in the design of the final clinical version of the medical workstation. We will now elaborate on a possible implementation of this system.

While the image file organization (in blocks and sets) in the prototype workstation was designed to comply with rules for statistical hypothesis testing, in a clinical system file browsers will be provided to select a patient study from a UNIX file system located on an optical or magnetic disk. As in the prototype system, a patient study will consist of the three image sequences (tunnel, left oblique, right oblique) and a patient information file that stores relevant personal patient data,

Re-viewer	Review time (sec)	Image processing (selections)			Se-lected Frames	Interaction (hits)		Viewing (frames)	
		Zoom	Filter	Zoom Filter		Hot Key	But-ton	Ani-mated	Ma-nual
A	242.3	0.8	0.9	1.9	2.5	13.4	4.5	130.1	7.9
B	175.8	4.1	19.0	4.7	0.0	77.7	0.0	0.0	77.7
C	275.6	0.1	7.2	0.1	8.3	0.0	81.7	13.3	73.0
D	204.4	3.2	6.5	3.6	0.8	61.9	13.0	1.1	72.3
E	225.6	0.1	0.1	0.1	1.3	0.1	84.6	0.4	80.6
F	250.4	1.0	3.7	1.0	1.3	50.8	63.3	9.3	112.7
G	190.4	0.3	1.0	4.6	2.2	71.1	0.1	0.8	68.9
H	212.1	4.3	7.0	3.3	0.9	60.1	62.9	0.0	122.0
Mean	222.1	1.7	5.7	2.4	2.1	41.9	38.8	19.4	76.9
σ	33.0	1.8	6.1	1.9	2.6	32.2	37.7	45.0	34.2

Table 1. Analysis of GUI usage for the eight reviewers A...H. Figures for individual reviewers correspond to the means for their assigned cases. The overall means and standard deviations in the bottom rows are derived from the individual reviewer means.

acquisition related data (such as the number of acquired images per view and scanner parameters), and comments of the radiologist or cardiologist at the imaging site. The patient information file will be designed to aid the expert reviewer in navigating through the patient studies available for review. At the imaging sites, the X-ray scanner will place the image files onto an optical disk or DAT tape. An acquisition module in the imaging site resident workstation will then translate the raw image files into the workstation's DICOM compliant format (if necessary) and merge them with the patient data into a patient study. Since the workstation program will run on any UNIX machine, expert reviewers will not need to be resident at the imaging site. A non-resident expert reviewer will either transfer the patient study electronically (via ftp) to his workstation using a network module in the workstation software, or request the files to be sent by mail on an exchangeable medium such as optical disk or DAT tape. In the former case, the patient information file and maybe a set of low resolution preview images could be loaded first to identify the correct study before the entire large-scale image data sequence would be transferred over the network. The network module could be based on the currently evolving World Wide Web browser technology.

In this clinical workstation we also plan to a replace the fixed-size ROIs by ROIs that can be dynamically defined by a rubber-band technique. We also would like to supplement our image processing toolbox by some more general routines, such as median filters, highpass filters for edge

detection, and additional levels of zoom. If the workstation platform is sufficiently powerful, we also intend to use on-the-fly image processing instead of the pre-computed approach used in the prototype. To accommodate images exceeding the size of our screens, we need to implement a grab-and-pan facility and a set of horizontal and vertical scrollbars. We also noticed during our experiments that being able to return to the lightbox display used in the preview stage of our workstation would add a great deal of functionality to the workstation. In this lightbox mode, both image processing tools and animation facilities would be fully functional, and a fourth window would be added to display the image processing results. With respect to the notebook queue, we plan to add a button with which selected images can be removed from the queue. Another new feature will be a facility that allows the user to compile a new study from subsets of current patient studies. The notebook queue represents a natural example of such a subset.

Oftentimes the digital images obtained from the X-ray scanner contain the object of interest embedded in a large uniform region which carries no image information and should therefore be removed at image acquisition time. As this border region is usually relatively distinct from the object area, we plan to include an automatic cropping algorithm in the workstation's image acquisition module.

6. CONCLUSIONS

We have demonstrated the design and implementation of a clinical image review system for the efficient analysis and display of multi-frame and multi-view cineradiographic digital data on a standard workstation. We feel that the design of our system is generally useful for many clinical settings in which large sets of image data must be reviewed. Our design differs from other currently available medical workstations in several ways. 1) It offers an efficient digital notebook folder in which the reviewer saves interesting images along with image enhancement measures for later closer review. In this queue, image enhancement information is stored separately from the raw image enabling the user to easily retrieve the original image. Since only a pointer to the image, and, in some cases, a small subimage is stored, the space consumption of the queue is minor. 2) Our design incorporates several measures to enable rapid and artifact-free image display: double frame buffering to prevent animation flicker, storage of pre-processed images for interactive filtering operations, and direct memory mapping for fast transfer of images into the system. 3) A task-oriented adaptive image processing algorithm is used. While the image processing module could have been accommodated fairly easily into the currently available systems, the first two features would have required extensive changes to these systems.

To provide the clinician with a maximum amount of flexibility and make the system easy to use, we provided a variety of ways a certain task could be achieved. The analysis of user actions during the review sessions suggest, that clinical reviewers vary greatly in their usage of the workstation and its features if they are given the opportunity to tailor the system to their own preferences. It seems therefore crucial for a digital clinical image review system, such as the one described here, to provide a certain amount of flexibility to the user.

Acknowledgments

The authors would like to thank the Shiley Heart Valve Research Center, Irvine, Ca for funding this project. We also thank its staff, in particular Dr. James Chandler, and the sequestered expert review panel for many inspiring comments and suggestions during the development of the soft-

ware. Finally, we would like to thank the anonymous reviewer for the elaborate comments that helped to grow this paper into a much more detailed and clearer presentation of our work.

REFERENCES

- [1] C. Honeyman, M.M. Frost, W. Huda, W. Loeffler, M. Ott, and E.V. Staab, "Picture Archiving and Communications Systems (PACS)," *Current Problems in Diagnostic Radiology*, Vol. 23, No. 4, pp. 101-58, 1994.
- [2] A.W. Wong, H.K. Huang, R.L. Arenson, and J.K. Lee, "Digital archive system for radiologic images," *Radiographics*, Vol. 14, No. 5, pp. 1119-26, 1994.
- [3] O. Ratib, "From multimodality digital imaging to multimedia patient record," *Computerized Medical Imaging and Graphics*, vol. 18, no. 2, pp. 59-65, 1994.
- [4] B.T. Ho, O. Ratib, and S.C. Horii, "PACS workstation design," *Computerized Medical Imaging and Graphics*, vol. 15, no. 3, pp. 147-155, 1991.
- [5] F.W. Prior, "Specifying DICOM compliance for modality interfaces," *Radiographics*, Vol. 13, No. 6, pp. 1381-8, 1993.
- [6] O. Ratib O, H. Hoehn, C. Girard, and C. Parisot, "PAPYRUS 3.0: DICOM-compatible file format," *Medical Informatics*, Vol. 19, No. 2, pp. 171-8, 1994.
- [7] Y. Ligier, O. Ratib, M. Logean, C. Girard, R. Perrier, and J.R. Scherrer, "Object-oriented design of medical imaging software," *Computerized Medical Imaging and Graphics*, vol. 18, no. 2, pp. 125-135, 1994.
- [8] O. Ratib, Y. Ligier, and J.R. Scherrer, "Digital image management and communication in medicine," *Computerized Medical Imaging and Graphics*, vol. 18, no. 2, pp. 73-84, 1994.
- [9] H. Ehrlicke, T. Grunert, T. Buck, R. Kolb, and M. Skalej, "Medical workstations for applied imaging and graphics research," *Computerized Medical Imaging and Graphics*, vol. 18, no. 6, pp. 403-411, 1994.
- [10] L. F. Hiratzka, N. T. Kouchoukos, G. L. Grunkemeier, D. C. Miller, H. E. Scully, and A. S. Weschler, "Outlet Strut Fracture of the Bjork-Shiley 60 Convexo-Concave Valve: Current Information and Recommendations for Patient Care," *Journal of the American College of Cardiology*, vol 11, pp. 1130-1137, 1988.
- [11] A.H. Albofathi, J.L. Hirsch, S.E. Nissen, D.W. Wieting, and J.G. Chandler, "Effect of valve profiling and increasing kVp on in situ radiographic detection of single leg separations of the Björk-Shiley convexo-concave heart valve," *Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, No. 16, pp. 528-529, 1994.
- [12] J.D. Birkmeyer, C.A.S. Marrin, and G.T. O'Connor, "Should patients with Björk-Shiley valves undergo prophylactic replacement?," *Lancet*, no. 340, pp. 520-523, 1992.
- [13] A.C. Eberhardt, C.E. Chassaing, R.S. Inderbitzen, D.W. Wieting, "Acoustic evaluation of progressive failure of Björk-Shiley Convexo-Concave heart valves," *Proceedings of the 7th IEEE Symposium on Computer-Based Medical Systems*, Winston-Salem, NC, June 1--12, 1994.
- [14] W.W. O'Neill, J.G. Chandler, R.E. Gordon, et.al., "Radiographic detection of strut separations in Björk-Shiley Convexo-Concave mitral valves," *New England Journal of Medicine*, no. 333, pp. 414-419, 1995.
- [15] K.A. Powell, S.N. Ganobcik, E.E. Herderick, K. Mueller, J.G. Chandler, S.N. Nissen, and J.F. Cornhill, "Quantitative detection and classification of single-leg separations in the outlet struts of Björk-Shiley convexo-concave heart valves," *1994 World Congress on Medical Physics and Biomedical Engineering*, Rio de Janeiro, Brazil.
- [16] K.A. Powell, J.F. Cornhill, S.N. Nissen, J.G. Chandler, A.H. Abolfathi, E.L. LaPresto, E.E. Herderick, K. Mueller, E. Cesmeli, and S. Chandra, "Digital radiographic analysis of single-leg separations (SLS) in the outlet struts of Bjork-Shiley convexo-concave valves," *Computer Assisted Radiology Car'93*, Berlin, Germany, 1993.
- [17] K.A. Powell, N. Obuchowski, K. Mueller, C. Hwang, S.N. Nissen, and J.F. Cornhill, "Quantitative detection and classification of single-leg separations in the outlet struts of Björk-Shiley convexo-concave heart valves," submitted to *Circulation*, 1995.
- [18] J. R. Landis, Director of the Center for Biostatistics and Epidemiology at Hershey Medical Center, *Personal Communication*.

- [19] E. Cesmeli, K.A. Powell, and J.F. Cornhill, "An Automated Method To Detect Single-Leg Separation In Radiographic Images Of Bjork-Shiley 60 And 70 Convexo-Concave Heart Valves," *Proceedings of the EMBC'95*, 1995 Annual Conference of IEEE Engineering in Medicine and Biology Society, pp. 427-428.
- [20] L.S. Wilson, B.D. Doust, M.S. Brown, and R.W. Gill, "Processing scheme for computer interpretation by using anatomic knowledge," *Radiology*, vol 197(P), special issue for the RSNA'95 conference, November 1995, p. 448.
- [21] A. Nye, *Xlib Programming Manual*, O'Reilly & Associates, Sebastopol:California, 1992.
- [22] J.R. Graham, "Programmer Performance Series: Basics," *SunExpert Magazine*, July 1995, pp. 53-60.
- [23] A. S. Tanenbaum, *Operating Systems: Design and Implementation*, Prentice-Hall, 1987.
- [24] J. Neider, T. Davis, and M. Woo, *Open GL Programming Guide*, Addison-Wesley, 1993.
- [25] J. Friedberg, L. Seiler, and J. Vroom, "Extending X for double-buffering, multi-buffering, and stereo," MIT and Digital Equipment Corporation, obtainable by anonymous ftp from <ftp.x.org/pub/DOCS/DBE/dbe.PS.Z>, 1990.
- [26] A. Restrepo and A.C. Bovik, "Adaptive Trimmed Mean Filters for Image Restoration," *IEEE Transactions of Acoustics, Speech, and Signal Processing*, Vol 36, No 8., 1988.
- [27] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.