The SIVA Demonstration Gallery for Signal, Image, and Video Processing Education

Umesh Rajashekar, *Student Member, IEEE*, George C. Panayi, Frank P. Baumgartner, and Alan C. Bovik, *Fellow, IEEE*

***Abstract*—The techniques of digital signal processing (DSP) and digital image processing (DIP) have found a myriad of applications in diverse fields of scientific, commercial, and technical endeavor. DSP and DIP education needs to cater to a wide spectrum of people from different educational backgrounds. This paper describes tools and techniques that facilitate a gentle introduction to fascinating concepts in signal and image processing. Novel LabVIEW- and MATLAB-based demonstrations are presented, which, when supplemented with Web-based class lectures, help to illustrate the power and beauty of signal and image-processing algorithms. Equipped with informative visualizations and a user-friendly interface, these modules are currently being used effectively in a classroom environment for teaching DSP and DIP at the University of Texas at Austin (UT-Austin). Most demonstrations use audio and image signals to give students a flavor of real-world applications of signal and image processing. This paper is also intended to provide a library of more than 50 visualization modules that accentuate the intuitive aspects of DSP algorithms as a free didactic tool to the broad signal and image-processing community.**

***Index Terms*—Demonstration library, interactive education, multidisciplinary, signal and image-processing education, visualization.**

I. INTRODUCTION

THE PRINCIPLES of digital signal processing (DSP) and digital image processing (DIP) have spread their roots far and wide, as evidenced by their applications to a very wide spectrum of problems. Astronomy, genetics, remote sensing, video communications, and ultrasonic imaging are just a tiny sampling of applications that reflect the multidisciplinary nature of DSP and DIP. Applications in DSP and DIP combine concepts from a variety of areas, such as visual psychophysics, audio and acoustics, optics, and computer science. Although well rooted in advanced mathematics unfamiliar to a majority of the general audience that uses it, the theory of DSP and DIP needs to be made "*accessible*" to practitioners from diverse backgrounds. Presenting such an interdisciplinary topic with perspicuity to a heterogeneous audience is challenging. With the recent trend of DSP drifting lower down the curriculum to even high school, as in the Infinity project [3], and the importance of designing DSP as a first course in electrical and computer engineering [4], the need for tools and techniques to present signal processing with minimal math to a nontechnical audience is becoming increasingly significant.

The main hurdle faced by a novice in DSP is that the mathematics that describes fundamental concepts can cloud intuition. To reinforce concrete fundamental concepts, most introductory courses assign computer-based DSP exercises. However, more often than not, the learning curve involved in becoming familiarized with the software detracts the student from assimilating the concept. More effective techniques to uncover the intuition behind "murky" equations are visualization tools that facilitate the aural and visual consumption of information. A ready-to-use set of demonstrations illustrating the concepts that the instructor deems important can help the student to begin experimenting and assimilating immediately without having to bother about programming intricacies. This situation also encourages students to experiment with their desired inputs at their leisure. Further, such tools bolster the success of distance learning by facilitating interactive education and are the topic of discussion in this paper.

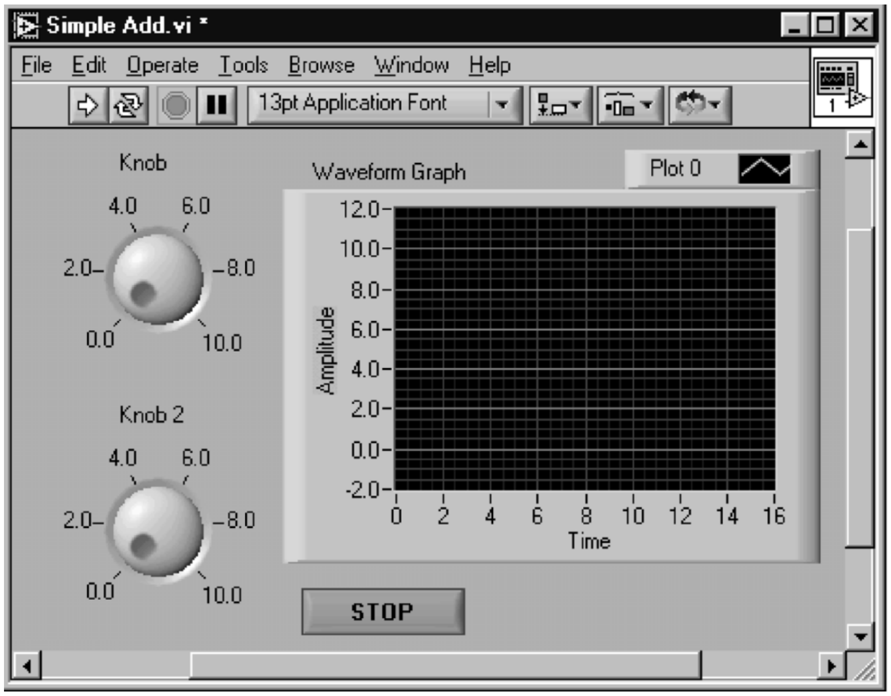
There have been many significant contributions to DSP educational tools developed primarily with MATLAB [1], [5]–[8], LabVIEW [2], [9], Java [10]–[12], and Mathematica [13]. Java-based tools enjoy the advantage of inherent platform independence [14] over most other implementations. Further, they do not need the user to have any specific software installed on their local machine, making this option most economical to students. Java-based tools are also inherently suitable for a Web-based education system, since they can be easily integrated into Web browsers. Common gateway interface has also been used for handling signal-processing routines, with Java used for the user interface [15]. While this approach sounds optimistic, one must bear in mind that developing educational tools with software specially designed for signal-processing applications is obviously less tedious. Many of these demonstrations have focused significantly on concepts such as the z transform [16] [17] and a few other elementary concepts in DSP [6].

This paper describes the Signal, Image, and Video Audiovisualization Demonstration Gallery (SIVA), comprised of two powerful visualization modules rich in fundamental concepts, developed and used at the University of Texas at Austin (UT-Austin) for signal- and image-processing courses. The modules consist of a LabVIEW-based demonstration suite for an undergraduate course titled "Digital Image Processing and Video Processing" and a MATLAB demonstration package for a graduate course titled "Digital Signal Processing." SIVA is tailored for an in-class and online instruction ambience with a powerful point-and-click type of graphical user interface (GUI). The demos have been seamlessly integrated into the class notes to provide contextual

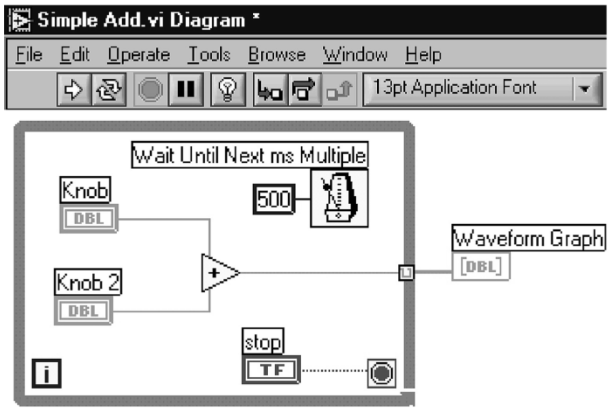
Manuscript received October 19, 2001; revised March 1, 2002. Portions of this manuscript appeared in [1], [2].

The authors are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, 78712-1084, USA (e-mail: umesh@ece.utexas.edu; baumgart@ece.utexas.edu; bovik@ece.utexas.edu).

Publisher Item Identifier 10.1109/TE.2002.804392.



(a)



(b)

Fig. 1. Typical GUI development environment in LabVIEW. (a) Front panel. (b) Block diagram.

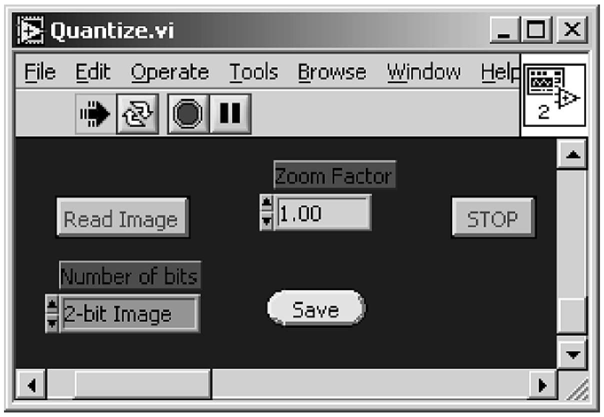
illustrations. This suite, with more than 50 demos spans a gamut of concepts in signal and image processing, much wider in scope than most of the current endeavors of similar flavor mentioned previously.

MATLAB and LabVIEW are used as the development tools for these demonstration modules. MATLAB has become the software of choice for DSP simulations as reflected by the ubiquity of this software in university labs. Inexpensive student versions of MATLAB are also available, making it one of the most accessible software platforms for DSP development. The environment is intuitive to work with and provides specialized toolboxes for signal and image processing, amongst many others. MATLAB also has the advantage that it is highly optimized for vectorized code, making it suitable for DSP algorithms.

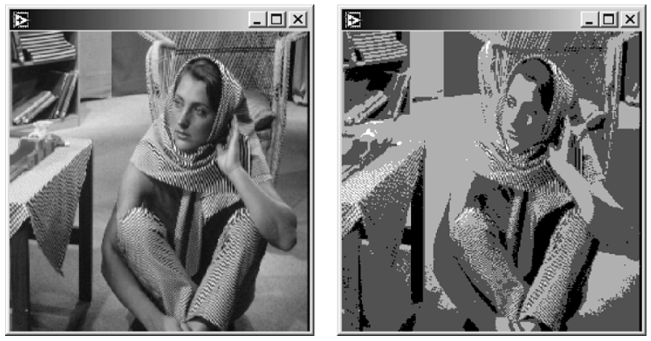
Similarly, LabVIEW is becoming a very popular software in most universities. One of the reasons for its popularity is that building front-end GUIs is extremely simple. Because the programming is graphical, it does not overwhelm programmers with syntactical details. LabVIEW also provides IMAQ Vision, a suite of image-processing routines (besides many others), which made it an attractive choice for building image-processing demos. Building educational tools with MATLAB and LabVIEW has the advantage that most end users are familiar with these tools and can build on the code or make minor modifications and customize it for their intended audience.

To obviate the need for a copy of MATLAB at the client end, MATLAB has developed the “MATLAB Server” solution [18]. Similarly, National Instruments has released their LabVIEW Player, which again makes unnecessary the need for a local copy of LabVIEW on the client end. The ability to program simply, impressive integrated graphical functions, availability for a wide variety of platforms, and extensibility to a Web-based education system, such as that used at the UT-Austin, made MATLAB and LabVIEW obvious choices for the developing platform.

The rest of this paper is organized as follows. The basic course structure of the two courses for which these tools have been

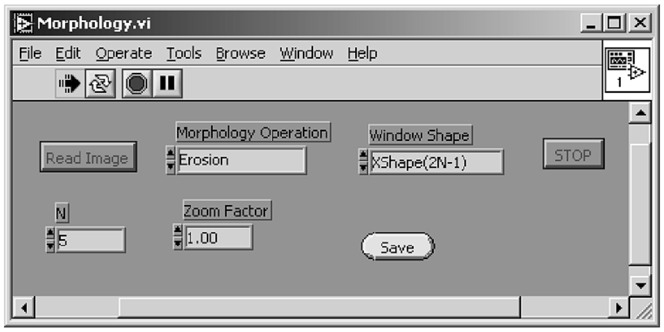


(a)

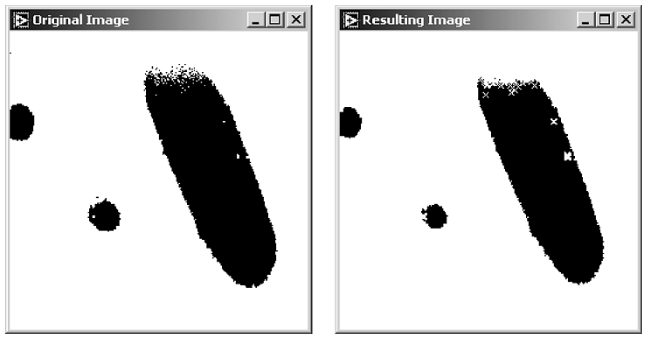


(b) (c)

Fig. 2. Effects of gray-level quantization. (a) Front panel. (b) "Barbara" at 8 b/pixel. (c) "Barbara" at 2 b/pixel.



(a)



(b) (c)

Fig. 3. Binary image morphology. (a) Front panel. (b) Original "cell." (c) "Cell" eroded.

developed for in-class audio-visual demonstrations is described in Section II. Section III is an overview of the DIP features of SIVA; Section IV is an overview of the DSP demos in SIVA; and Section V presents the conclusion.

II. Overview of Courses

Though intended for an electrical and computer engineering curriculum, the junior/senior course "Digital Image and Video Processing" (EE 371R) and the graduate course "Digital Signal Processing" (EE 381K-8) attracts students from various other majors, such as psychology, aerospace, geology, astronomy, and computer science. There is also a steady enrollment of professionals from the local industries at Austin, TX. The objective of both courses is "*to make signal processing accessible to an audience with heterogenous backgrounds by augmenting theory with numerous audio-visual examples*." To encourage a Web-based educational system, WebCT [19] is used as an instruction medium to make all course material and demonstrations available over the World Wide Web.

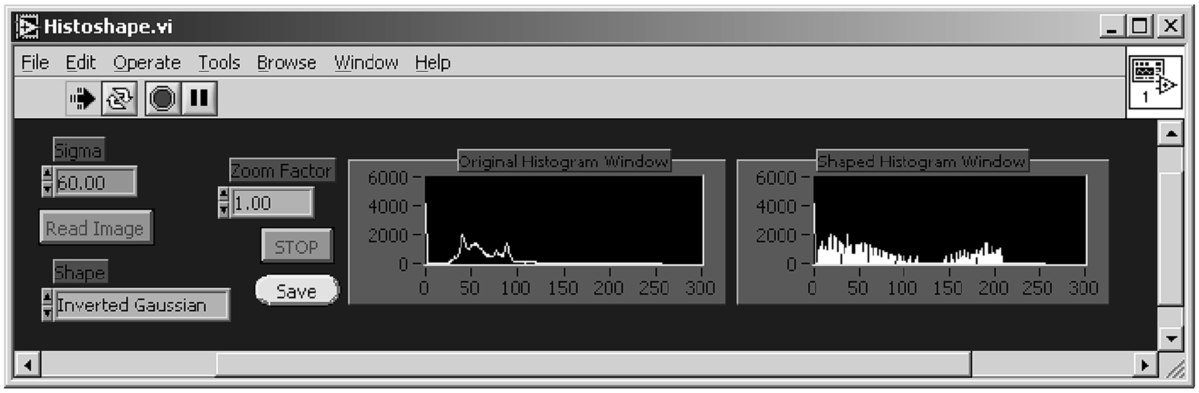
A. EE 371R: DIP and Video Processing

Introductory material covered in this junior/senior course on digital image and video processing includes binary image processing, image analysis, and image enhancement, while the more advanced material covers such topics as Hough transforms, edge detection, and video processing.

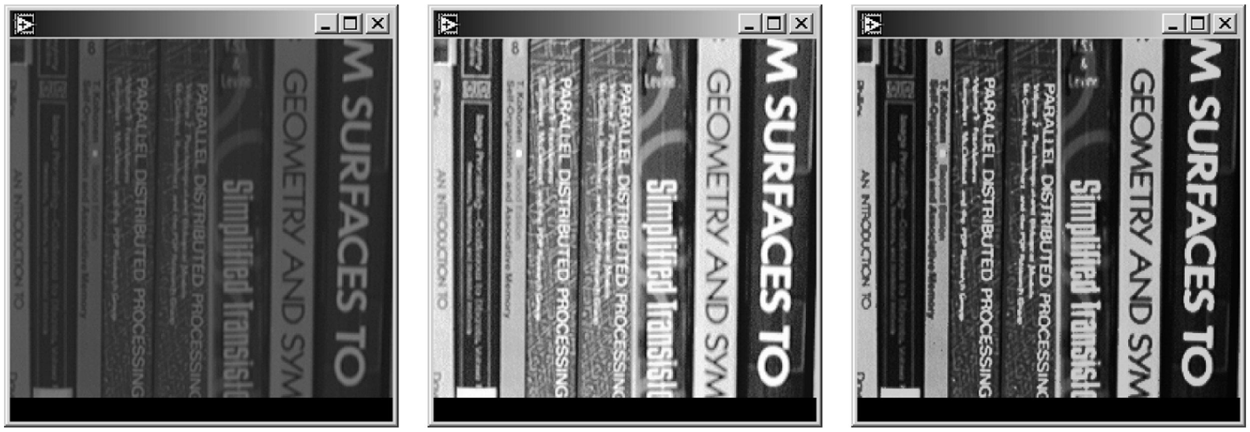
Along with course material [20] designed assiduously by the course instructor to keep the level of math accessible to the audience, a recent handbook [21] was introduced for the first time in fall 2000 to provide the audience with an invaluable reference and classroom text for the course. To assist visual interpretation of ideas discussed, in-class demonstrations using SIVA are used to complement lectures. Simple but intuitive MATLAB-based assignments are designed to reinforce the concepts without overwhelming students with programming intricacies. Students are also encouraged to investigate and apply the concepts learned in the course to their respective fields, and a monetary award is given as an added incentive to the best class project. Students are provided with digital camcorders (Canon ZR-10) and firewire cards to download digital video onto computers, video-editing software, Web cameras (Vista Imaging) and state-of-the-art videoconferencing equipment (Canon VC-C4) to develop their projects. The availability of equipment has greatly assisted in motivating students to build intellectually stimulating projects.

B. EE 381K-8: DSP

This graduate course in digital signal processing is offered every spring at UT-Austin. In contrast to EE 371R, this course is mathematically rigorous. The topics covered include signal representations, Fourier series expansions, z transforms, filter design, nonlinear filters, discrete-time random processes, quantization effects, multirate processing, and subband filter banks. Course notes [22], meticulously designed for the audience, are provided online on the course Web site. To emphasize the effects of DSP algorithms on real-world signals, the DSP section of SIVA was designed in harmony with the modules handled in



(a)



(b) (c) (d)

Fig. 4. Histogram shaping. (a) Front panel. (b) Original "books." (c) Histogram—flat. (d) Histogram—inverted Gaussian.

class. Many audio signals are used during the lectures to interpret theory. All demos are linked to the course notes for in-class demonstrations.

III. SIVA Image Processing Demonstration Gallery

LabVIEW [23] is a graphical programming language used as a powerful and flexible instrumentation and analysis software system in industry and academia. LabVIEW uses a graphical programming language, G, to create programs called virtual instruments (VIs) in a pictorial form, eliminating much of the syntactical details of other text-based programming languages, such as C and MATLAB. LabVIEW includes many tools for data acquisition, analysis, and display of results. LabVIEW is available for all the major platforms and is easily portable across platforms. LabVIEW has the ability to create stand-alone executable applications, which run at compiled execution speeds.

Another advantage of LabVIEW is that it includes built-in applications, such as the IMAQ Vision for image processing. IMAQ Vision includes more than 400 imaging functions and interactive imaging windows and utilities for displaying and building imaging systems, giving designers the opportunity to create examples for many important concepts in image processing and using them for educational purposes. An excellent introduction to LabVIEW is provided in [24] and [25]. Most technical information for the development of the VIs and IMAQ are provided in [26] and [27], respectively. Information specific to the VIs developed for this course can be obtained in [28].

A. LabVIEW Development Environment

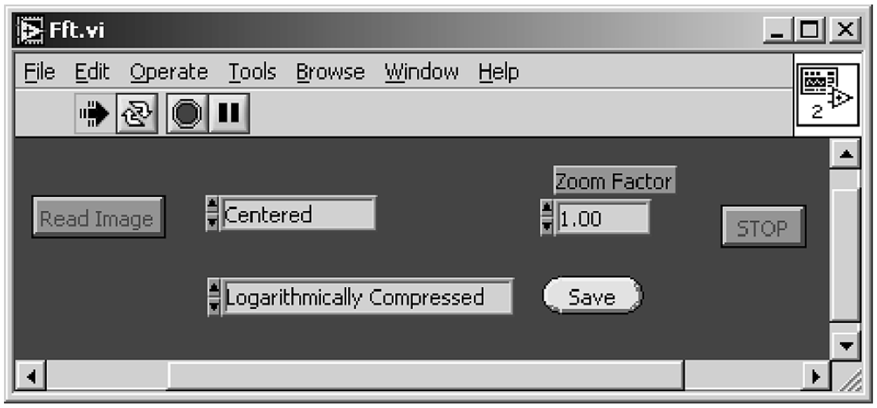
Each VI contains three parts.

- The front panel contains the user interface control inputs, such as knobs, sliders, and push buttons, and output indicators to produce such items as charts and graphs. Inputs can be fed into the system using the mouse or keyboard. A typical front panel, used to provide intuitive GUIs to vary the parameters of the algorithm, is shown in Fig. 1(a).
- The block diagram shown in Fig. 1(b) is the equivalent of a "source code" for the VI. The blocks are interconnected, using wires to indicate the data flow. Front-panel indicators pass data from the user to their corresponding terminals on the block diagram. The results of the operation are then passed back to the front-panel indicators.
- Sub-VIs are analogous to subroutines in conventional programming languages.

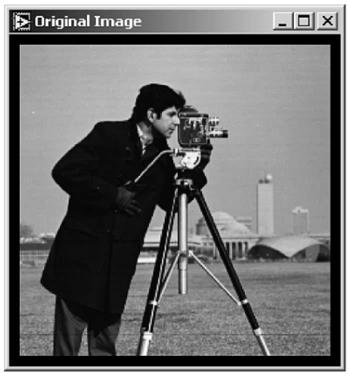
B. Examples of Demos

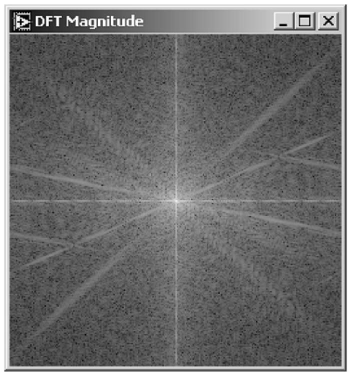
SIVA encompasses a wide range of VIs that can be used in conjunction with class lectures on image processing. Because of space constraints, only a small number of the LabVIEW VIs that were developed are described. The reader is invited to explore and download the other DIP demos in SIVA from the Web site mentioned in [29].

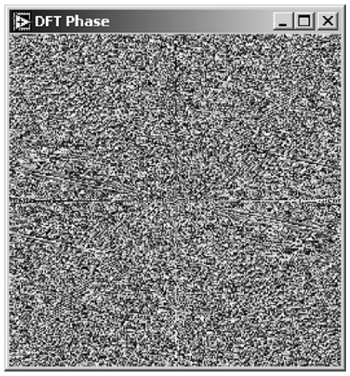
1) A–D Conversion: Sampling and quantization are fundamental operations in signal and image processing that transform


Fft.vi
File Edit Operate Tools Browse Window Help
Read Image
Centered
Zoom Factor
1.00
STOP
Logarithmically Compressed
Save

(a)







(b) (c) (d)

Fig. 5. Discrete Fourier transform. (a) Front panel. (b) Original "cameraman." (c) DFT magnitude. (d) DFT phase.

a continuous signal to a discrete one. Though the theory is mathematically cumbersome, the effects of sampling and quantization can be visualized effectively using the VIs shown in Fig. 2. False contouring effects resulting from gray-level quantization are visually obvious in Fig. 2(c). This VI reads in an 8-b/pixel image and creates an output whose bit depth (1, 2, 4, or 8 b/pixel) can be specified from the front panel. The sampling VI (not shown) can be used to visualize aliasing caused by sampling.

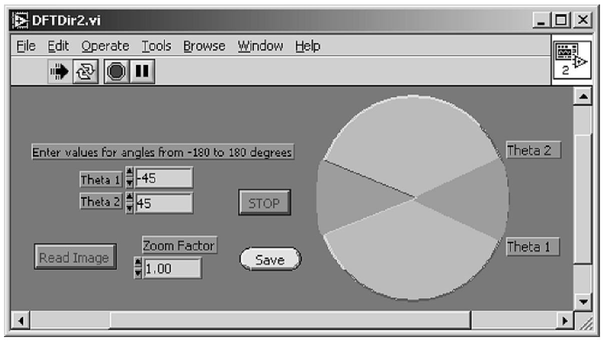
2) *Binary Image Processing:* Binary images have only two possible "gray levels" and are therefore represented using only 1 b/pixel. Besides simple VIs used for thresholding gray-scale images to binary, other VIs were developed to demonstrate the effects of binary morphology. Morphological operations are defined by moving a structuring element over the image and performing a logical operation on the pixels covered by the structuring element, resulting in another binary image. The morphology VI shown in Fig. 3 can be used to demonstrate the effects of various morphological operations on binary images, e.g., median, dilation, erosion, open, close, open–close, and close–open. The user also has the option of varying the size of the structuring element, which can be any of the following: row, column, square, cross, or X-shape.

3) *Histogram and Point Operations (Gray-Scale):* To demonstrate effects of elementary gray-scale image processing, VIs that perform linear (offset, scaling, and full-scale contrast stretch) and nonlinear (logarithmic range compression) point operations on images were developed. A more advanced VI, shown in Fig. 4, demonstrates the effects of histogram shaping. The histograms of the input image and the resulting image after the linear point operation are also displayed on the front panel to verify the results of the shaping (e.g., the histogram in Fig. 4(d) is inverse Gaussian-like).

4) *Image Analysis (Frequency Interpretations):* This consists of discrete Fourier transform (DFT) and directional DFTs.

a) DFT: Though a difficult concept for many students to comprehend, a lucid understanding of Fourier transforms is critical to the more advanced topics of image filtering and spectral theory. The study, therefore, begins by introducing the concept of digital frequency, using two-dimensional (2-D) digital sinusoidal gratings. The DFT demonstration VI (shown in Fig. 5) computes and displays the magnitude and the phase of the DFT for gray-level images. The DFT can be displayed with its low frequencies clustered together at the center of the image or distributed at the periphery. An option is provided to display the logarithmically compressed full-scale contrast-stretched version of the magnitude spectrum to reveal low-contrast values.

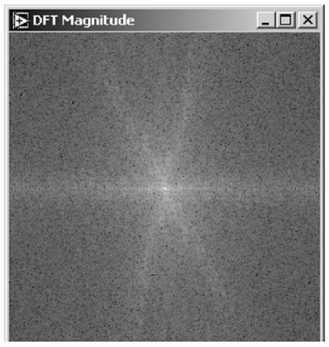
b) Directional DFTs: When the DFT of an image is brighter along a specific orientation, it implies that the image contains highly oriented components in that direction. Oriented binary images can be used to mask the DFT of these images, which, when operated on by the inverse DFT, produce images with only highly oriented frequencies remaining. To demonstrate the directionality of the DFT, the VI shown in Fig. 6 was implemented. As shown on the front panel in Fig. 6(a), the input parameters, Theta 1 and Theta 2, are used to control the angle of the wedge-like, zero-one mask. It is instructive to note that zeroing out some of the oriented components in the DFT


DFTDir2.vi
File Edit Operate Tools Browse Window Help
Enter values for angles from -180 to 180 degrees
Theta 1 -45
Theta 2 45
STOP
Read Image
Zoom Factor 1.00
Save
Theta 2
Theta 1

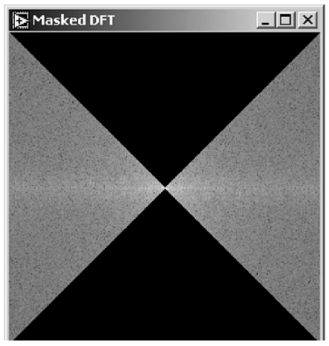
(a)

Original Image

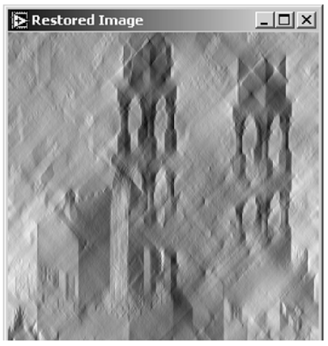
(b)



(c)



(d)



(e)

Fig. 6. Directional DFT. (a) Front panel. (b) Original "Escher." (c) DFT magnitude. (d) Masked DFT. (e) Inverse DFT after masking.

results in the disappearance of the conduits in the "Escher" image in Fig. 6(e).

5) Image Filtering: SIVA includes many demos to illustrate the use of linear and nonlinear filters for image enhancement and image restoration. The use of low-pass filters for noise smoothing and inverse and pseudoinverse filters for deconvolving images that have been blurred are examples of some demos for linear image enhancement in SIVA. SIVA also includes demos to illustrate the prowess of nonlinear filters over their linear counterparts. Fig. 7 demonstrates the result of filtering, with both a linear filter (average) and a nonlinear (median) filter, a noisy image corrupted with "salt & pepper noise."

6) Image Compression: The VI on block truncation coding (BTC) shown in Fig. 8 provides an introduction to lossy image compression. The user can select the number of bits used to represent both the mean of each block in BTC (in $B1$) and the block variance (in $B2$). The compression ratio is computed and displayed on the front panel in the CR indicator in Fig. 8(a).

Other advanced VIs developed include many linear and nonlinear filtering for image enhancement, other lossy and lossless image compression schemes, a large number of edge detectors for image feature analysis, and Hough transforms. The elementary ones have been presented here for the purpose of illustration.

IV. SIVA Signal Processing Demonstration Gallery

MATLAB [30] is a high-performance language for technical computing. It integrates computation, visualization, and programming into an easy-to-use environment [31]. The basic data element in MATLAB is an array. Many matrix-based functions, such as matrix multiplication and array dot product, can be executed in a fraction of the time it would take to write a similar program in a scalar, noninteractive language such as C or Fortran. MATLAB features a family of application-specific solutions called toolboxes for such applications as signal processing, neural networks, and wavelets. These toolboxes are a library of functions written as M files.

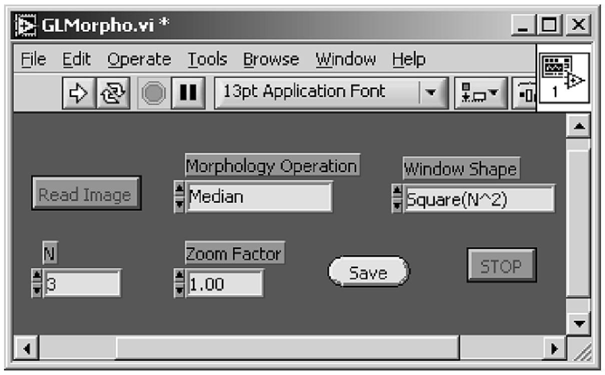
The signal-processing toolbox [32], for example, includes an interactive environment for analyzing and manipulating signals and designing filters. MATLAB has a number of easy-to-use plotting and graphical functions, which make MATLAB an attractive choice for developing attractive visualization applications. The vectorized nature of MATLAB and the abundant collection of functions and visualization options make it a good choice for visualizing DSP concepts. MATLAB provides a powerful GUI development tool critical to the creation of educational tools for classroom instruction needs. Current versions of MATLAB also provide powerful user-friendly debugging tools.

A. GUI Development Using MATLAB

MATLAB provides the GUI Design Environment (GUIDE) to develop impressive GUIs quickly using drag-and-drop objects, such as buttons, sliders, and pop-down menus [33]. Fig. 9 shows a typical GUI development environment. The GUIDE control panel provides the drag-and-drop objects, the properties of which are controlled by the GUIDE property editor. An M file performing a particular task for each object in the GUI is written separately, using many of the in-built MATLAB functions. The GUIDE callback editor manages the actions associated with the selection of a particular object (e.g., clicking a button) by linking an object to its respective M file (as shown by the sequence of arrows in Fig. 9). Once the GUI is developed, one can use the mouse and on-screen options to visualize, hear, and manipulate signals (audio and images). A number of signals can be analyzed, e.g., in audio, male and female voices, music, and standard waves (e.g., sine, chirp, square, and triangle). A few demos also use images as inputs to illustrate multidimensional DSP algorithms. For classroom instruction, the demos are hyperlinked from an HTML document.

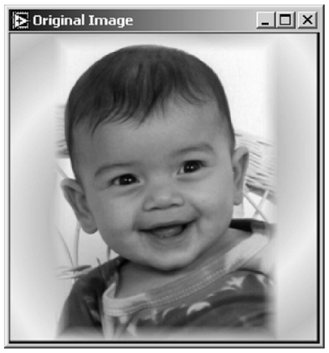
B. Examples of Demos

Using the powerful graphics and simple functionality of MATLAB, a number of DSP demonstrations that can be used for a classroom teaching environment have been developed to demonstrate visually the fundamental DSP concepts using

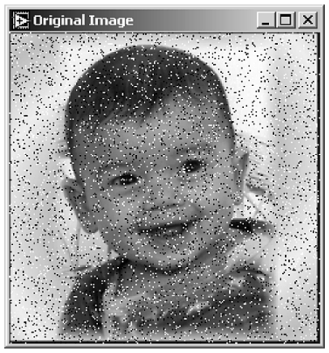


Morphology Operation: Median
Window Shape: Square(N^2)
N: 3
Zoom Factor: 1.00

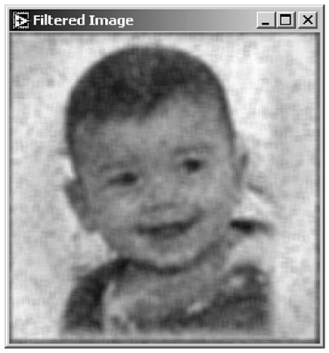
(a)



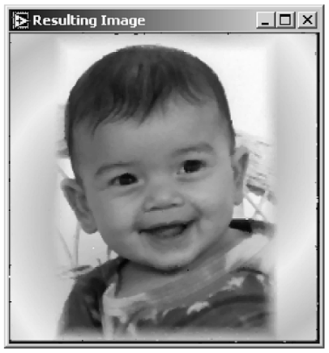
(b)



(c)



(d)



(e)

Fig. 7. Nonlinear filtering. (a) Front panel. (b) Original "Dhivya." (c) Salt & pepper noise. (d) Average filtering. (e) Median filtering.

mainly audio and a few synthetic signals. Currently, all demos run on the Windows platform but can be easily transported onto other platforms as well. In the following subsections, an overview of a few modules that were developed is presented. The reader is invited to experiment with the others by downloading them from the Web site mentioned in [29].

1) Signal Representations: In this introductory module, the focus is on the representations of an input signal in different domains: time, frequency, and time–frequency domains. The user can select an audio sample to be viewed in any of these domains, change its sampling frequency and the number of bits per sample, and hear the original and the modified signal. Selecting a chirp signal and sampling it appropriately can be used effectively to visualize and hear aliasing, as shown in Fig. 10. Other simple demos have been developed to illustrate the difference between continuous, discrete time, and digital signals.

2) Fourier Series: One of the mathematically intriguing techniques of signal decomposition is that of the Fourier series. The concept of many sinusoids perfectly aligning to interfere constructively and destructively to create the time-domain signal is illustrated powerfully in the demo shown in Fig. 11, using a saw-tooth waveform as an example. The dotted line shows the reconstruction using six Fourier series coefficients. The slider "coefficients used" in the GUI enables the user to add more sinusoids and demonstrate the construction using the sinusoidal basis.

3) z Transform: The z transform demo (shown in Fig. 12) demonstrates the powerful graphics capability of MATLAB. The effect of zeros and poles on the impulse and frequency response of a system can be demonstrated. The user can place poles and zeros on the z plane using the click of a mouse. The three-dimensional 3-D effect of the poles and zeros [Fig. 12(b)], along with other options, such as frequency [Fig. 12(c)] and impulse responses [Fig. 12(d)], can be visualized.

4) Gibb's Phenomenon: When given the frequency-domain specifications, windowing the inverse Fourier transform is a common technique used to design filters in DSP. The GUI in Fig. 13 depicts the ripple effects of Gibb's phenomenon, caused by truncating.

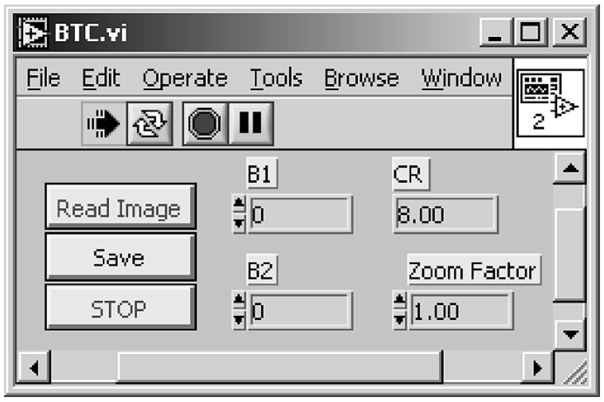
5) Filter Design: Filter design demos for finite-impulse response (FIR) filters and infinite-impulse response (IIR) filters can be used to illustrate the effects of filtering real-world signals. The demos have options to view the pole-zero plots of filters, the magnitude and phase response of the designed filter [Fig. 14(b)], and to select audio signals and see and hear the effects of filtering them. A frequency sampling technique for FIR filter design is shown in Fig. 14. The effect of filtering a linear chirp signal using the designed filter is illustrated in Fig. 14(c). Also included in the GUI are options to control the filter type and cutoff frequencies.

6) Decimation: The difference between subsampling and decimation is described intuitively in Fig. 15. The aliasing introduced because of subsampling is visually obvious in Fig. 15(c), where the spectrum replicas outside the $[-\pi, \ \pi]$ digital frequency band have entered the $[-\pi, \ \pi]$ band (note the reversed direction of the triangular bands). Selecting the low-pass filter option in the GUI filters the original signal with an appropriate low-pass filter before subsampling in order to prevent any aliasing, evident in Fig. 15(d). The aliasing effects are audible when the subsampled signals are played.

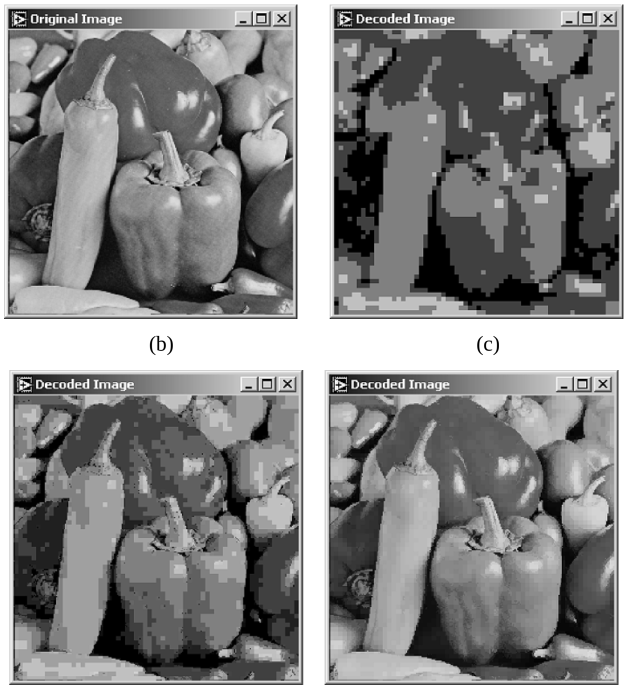
Besides the elementary examples discussed previously, a number of other demos explain concepts in Fourier series, discrete Fourier transforms, filter design, multirate DSP, short-time Fourier transforms, etc. A median filter demo (for audio and images) was also developed to illustrate the superior performance of nonlinear filters over linear systems.

V. Conclusion

The importance of stressing practical applications in a DSP course has been prevalent for a long time now. Making DSP accessible to an ever-growing nonexpert audience is highly desirable. In this paper, a library of demonstrations built, and successfully used at UT-Austin, intuitively introduce the concepts of



(a)



(b) (c) (d) (e)

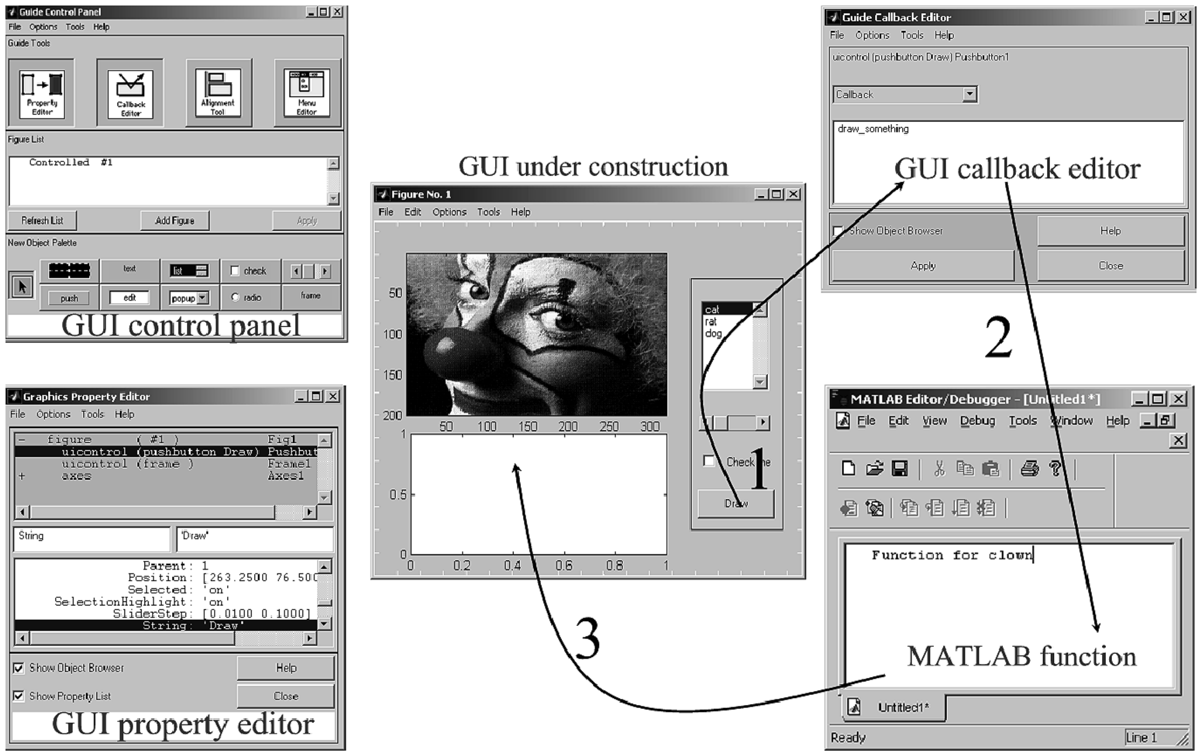
Fig. 8. Block truncation coding. (a) Front panel. (b) Original "Peppers." (c) 2 b for mean; 0 b for variance. (d) 3 b for mean; 4 b for variance. (e) 6 b for mean; 5 b for variance.

digital signal processing and digital image and video processing to a disparate audience consisting of graduate and undergraduate students from a variety of backgrounds. These classes are amazingly popular, typically attracting more than 80 students per class in recent semesters. The inertia to develop such demos is understandable since a high investment of time and effort is required. Therefore, the SIVA visualization package is provided free on the World Wide Web [29] for pedagogic purposes. More than 40 users from various countries are already using SIVA for their projects and as a teaching tool for signal and image processing courses. It is the hope of the authors that this paper will succeed in attracting the attention of many more DSP instructors to the availability of SIVA.

Encouraged by the success of these visualizations in tandem with class lectures, work is in progress to build another visualization in the area of digital video processing. The video processing demos in SIVA will use the LabVIEW environment and IMAQ Vision to demonstrate such key concepts as motion estimation and compensation, motion-compensated filtering, video compression and reconstruction, the effects of noise, and preprocessing. The results from these demos can be compared both objectively (e.g., peak signal to noise ratio) and subjectively for assessment of video reconstruction quality. Within many demos, users will have the option to vary computation methods for additional comparison. Fig. 16, for example, illustrates the estimation of optical flow between two scenes of a video.

To illustrate a practical optical flow estimating technique, the motion-estimation demo shown in Fig. 17 allows the use of different search methods, such as the three-step search, the cross search or brute force, and different match methods, such as the minimum mean-square error, maximum absolute difference, maximum matching-pixel count, or maximum correlation [34]. In addition, many demos will operate on live video acquired in-class into LabVIEW from a digital camcorder and a laptop IEEE 1394/firewire port. These demos will add flexibility and intrigue to the learning process and help emphasize both the simplicities and the intricacies of video processing, which may be overlooked when using a predefined set of videos that have controlled acquisition environments. The net result will be a package of simple, qualitative, yet practical video processing learning tools.

The LabVIEW demos will be converted to LabVIEW player format in the near future, making unnecessary the need for local copies of LabVIEW on the client end. It is also proposed to expand such tools to encompass signal-processing aspects of wavelets in the coming months. The availability of the free MATLAB-based Wavelab package [35] is a strong motivation for building these demos using MATLAB. The authors are optimistic that these libraries will be instrumental in adding value to the way digital video processing and wavelets will be taught at UT-Austin and, hopefully, at many other universities around the world.

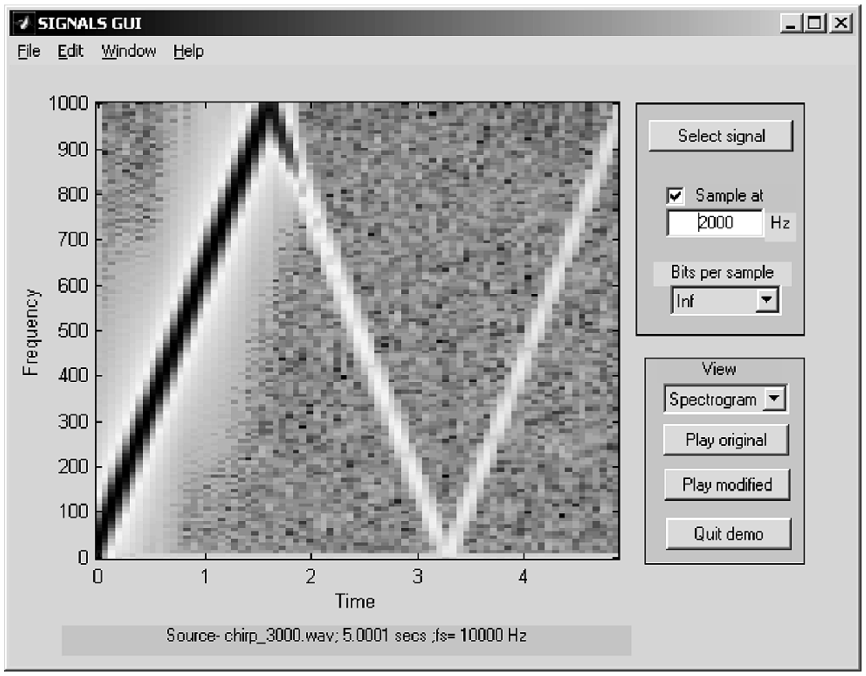


```
uicontrol (pushbutton Draw) Pushbutton1
Callback
draw_something
```

```
-  figure        ( #1 )                   Fig1
     uicontrol (pushbutton Draw)  Pushbut
     uicontrol (frame )           Frame1
+    axes                         Axes1
String    'Draw'
              Parent: 1
            Position: [263.2500 76.500
            Selected: 'on'
  SelectionHighlight: 'on'
          SliderStep: [0.0100 0.1000]
              String: 'Draw'
```

```
Function for clown
```

Fig. 9. Typical GUI development environment in MATLAB.



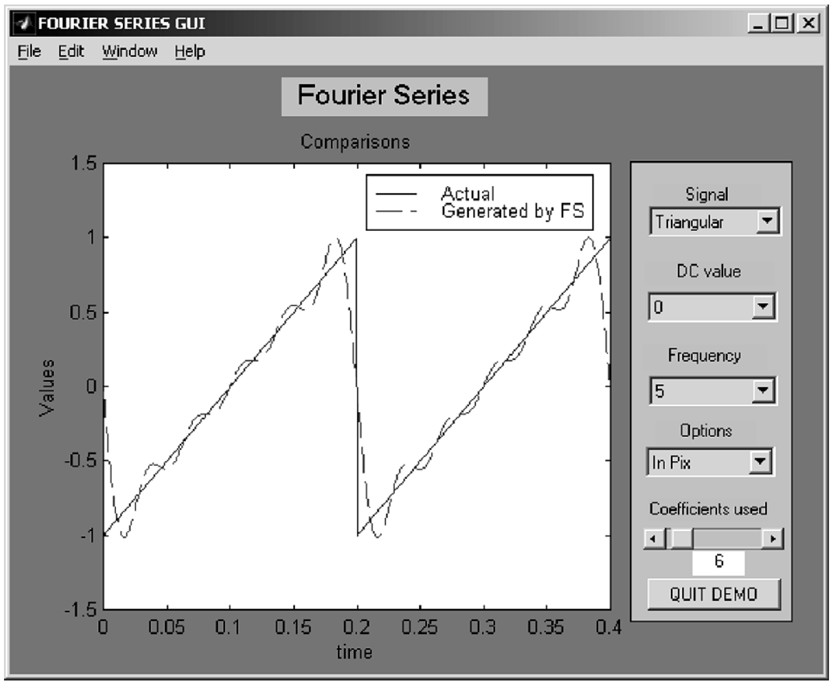
Sample at [✓] 2000 Hz

Bits per sample: Inf

View: Spectrogram

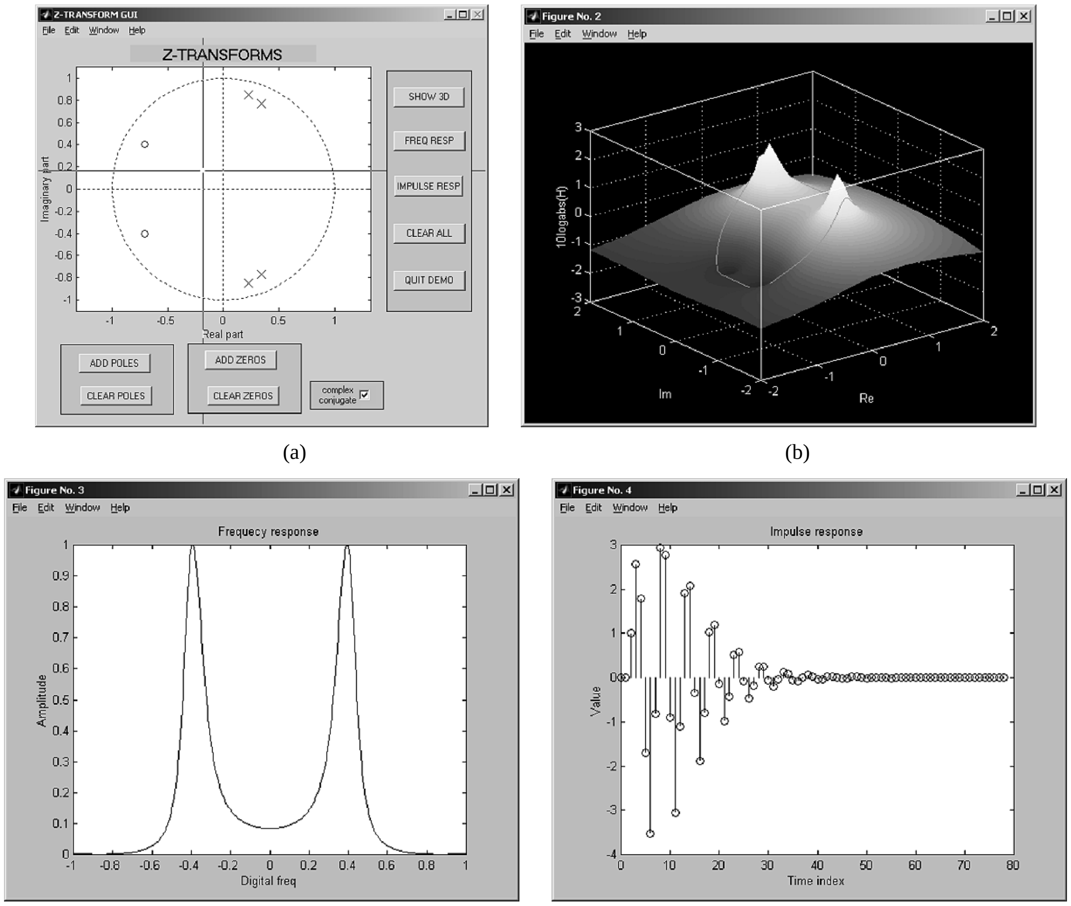
Source- chirp_3000.wav; 5.0001 secs ;fs= 10000 Hz

Fig. 10. Signal representations.



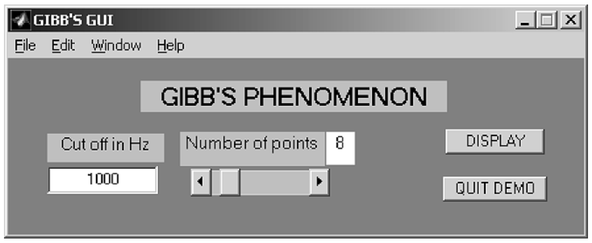
Signal: Triangular
DC value: 0
Frequency: 5
Options: In Pix
Coefficients used: 6

Fig. 11. Fourier series.



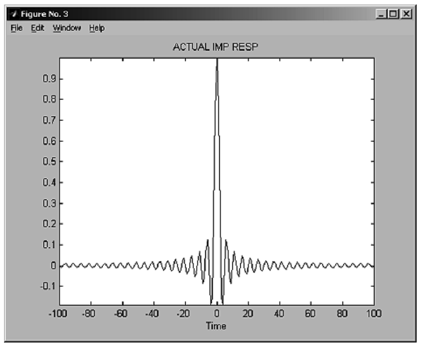
(a) (b) (c) (d)

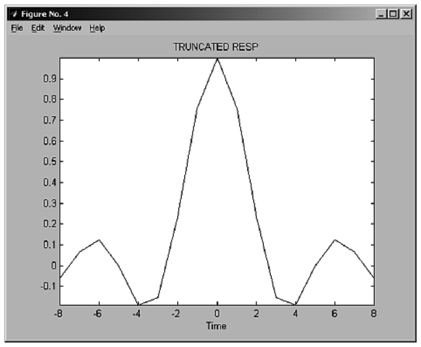
Fig. 12. z transform. (a) z transform GUI. (b) 3–D pole-zero response. (c) Magnitude response. (d) Impulse response.

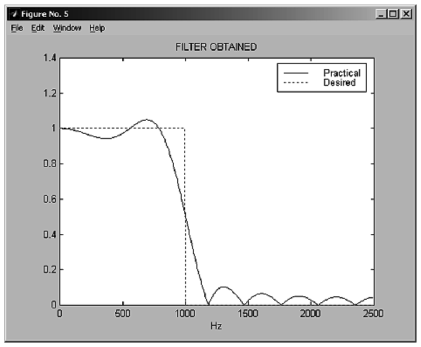


Cut off in Hz: 1000
Number of points: 8

(a)

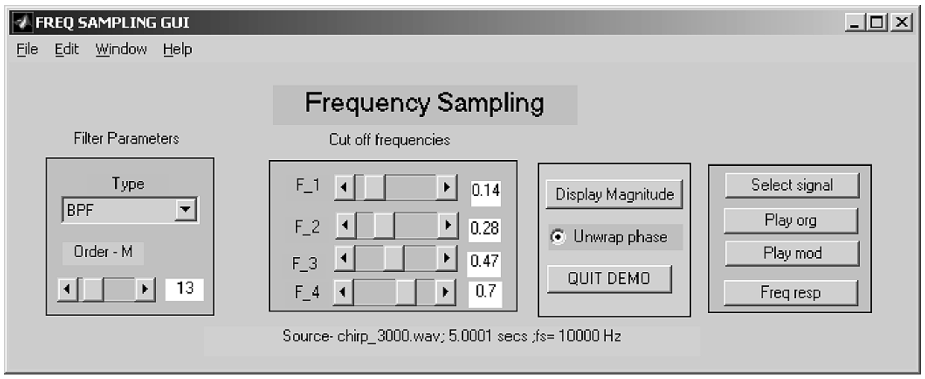






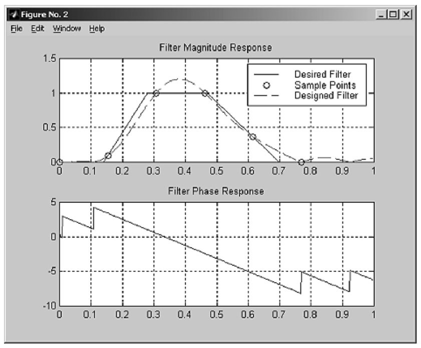
(b) (c) (d)

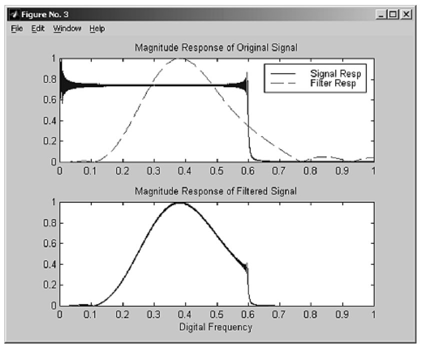
Fig. 13. Gibb's phenomenon. (a) GUI for Gibb's phenomenon. (b) Impulse response of desired low-pass filter. (c) Truncated impulse response. (d) Desired and designed responses.



Type: BPF
Order - M: 13
F_1: 0.14
F_2: 0.28
F_3: 0.47
F_4: 0.7
Unwrap phase: ✓
Source- chirp_3000.wav; 5.0001 secs ;fs= 10000 Hz

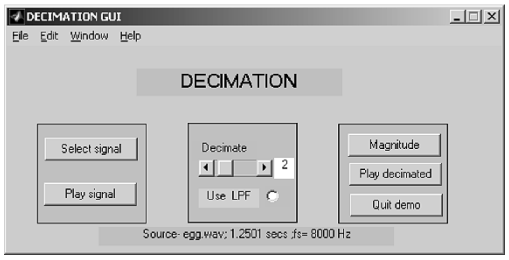
(a)



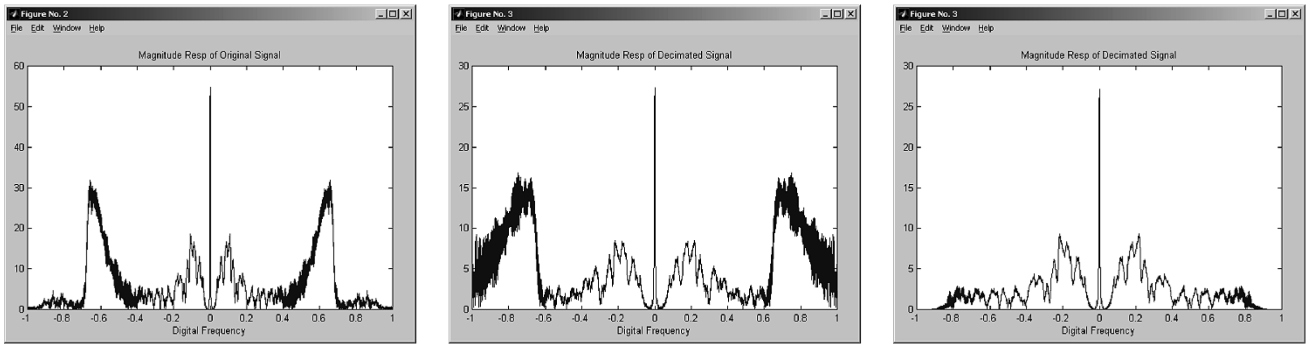


(b) (c)

Fig. 14. FIR filter design by frequency sampling. (a) Frequency sampling GUI. (b) Frequency response of desired and designed filters. (c) Results of filtering of chirp signal.

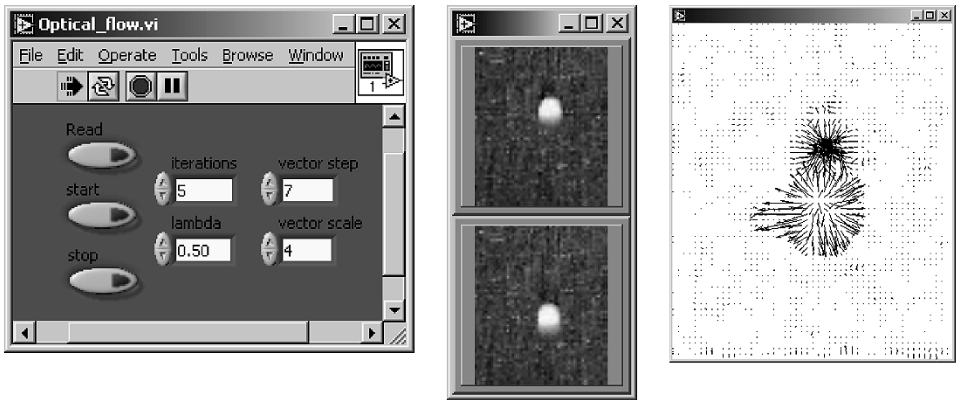


(a)



(b) (c) (d)

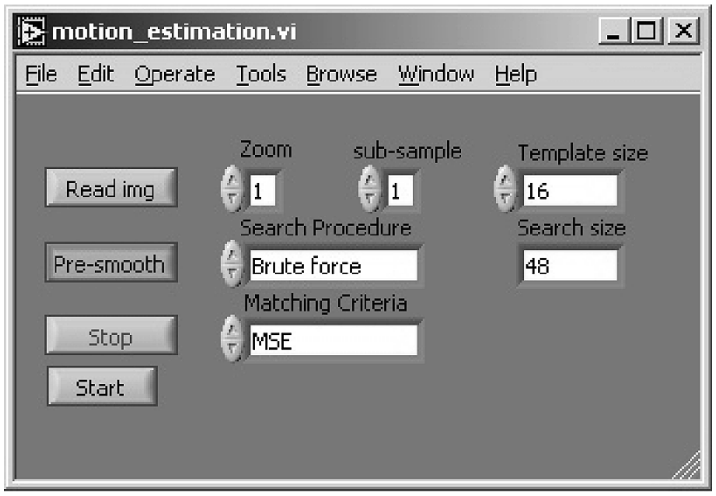
Fig. 15. Decimation. (a) Decimation GUI. (b) Frequency spectrum of audio signal. (c) Spectrum after subsampling by factor of 2. (d) Spectrum after decimation by factor of 2.



iterations 5
vector step 7
lambda 0.50
vector scale 4

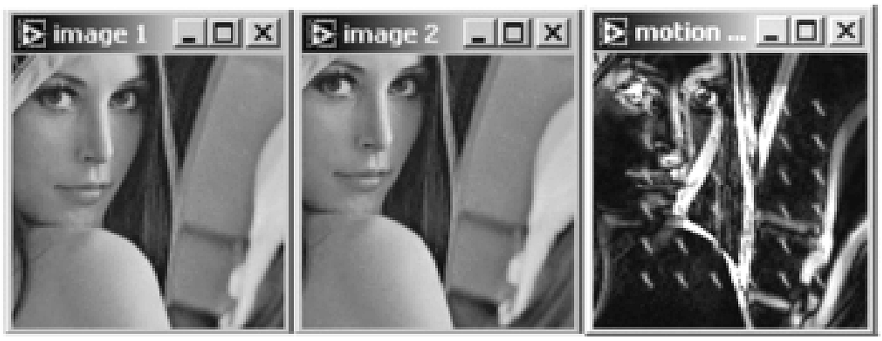
(a) (b) (c)

Fig. 16. Optical flow. (a) Front panel. (b) Successive video frames. (c) Estimated optical flow.



Zoom: 1
sub-sample: 1
Template size: 16
Search Procedure: Brute force
Search size: 48
Matching Criteria: MSE

(a)



(b) (c) (d)

Fig. 17. Block-based motion estimation. (a) Front panel. (b) Original "Lena." (c) "Lena" shifted northwest by a few pixels. (d) Arrows indicating estimated motion for each block.

ACKNOWLEDGMENT

This courseware development work was largely supported by grants from the Texas Telecommunication Engineering Consortium (TxTEC) and by software assistance from National Instruments, Inc.

REFERENCES

[1] U. Rajashekar and A. C. Bovik. Interactive DSP education using MATLAB demo. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/
[2] G. C. Panayi, U. Rajashekar, and A. C. Bovik. Image processing for everyone. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/
[3] The Infinity Project [Online]. Available: http://www.infinity-project.org/home.html
[4] T. Barnwell and B. Evans. DSP as a first course. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/program.html#dspcourse
[5] G. C. Orsak and D. M. Etter, "Collaborative DSP education using the Internet and MATLAB," *IEEE Signal Processing Mag.*, vol. 12, pp. 23–32, Nov. 1995.
[6] R. Radke and S. Kulkarni. An integrated MATLAB suite for introductory DSP education. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/
[7] M. W. J. Williams and G. C. Orsak. Peruna and pony express: Two MATLAB-based educational software packages for signal processing and communications. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/
[8] J. Rosenthal and J. H. McClellan, "Animations and GUIs for introductory engineering courses," in *Proc. Int. Conf. on Electrical Engineering*, Aug. 2001, pp. 6E411–6E416.
[9] National Instruments LabVIEW Player VI Gallery [Online]. Available: http://zone.ni.com/devzone/explprog.nsf/webLabVIEWenabled
[10] Y. Cheneval, L. Balmelli, P. Prandoni, J. Kovacevic, and M. Vetterli, "Interactive DSP education using Java," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, 1998, pp. 1905–1908.
[11] A. Spanias, A. Constantinou, J. Foutz, and F. Bizuneh. An online signal processing laboratory. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/program.html
[12] J. Shaffer, J. Hamaker, and J. Picone, "Visualization of signal processing concepts," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, 1998, pp. 1853–1856.
[13] B. L. Evans, L. J. Karam, K. A. West, and J. H. McClellan, "Learning signals and systems with mathematica," *IEEE Trans. Educ.*, vol. 36, pp. 72–78, Feb. 1993.
[14] J. Gosling and H. McGilton. (1996) The Java Language Environment: A White Paper. [Online]. Available: http://java.sun.com/docs/white/langenv/
[15] R. Martti and K. Matti, "An interactive DSP tutorial on the web," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, 1997, pp. 2253–2256.
[16] C. Ulmer. "How to use PEZ". [Online]. Available: http://www.ece.ubc.ca/verb1~1edc/466/pezdoc/Demos/use.html
[17] C. Vignat and S. Vallèry. ZeroPole: A new tool for teaching filter theory and design. presented at 1st Signal Processing Education Workshop. [Online]. Available: http://spib.ece.rice.edu/DSP2000/
[18] MATLAB Server, Mathworks, Inc.. [Online]. Available: http://www.mathworks.com/products/webserver/
[19] WebCT Homepage [Online]. Available: http://www.webct.com/

[20] A. C. Bovik. EE 371R Digital Image and Video Processing Course Notes. [Online]. Available: http://live.ece.utexas.edu/class/ee371r

[21] ——, *Handbook of Image and Video Processing*, 1st ed, ser. Communications, Networking, and Multimedia Series. New York: Academic , Mar. 2000.

[22] ——, EE 381-K Digital Signal Processing Course Notes. [Online]. Available: http://live.ece.utexas.edu/class/ee381k

[23] LABVIEW homepage [Online]. Available: http://www.ni.com/labview

[24] L. K. Wells and J. Travis, *LabVIEW for Everyone*, 1st ed. Upper Saddle River, NJ : Prentice-Hall, 1997.

[25] R. H. Bishop, *LabVIEW Student Edition 6i*, 1st ed. Upper Saddle River, NJ: Prentice-Hall, 2001.

[26] *Labview User Manual*, 1996.

[27] *BridgeVIEW and LabVIEW IMAQ Vision for G reference manual*, 1996.

[28] G. C. Panayi, "Implementation of Digital Image Processing Functions Using LabVIEW," M.S. thesis, Dept. of Electrical and Computer Engineering, Univ. of Texas at Austin , May 1999.

[29] The SIVA Demonstration Gallery [Online]. Available: http://live.ece.utexas.edu/class/siva

[30] MATLAB homepage [Online]. Available: http://www.mathworks.com

[31] MathWorks Inc. (1999, May) Getting Started with MATLAB—Version 5. [Online] http://www.mathworks.com/access/helpdesk/help/techdoc/learn_matlab/gs_colle.shtml

[32] The Mathworks Inc.. (1996, Dec.) MATLAB Signal Processing Toolbox. [Online] http://www.mathworks.com/access/helpdesk/help/toolbox/signal/signal.shtml

[33] The Mathworks Inc.. (1997, June) Building GUIs With MATLAB-Version 5. [Online] http://www.mathworks.com/access/helpdesk/help/techdoc/creating_guis/creating_guis.shtml

[34] M. A. Tekalp, *Digital Video Processing*, 1st ed. Upper Saddle River, NJ : Prentice-Hall , 1995.

[35] Wavelab 802 [Online]. Available: http://www-stat.stanford.edu/verb1~1wavelab/

Umesh Rajashekar (S'97) received the B.E. degree in electronics and communication engineering from the Karnataka Regional Engineering College, Surathkal, India, in July 1998 and the M.S. degree from the University of Texas at Austin (UT-Austin) in August 2000. He is currently pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering at UT-Austin.

Since spring 2000, he has been a Research Assistant in the Laboratory for Image and Video Engineering at UT-Austin, where he is investigating image statistics at point of gaze. His interests also include developing didactic tools for education.

Mr. Rajashekar was awarded the 2000–2001 Texas Telecommunications Engineering Consortium Graduate Fellowship from the University of Texas.

George C. Panayi was born in Larnaca, Cyprus, on November 13, 1972. After high school, he received the Higher National Diploma in electrical engineering from the Higher Technical Institute in Nicosia, Cyprus. He received the B.S. degree in computer science with honors and the M.S. degree in engineering from the University of Texas at Austin (UT-Austin) in 1997 and 1999, respectively.

In July 1993, he joined the Cyprus army. In the army, he had four months of officer's training in Greece and then served as a Second-Lieutenant for the supply and transportation division. Since July 1997, he has been affiliated with the Laboratory for Vision Systems at UT-Austin.

Frank P. Baumgartner is currently pursuing the B.S.E.E. degree at the University of Texas in Austin (UT-Austin), where he is focusing on image and video processing.

He is currently a Research Assistant in the Laboratory for Vision Systems (LVS) at UT-Austin. There, he develops video processing tools for use in a learning environment. His interests include the human psychovisual system, machine vision, and development of effective learning tools.

Alan C. Bovik (S'80–M'80–SM'89–F'96) received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1980, 1982, and 1984, respectively.

During spring 1992, he held a visiting position in the Division of Applied Sciences, Harvard University, Cambridge, MA. He is currently the Robert Parker Centennial Endowed Professor in the Department of Electrical and Computer Engineering and the Director of the Laboratory for Image and Video Engineering (LIVE) in the Center for Perceptual Systems, the University of Texas at Austin (UT-Austin). He is also a frequent consultant to legal, industrial, and academic institutions. He has published more than 300 technical articles in the research areas of digital video, image processing, and computational aspects of biological visual perception. He holds two U.S. patents. He is also the editor/author of the *Handbook of Image and Video Processing* (New York: Academic, 2000). He has been a Member of the Editorial Board for numerous professional society publications, including*Pattern Recognition* (1988–present), *The Journal of Visual Communication and Image Representation* (1992–1995), *Graphical Models and Image Processing* (1995–1998), *Pattern Analysis and Applications* (1997–1998), and *Real-Time Imaging* (2000–present).

Dr. Bovik is a registered Professional Engineer in the State of Texas. He has been a two-time Honorable Mention winner of the international Pattern Recognition Society Award for Outstanding Contribution in 1988 and 1993. He received the Institute of Electrical and Electronics Engineering (IEEE) Signal Processing Society Meritorious Service Award in 1998, was named Distinguished Lecturer of the IEEE Signal Processing Society, and received the IEEE Third Millennium Medal and the University of Texas Engineering Foundation Halliburton Award, all in 2000. In addition, he has served on the Board of Governors of the IEEE Signal Processing Society from 1996 to 1998 and as the Founding General Chairman of the First IEEE International Conference on Image Processing, Austin, TX, in November 1994. He has served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL Processing (1989–1993), Steering Committee Member of the IEEE TRANSACTIONS ON IMAGE PROCESSING (1991–1995), Associate Editor of the IEEE SIGNAL PROCESSING LETTERS (1993–1995), Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING (1996–2001), and Editorial Board Member of the PROCEEDINGS OF THE IEEE (1998–present).