Image Processing and Image Filters in Java

by John F. McGowan, Ph.D.
Desktop Video Expert Center
NASA Ames Research Center
Introduction

- Java is too slow for production image processing, but may offer advantages for prototyping and demonstration of image processing algorithms.

- Algorithm developers and scientists want a tool that allows them to concentrate on their work (math, science), not on the tool (C, Mathematica, Java).
Image Filters in Java


- java.awt.image Package
  - ImageFilter and RGBImage Filter classes
  - user derived filter classes

- In JDK 1.0
- Extended slightly in JDK 1.1
- Extended heavily in JDK 1.2 Beta
Image Filters and NASA

- Astronomical Images
  - Planets
  - Sun and Stars
- Satellite Images
- Aerial Photographs
- Images from Simulations
  - Computational Fluid Dynamics
  - Solar Models
Image Filters and NASA

- Edge Detection and Enhancement
- Image Segmentation
- Image Enhancement
- Image Analysis
  - e.g. Fourier Transforms
- Computer Vision
- Special Effects
Outline of Talk

- Desktop Video Expert Center
- Algorithm Development
- Image Filters and Image Processing in Java
  - Java Demo
  - How Java Images Work
- Conclusions
Desktop Video Expert Center

- Advanced Applications (R and D)
- Installation and Support of Videoconferencing Systems
- NASA Space Shuttle MBONE (Multicast Backbone) Broadcasts over Internet
- Evaluations of COTS Desktop Video Products
- http://zeus.arc.nasa.gov/
DVEC Advanced Applications

- Research and Evaluate Future Desktop Video and Networked Video Technologies.
- Develop New Technologies as Appropriate
- Software and Algorithm Development
- http://zeus.arc.nasa.gov/adv_apps.html
Some Current Projects

- **DCTune: Perceptual Optimization of JPEG Images** (with Vision Science and Technology Group)
- **Studies of Wavelet, Fractal, and Other Leading Edge Video Coding Technologies**
  - Smooth full-motion video over Internet?
  - Smooth full-motion video over telephone?
DCTune

- Patented NASA algorithm developed by Andrew B. Watson, Al Ahumada, and others with the Vision Science and Technology Group.
- Mathematica Prototype
- Converted to compiled C language binary executable
- http://vision.arc.nasa.gov/dctune1.1.html
Who am I?

- Converted DCTune from Mathematica Prototype to portable ANSI C language version.
- Developed commercial MPEG-1 and MPEG-2 audio and video playback (decoder) software written in C for PC/Windows, Power Macintosh, and Unix platforms.
Who am I?

- Image, Video, and Audio Compression Algorithms
- Ph.D. in Physics, University of Illinois at Urbana-Champaign
  - Maximum Likelihood Fitting Methods
  - Monte Carlo Simulations
  - Pattern Recognition
Developing Algorithms

- Cumbersome process
- Symbolic Manipulation Programs such as Mathematica, MATLAB, and MAPLE
- C/C++ Prototypes
- Pros and Cons to Both Methods
- Looking for Better Tools to Develop, Prototype and Demonstrate Algorithms
Traditional Algorithm Development

- Proof of Concept in Software on Supercomputer, Mainframe or Unix Workstation
- Make Presentation or Publish a Paper
- Present Simple Demo
  - Grayscale Images
  - Very Slow
- Fund Further Development
Traditional Algorithm Development

- Design and Fabricate VLSI Chip
- Implementing the Algorithm
- Develop Better Software and **wait** for CPU’s to get Fast Enough
  - Optimize the Software for Speed
  - Add Color Support
  - Support Arbitrary Picture Dimensions
The New World

- CPU Speeds Heading Above 500 MHz
- Algorithms Can Run on Desktop PC
- Very Fast Transition from Prototype to Product
- Argues for developing in C/C++ in PC/Windows environment.
- What can Java do?
Mathematica Advantages

- Symbolic manipulation
- Interpreted, Interactive
- High Level Mathematics
  - matrix multiplication \( m = a \cdot b \)
  - special math functions
  - much more
- Built-In Graphical Rendering
  - Show[graphics]
Mathematica Disadvantages

- Slow
- Proprietary (Wolfram Research)
- Expensive ($2000)
- Requires Much Memory (300 MB)
- Arcane Syntax
  - e.g., MapThread[f, {{a,b,c},{ap,bp,cp}}]
- Learning Curve for C or other procedural language programmers
Other Symbolic Packages

- MATLAB
- MAPLE
- REDUCE

Probably similar advantages and disadvantages to Mathematica.
Many Symbolic Language Prototypes

- Hard to reach mass audience.
  - For example, must own Mathematica
- Usually must convert to C, assembler, or an ASIC for a product.
- Usually must embed C version in platform OS or API
  - Video for Windows codec
  - QuickTime Component
C/C++ Advantages

- Fast/Compiled
- Executables are Free
- Unix GNU Compiler is Free
- Mac and PC Commercial Compilers
- De Facto Standard
  - Lingua Franca of Programmers
  - ANSI Standard
C/C++ Disadvantages

- No Built-In Graphics
  - Microsoft Windows (PC)
  - Mac Toolbox (MacOS)
  - X Windows (Unix)
  - Many Graphic File Formats (e.g. PPM)
- No High Level Mathematics
- No symbolic manipulation
- Porting Problems
ANSI C Porting Problems

- **Byte-Order Dependent Code**
- **Gaps in Standard** such as definition of bit shift by a negative number: `a >> -1`
- **Bugs in Compilers**. Video Compression Algorithms Stress Compilers.
- **Complex Multimedia Algorithms** Don’t Always Port Easily
Many C/C++ Prototypes

- MPEG Software Simulation Group
- Berkeley MPEG Tools
- Wavelet Image Construction Kit
- Telenor H.263 Code

- Read/write sequences of still images (PPM, TGA etc.)
- Often Grayscale Only
- Much work to convert to graphical applications.
Possible Java Advantages

- Immediate Demonstration on Web
  - Administrators and Funding Agencies
  - Colleagues
  - Potential Customers and Licensors
- Built-In Graphics Rendering
- Image and ImageFilter Classes
- Portable
- Free
Possible Java Disadvantages

- Slow (Interpreted Byte-code)
- No High Level Mathematics
- No symbolic manipulation
- Algorithm Must be Ported from Java to C to Create Product
- Not Portable
- Not Free
Java Image Processing Demo

- Demonstration of several standard edge enhancement and edge-detection algorithms.
- Moderately complex algorithms.
- A Real World Test of Java
- http://zeus.arc.nasa.gov/sword.html
Motivation for Demo Algorithms

- Study edge enhancement and edge detection algorithms for possible incorporation in a compression algorithm.
- Edge detection solves many problems: character recognition, flaw detection in silicon wafers, remote surveillance, and many other applications.
Why Edge Detection for Compression?

- Leading lossy video compression algorithms such as Discrete Cosine Transform and wavelets throw away or distort critical details at edges.
- Need to preserve sharp edges and lines for low bitrate image coding to look “natural” to a human viewer!
Edges and Block DCT

JPEG Encoder in Paint Shop Pro
Edges and Wavelets

Wavelet Image Construction Kit (Geoff Davis)
Demo Java Filter Applet

- Java Filter Applet (using JDK 1.1.4)
- http://zeus.arc.nasa.gov/sword.html
- The Filter Applet works differently under different Web browsers and operating systems. :-(
- Fails completely for a few browsers and operating system combinations. :-(
Images in Java

- Poorly documented
- java.awt Package
  - Image class
- java.awt.image Package
  - ImageProducer, ImageConsumer, etc.
- Here is my educated guess how it works
Images in Java

- Image class
- ImageProducer interface
- ImageConsumer interface
- ImageFilter class (implements ImageConsumer interface)
- FilteredImageSource class (implements ImageProducer interface)
Java Graphics and Images

Applet::paint(Graphics g) { g.drawImage(image,...) }

image = createImage(FilteredImageSource)

Java Virtual Machine decides when to update data?
Java Image Processing

Java
Graphics

FilteredImageSource

ImageFilter

ImageProducer
Applet.getImage().getSource()

Java Virtual Machine decides when to update data?
Image Class

- Abstract class represents a **displayable image** in a platform-independent way.
- **java.awt** Package (NOT **java.awt.image**)
- Don’t use constructor to create an Image
  - Applet.getImage() or createImage(producer)
- **Image.getSource()** returns the **ImageProducer** that produces the image data.
ImageProducer Interface

- Defines the methods that any class that produces image data must define to enable communication with ImageConsumer classes.
- FilteredImageSource (java.awt.image) is an ImageProducer
- java.awt.image Package
ImageProducer Interface

- **SOURCE of the image data**
- **not the IMAGE data**
- **file on hard disk**
- **URL of image (JPEG or GIF) on network**
- **a filter that processes an image**
  - FilteredImageSource
ImageConsumer Interface

- Interface defines the methods necessary for a class that consumes image data to communicate with a class that produces image data, an ImageProducer.
- ImageFilters in Java implement the ImageConsumer Interface
- java.awt.image Package
ImageFilter Class

- `java.awt.image` Package
- Implement the `ImageConsumer` interface
- Represent a filter that performs an operation on an image
- Filter method run when the `ImageFilter` retrieves new data from an `ImageProducer`? (JVM?)
FilteredImageSource Class

- Implements the ImageProducer Interface
- Constructor
  
  FilteredImageSource(ImageProducer orig, ImageFilter imgf)

- Image.getSource() retrieves the ImageProducer associated with a Java Image
FilteredImageSource Class

- Represents an object that takes data from another ImageProducer and applies a filter, represented by an ImageFilter class, and passes the filtered image on to another java object.
- Is itself an ImageProducer
Image Events?

- **ImageProducers** send “events” to **ImageConsumers** requesting image data.
- **Images, ImageConsumers, ImageProducers** maintain local buffers with copies of the image.
- **Different Java Virtual Machines** decide to send image events at different times.
Image Events?

- Standard does not define or products incorrectly implement the handshaking between ImageProducers, Consumers, and other graphics components.

- Updating of data in different objects occurs at different times with different Java Virtual Machines.
Conclusions

- **ImageFilter Class is a Great Idea.**
- **Java GUI easy to code.**
- **Slow, but slow is acceptable for a prototype.**
- **Portability is NOT THERE YET!**
  - Biggest problem with Java ImageFilters.
  - Limits DEMONSTRATION of prototypes.
Suggestions for Sun

- Improve Portability
  - standardize image updating?
- More Speed (Interpreted Java)
- Java to Machine Code Compiler
- High Level Math Classes
  - Matrices and Matrix Operations
  - Special Functions (e.g. Error Function)
Where to Get This Talk

- http://zeus.arc.nasa.gov/sword.html
  - The Java Demo
- http://zeus.arc.nasa.gov/jugfeb18.pdf
  - Adobe PDF Format Version
- http://zeus.arc.nasa.gov/jugfeb18.ppt
  - Microsoft *Windows* Power Point Version
- http://zeus.arc.nasa.gov/
  - Desktop Video Expert Center Web Site