Software development environments for image acquisition and processing

Plamen Tzvetkov, Georgi Petrov, Panayot Iliev
Technical University of Sofia, Sofia, 8 KlOhridski blvd., e-mail: tzvetkov@tu-sofia.bg
New Bulgarian University, Sofia, 21 Montevideo blvd., e-mail: gpetrov@nbu.bg
New Bulgarian University, Sofia, 21 Montevideo blvd., e-mail: piliev@nbu.bg

Abstract: Today most of worldwide technical universities have well established practice to include in their educational programs courses related to digital signal and image processing (DSP). The most of available study programs teach and use DSP as a compulsory “tool” for any kind of research. Because of this they may not be able to understand all massy image DSP technology, including sophisticated hardware development and low level system programming. They will only need to know the basic concepts of image acquisition & camera technology, processing mathematic, processor types and architectures. All students need is to have a basic vision of image DSP tips and tricks, and some computer programming skills. Most of described here software packets may be used even from very beginners in programming, using drag and drop user interface they can fast create powerful image DSP solutions without writing any line of source code, or just writing mathematical expressions.

The Image Acquisition Process
Before continue with different software packets, called Integrated Development Environments (IDE) available on the market, we may need to clarify the basic concepts of image acquisition and processing, or simply to say “What do I need to know?”. First of all you need to know does your application will work in real time or you will use already scanned images or image sequences. If need real time applications you need to select a proper video camera. Today you can find cheap video cameras for less than 20$, but they may not produce images with good resolution and quality. You need to consider what light spectrum your application needs to capture. Any Web camera will catch a visible light. In some cases you will need to handle images in low luminance, such as security applications and industrial control systems. Other special applications such as medical and chemical will have to manage with ultra violet light and even gamma rays – such as medical roentgen. Thus why camera selection is very complicated problem, many of commercially designed video cameras, such as web cameras are not sensitive and reliable enough to be used in life support applications.

Many other applications may need to handle a faster video stream, such as video security systems, flight control applications, motion analysis systems, car radar and others. After selecting your camera requirements you will need to spend much more money to take a really good image acquisition card. Because all described here software packets are personal computer (PC) oriented; this may be a PCI extension board or simple USB video capture. USB interface is acceptable only for slow speed video applications, 30 fr. /s, scanners and any other capture devices that work in non real time regime.
There are many commercial solutions that offer different types of image processing hardware, but we highly recommend you to start developing process from a PC platform. This is because PC architecture allows you a faster software development cycle, user friendly interface, and what is more important – you have large varieties of commercially IDE packets for image processing that simplify your design needs.

Choosing the Right Camera
- What is my target application?
- What kind of image acquisition is better suited – scanner, camera, and linear scan?
- Light spectrum – visible light, infra red, ultra violet, gamma rays?
- What is the acceptable resolution, and color depth of produced images?

Table 1 different types of cameras source: National Instruments

<table>
<thead>
<tr>
<th></th>
<th>Analog Cameras</th>
<th>Parallel Digital Cameras</th>
<th>Camera Link Cameras</th>
<th>IEEE 1394 Cameras</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
<td>Slow</td>
<td>Fast</td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td><strong>Spatial Resolution</strong></td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>Simple and easy</td>
<td>Advanced</td>
<td>Advanced</td>
<td>Simple and easy</td>
</tr>
<tr>
<td><strong>Pixel Depth</strong></td>
<td>8-bit to 10-bit</td>
<td>Up to 16-bit</td>
<td>Up to 16-bit</td>
<td>Typically 8-bit</td>
</tr>
<tr>
<td><strong>Cabling</strong></td>
<td>Simple BNC cabling</td>
<td>Thicker, custom cabling</td>
<td>Simple, standard cabling</td>
<td>Simple, standard cabling</td>
</tr>
</tbody>
</table>

Image Acquisition Hardware
You should consider the following hardware features for your machine vision or scientific imaging applications:
- Is it possible to use a PC based platform instead of DSP/FPGA solution;
- Does devices have onboard memory for faster preprocessing;
- Preprocessing functions that I need;
- Fast data transfer to PC memory – USB, PCI, FireWire;
- Integration with data acquisition and motion control hardware and other sensors;
- Can I program or simply use drag and drop programming interface to build my applications.

When talking for hardware implementation, some applications may use DSP and FPGA processors to maintain different tasks in image processing. Real time applications can’t be made without these specially designed platforms for signal processing. Some platforms may use a combination of DSP and FPGA chips with large amounts of inbuilt memory, multiple ADC channels for video capturing.

Image Acquisition and Processing Software
Matlab
We will start our “trip” in image processing IDE from the most popular educational packet Matlab. It is widely used in education. Matlab supports different kind of user interface: you can write formulas using mathematical expressions or programming language, simulink interface to make any kind of simulations with drag and drop icons, and use cross compilers to interface ready to use microcontroller and DSP kits. For the purpose of image processing Matlab has Image Acquisition Toolbox and Image Processing Toolbox. It makes you easy to perform image processing, analysis, and algorithm development. The Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can restore noisy or degraded images, enhance images for improved intelligibility, extract features, analyze shapes and textures, and register two images. Most toolbox functions are written in the open MATLAB language. This means that you can inspect the algorithms, modify the source code, and create your own custom functions.

Key Features of Image Acquisition Toolbox
- Toolbox. Identifying Available Devices - Identify installed image acquisition devices using the IMAQHWINFO function.
- Determine device information and capabilities.
- Accessing Devices and Video Sources - Access image acquisition devices using the VIDEOINPUT function.
- Select and access the current video source using the GETSELECTEDSOURCE function.
- Working With Properties - Access and configure a video input object's properties using the GET and SET functions.
- Access property help and information using the IMAQHELP and PROPINFO functions.
- Managing Image Acquisition Objects - Locate image acquisition objects using the IMAQFIND function. Remove image
acquisition objects from memory using the DELETE and CLEAR functions. Reset the Image Acquisition Toolbox using the IMAQRESET function.

**Key Features of Image Processing Toolbox**
- Image enhancement, including linear and nonlinear filtering, filter design, deblurring, and automatic contrast enhancement;
- Image analysis, including texture analysis, line detection, morphology, edge detection, segmentation, region-of-interest (ROI) processing, and feature measurement;
- Color image processing, including color space conversions and device-independent ICC profile import and export;
- Spatial transformations and image registration, including a graphical tool for control-point selection;
- Image transforms, including FFT, DCT, Radon, and fan-beam projection;
- Interactive image display and modular tools for building image GUIs;

- Support for multidimensional image processing.

**Mathcad**
The *Image Processing Extension Pack* lets you explore image processing with Mathcad. Perform smoothing, crisping, edge detection, erosion, and dilation algorithms on color and grayscale images. Or, compare different convolution kernels and experiment with filters in the Fourier transform domain. There’s even a 2D wavelet transform function allowing you to experiment with this popular transform method. You can read and write images in various formats, including BMP, GIF and JPEG. In addition to all the built-in processing functions, there’s a custom image viewer that updates images automatically along with your equations in Mathcad. The library contains more than 100 image processing functions that augment your Mathcad functions. Mathcad is an interpreter, so the speed is limiting you only for algorithm development process.

Figure 3. Using Mathcad in DSP is simple, you can write formulas without knowing how to program source: Mathcad help

---

**National Instruments - LabView Vision Library**
This company supports LabView extension libraries for image processing. Software packet and drivers support standard video captures and cameras; Vision supports the following basic functionality:
- Image Acquisition - Signal I/O, Integration with DAQ Hardware, Basic Acquisition
- Image Analysis - Optical Character Recognition, Blob Analysis, Spatial Calibration, Searching and Recognition, Color, Regions of Interest, Frequency Analysis, Gauging
- Image I/O - File storage, Display, General, Networking

- Image Processing - Operators, Filtering, Morphological Operations, Grayscale Processing
- Sample Image Libraries
- LabVIEW Real-Time

Machine Vision Systems with LabVIEW helps you to build your machine vision system with LabVIEW Real-Time provides an environment for complete deterministic control. Because the underlying real-time operating system is a simplified kernel, you have fewer chances for latency or delays caused by interrupt service routines. Machine vision systems running LabVIEW Real-Time easily integrate with other measurement and automation devices.
**National Instruments**

**Vision Academic Starter Kit**
The NI VASK consists of machine vision hardware and software, designed for developing machine vision and scientific imaging applications. Hardware and Software for Developing Machine Vision and Scientific Imaging Applications supports:
- High-level machine vision and image processing functions as well as display tools
- High-speed pattern matching for locating objects of various size and orientation
- 4-channel monochrome image acquisition board for standard or nonstandard video sources
- Compatible with double-speed 60 frames/s progressive scan cameras
- 8 or 10-bit digitization, Interlaced/noninterlaced acquisition
- Blob analysis for calculating 82 parameters, including the area, perimeter, and location of objects

“Processing”
Is an open source programming language and environment for people who want to program images, animation, and sound. It is used by students, artists, designers, architects, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. Processing is developed by artists and designers as an alternative to proprietary software tools in the same domain.

**Image++**
Image++ is an Object-oriented graphic development tool for digital image processing and Machine Vision. It supports Microsoft and Borland compilers. It generates DLL's out of C++ source. Free C++ source code of many arithmetic operations, binarization, morphology, filter, FFT, Hough, histogram functions, color conversion, measurement functions is supplied. Image++ is ideally suited for education in schools, universities, Machine Vision application, medicine, image enhancement, etc. Comes with full Setup, uninstall feature and On-Line product help. A complete manual can be downloaded.

**Free Software for End Users**

**VirtualDub**
This program is a video capture/processing utility for 32-bit Windows platforms (95/98/ME/NT4/2000/XP), licensed under the GNU General Public License (GPL). It lacks the editing power of a general-purpose editor such as Adobe Premiere, but is streamlined for fast linear operations over video. It has batch-processing capabilities for processing large numbers of files and can be extended with third-party video filters. VirtualDub is mainly geared toward processing AVI files, although it can read (not write) MPEG-1 and also handle sets of BMP images. VirtualDub has the capability to load third-party DLLs to use different video filters. The filters that are built into VirtualDub use the same interface that is exported to DLLs, so you can write filters similar to or better than the included ones.
Software Specialized for Programmers

Image Acquisition - Scanning

TWAIN

The TWAIN initiative was originally launched in 1992 by leading industry vendors who recognized a need for a standard software protocol and applications programming interface (API) that regulates communication between software applications and imaging devices (the source of the data). TWAIN defines that standard. The three key elements in TWAIN are the application software, the Source Manager software and the Data Source software. The application uses the TWAIN toolkit which is shipped for free. Packet can help you:

- Ensure users of image acquisition software and devices experience success every time;
- Ensure image-handling software and hardware compatibility;
- Define and distribute the TWAIN specification;

Video Recording and Rendering

Here we describe two programming interfaces for capturing and processing video and images for Windows and Linux operating systems:

DirectShow

Formerly ActiveMovie is a multimedia architecture developed by Microsoft. It is part of Windows 98, Windows 2000, Windows XP and Internet Explorer, and available separately for free download from Microsoft (as part of DirectX). DirectShow divides the processing of multimedia tasks such as video playback into a set of steps known as filters. Filters have a number of input and output pins which connect them together. The generic design of the connection mechanism means that filters can be connected in many different ways to achieve different tasks, and developers can add their own effects or other filters at any stage in the graph. DirectShow filter graphs are widely used in video playback (in which the filters will provide steps such as file parsing, video and audio de-multiplexing, decompressing and rendering) as well as being used for video and audio recording and editing. Interactive tasks such as DVD navigation are also successfully based on DirectShow. To get started on developing a DirectShow filter, try the filter wizard for Visual Studio from John McAleely. This is way of creating a Visual C++ project for a DirectShow filter. It will create the code for a filter with no pins. This can then be modified by you to add pin and other code to create a useful filter.

OpenGL

OpenGL is the industry's most widely used, supported and best documented 2D/3D graphics API making it inexpensive & easy to obtain information on implementing OpenGL in hardware and software. There are numerous books, tutorials, online coding examples, coding seminars, and classes that document the API, Extensions, Utility Libraries, and Platform Specific Implementations. OpenGL supports: Microsoft Windows; Apple Mac OS and X Window System for SUN, FreeBSD and Linux.

Image SDK/ActiveX

Is a software development tool that helps to create applications with sophisticated image processing capabilities? The SDK works on every Windows operating system from Windows Server 2003 to Windows 95. It includes both C/C++ libraries and ActiveX controls. It can be used with programming languages like C, C++, Visual Basic, Delphi, MS FoxPro, and MS Access. Document Imaging SDK/ActiveX is also .NET compatible meaning that VB.NET, C#, and J# programmers can also take full advantage of the product. Some of basic functionality:

- Display any image to any display and 1, 8, and 24 bit images on any display device;
- Use 4 dithering methods to convert color images to monochrome and grayscale;
- Full control of how images are displayed: normal, scale to DPI, fit to width, fit to window;
- Rotate images in 90-degree increments, rotate images to a specific degree, flip images, skew and de-skew images, crop images and anti-aliasing and advanced zoom;
- Scanning support based on TWAIN drivers.
- Apply sophisticated image processing algorithms and filters.

Selecting a proper programming language

When we program applications that will handle with large amounts of data such as images, and if it will perform mathematical intensive operations such as FFT and statistic the chose of proper language is the most important thing. Because for education needs you will mainly use a PC platforms, the most suitable is C++, .NET framework supports Java, Visual Basic, C#. Linux programmers prefer to use Java. C# is an interpreter language and it works about 4-10 times slower than C. When you deal with real time systems DSP and FPGA use assembler, C or VHDL languages.

References:
[8] http://www.codeproject.com – a FREE place for sharing your code with other professional programmers


Ass. Prof. Panayot Yordanov Iliev MS in Radio electronics (1972). PhD in Image processing and analysis (1980); Working in NBU department Telecommunications; Area of scientific interests intelligent measurement systems, simulation, control, and DSP.

Authors: