A Fingertip Detection and Tracking System as a Virtual Mouse, a Signature Input Device and an Application Selector

Abstract

This project presents vision-based fingertip detection and tracking system on a 2D plane using a camera as an input interface to the computer. An intensity based approach is used to detect the arbitrary shaped, uniform colored 2D area on which the hand operates, and then the fingertip is effectively detected and tracked using the sampled hand contour. Grid sampling approach is used for a fast implementation. The system achieves speeds of up to 30 fps.Tip pointers such as a stylus or a pen can be used in place of the fingertip, making the device user-friendly. The system is used to implement a virtual mouse, a signature input device and an application selector.

1. Introduction:

In the field of computer vision, many prototypes of intelligent vision-based interface systems have been developed that are more intuitive and cost-effective than the conventional interface devices such as the mouse and the keyboard. We describe such a system that detects and tracks the fingertip in a 2-D plane, and then demonstrate its application as a virtual mouse, a signature input device and an application selector. The system is shown in Figure 1.

It comprises a panel, which is a 2D object of an arbitrary shape and a uniform color. The panel acts as a pad on which the finger is to be operated. A camera is focused on the panel to observe the tip pointer. This acts as the interface link between the user and the computer. Firstly, we present an overview of the system. Then, we present an analysis of the system to describe in detail, the techniques used for detecting the panel and detecting and tracking the fingertip. Finally, we present the application of the fingertip detection and tracking system as (a) a virtual mouse, (b) a signature input device and (c) an application selector. The system achieves a speed of up to 30 frames per second on a computer running on Intel Core 2 Duo processor, 1.73 GHz with 1GB RAM.

1.2. Company Profile.

2. System Study

The System study phase analyzes the problems of existing systems, defines the objectives to be attained by a solution and evaluates various solution alternatives

2.1 Existing System

- Exploring vision-based interfaces is motivated by the unnaturalness of some of the conventional input devices such as mice and joysticks in many intelligent environments where intuitive interactions and teleoperations are required.
- The bottleneck of such devices comes from the lack of flexibility due to the constraints from the environments, and the lack of the feeling of immersiveness in human computer interaction.

2.2 Proposed System

- Future work on the system can include the integration of a character-recognition system with this system.
- Various extensions and improvements in the fingertip detection and tracking system can be explored.

2.3 Advantages of Proposed System

- We have developed a prototype of vision-based fingertip detection and tracking system on a 2D plane based on the intensity of captured images.
- This approach combined with the method of grid sampling results in speeds up to 30 frames per second with images of resolution 640x480 pixels.
- This is higher than the speeds that have been achieved by some similar methods as documented in.

3. System Analysis & Feasibility

The analysis of a problem that we will try to solve with an information system; describes what a system should do.

3.1 Feasibility Study:

Next step in the analysis is feasibility study. By performing feasibility study the scope of the system will be defined completely.

Most computer systems are developed to satisfy a known user requirement. This means that the first event in the life cycle of a System is usually the task of studying whether it is feasible to a Computerize a system under consideration or not. Once the decision is made, a report is forwarded and is known as Feasibility Report.

The Feasibility is studied under three contexts

a. Technical Feasibility
b. Economic Feasibility
c. Operational Feasibility

3.1.1 Technical Feasibility

What resources are available for given developer system? Is the problem worth solving? In the proposed system, technical feasibility centers on the existing computer system (hardware, software etc.) and what extent it can support the proposed system. There should not be more cost involved here for the hardware because all the hardware required are present in the existing system and software specified also exists. There fore, now we needed to install the software on existing system for the project. And the operation of this system requires knowledge about Windows XP or window Professional. This assistance would be easily available.

Even though these technical requirements are needed for implementing the system, once the code is generated and compiled, the executable code of the project is sufficient to run the application.

Hence the proposed system is technically feasible.

3.1.2 Economic Feasibility

Economic feasibility is used for evaluating the effectiveness of a candidate system. The procedure is to determine the cost and benefits/savings that are expected from a candidate system and compare with the costs. If cost is less and benefit is high, then the decision is made to design and implement the system. All the required facilities, hardware and software, to be used are initially may be costly, but when put to use it proves to be much more economical than the existing system. Regarding the maintenance, since the source code will be with the company, any small and necessary changes can be done with minimum maintenance cost involved in it. The system that is developed and installed must be good investment for organization. The organization has to spend the amount for technology, as it is not computerized. The present system performance is high when compared to the previous system. So for the organization the cost factor is acceptable, so it is economically feasible.

If installed will certainly be beneficial since there will be reduction in manual work, and increase in the speed of work there by increasing the profit of company and saving time. The proposed system is cost effective one compared with the current existing system. Hence the system is economically feasible.

3.1.3 Operational feasibility

Fingertip Detection and tracking system is mainly involved in looking to Virtual mouse operations. This is done by using special control and monitoring software. The monitoring system should include features like

- Automatically opening the application based on the virtual mouse click.
- Finger tip signature will be taken as the input to authenticate the application
- Automatic typing is done in the screen.

The main problem in developing a new system is getting acceptance and the co-operation from the users because many users are reluctant to operate on a new system. The software being developed is more interactive. With the developing system, it is instantaneous; moreover even a new person can operate the system and easily execute the system. So it is operationally feasible.

Since the system relives the management of maintaining an voluminous records in a login at user level, the proposed system would be acceptable by the user and is operationally feasible.

3.2 Packages Selected:

The Microsoft .NET Framework 3.0 is the new managed code programming model for Windows. It combines the power of the .NET Framework version 2.0 with new technologies for building applications that have visually compelling user experiences, seamless communication across technology boundaries, and the ability to support a wide range of business processes. These new technologies are Windows Presentation Foundation, Windows Communication Foundation, Windows Workflow Foundation, and Windows CardSpace. The .NET Framework 3.0 is included as part of the Windows Vista operating system; you can install it or uninstall it using Windows Features Control Panel. This redistributable package is for Windows XP and Windows Server 2003.

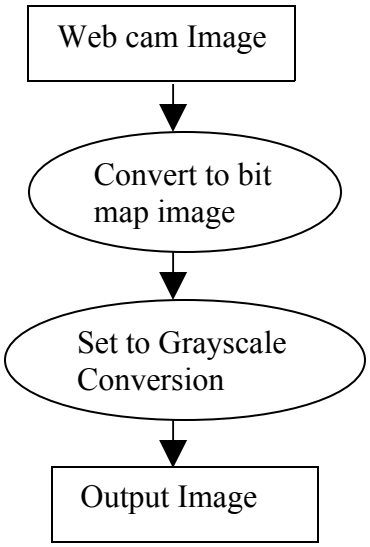
Microsoft Windows XP Service Pack 2 (SP2) provides new proactive security technologies for Windows XP to better defend against viruses, worms, and hackers. In addition to a more robust security infrastructure, SP2 improves the security configuration options of Windows XP and provides better security information to help users faced with security decisions.

3.3 Resource Required

Planning and analyzing the resources is also the one of the major part of the SDLC to complete the project with in a given time. In this phase we need to analyze the availability of the resources that are required to design, develop, Implement and Test the project. The resources to be analyzed are Employees, Time and the SRS. Teams of three members are involved in the entire SDLC life cycle except the testing phase. The testing phase is guided by the Manual testers before the hosting the application in the server space.

Time Analyzed to complete the project is approximately three months with 3 hrs on daily basis except week ends. SRS is prepared and provided as per the URS.

3.4 Dataflow Diagram

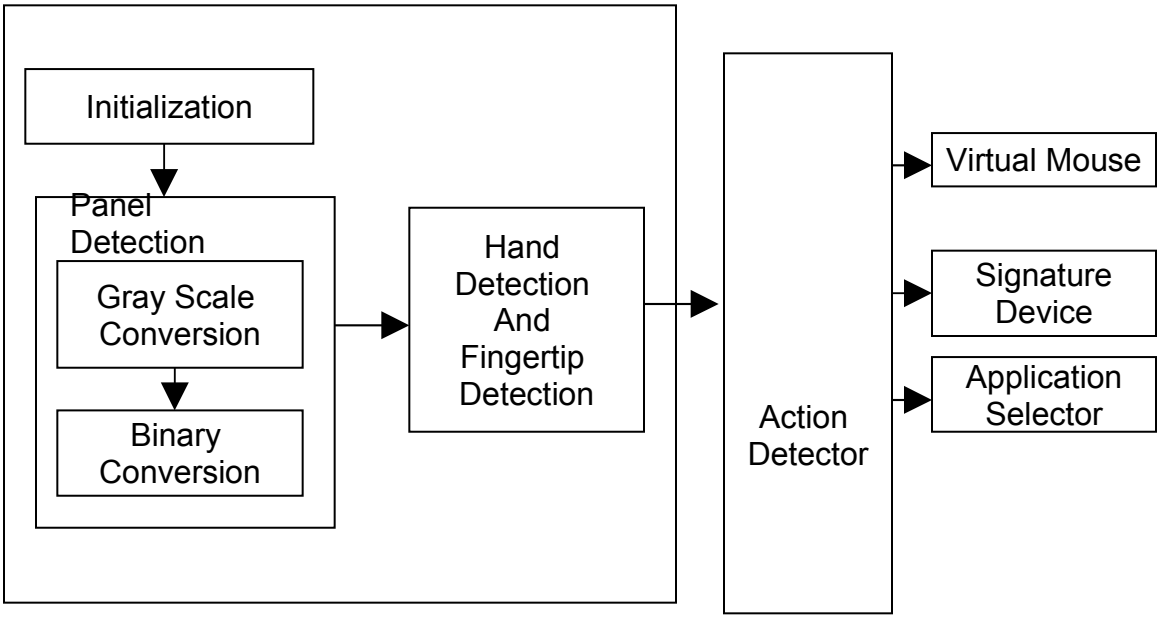


4. System Design

In the Design phase of SDLC both the Logical and physical design specifications for the systems solution are produced.

4.1 Architectural Design

Architecture diagram shows the relationship between different components of system. This diagram is very important to understand the over all concept of system



4.2 Hardware Specifications

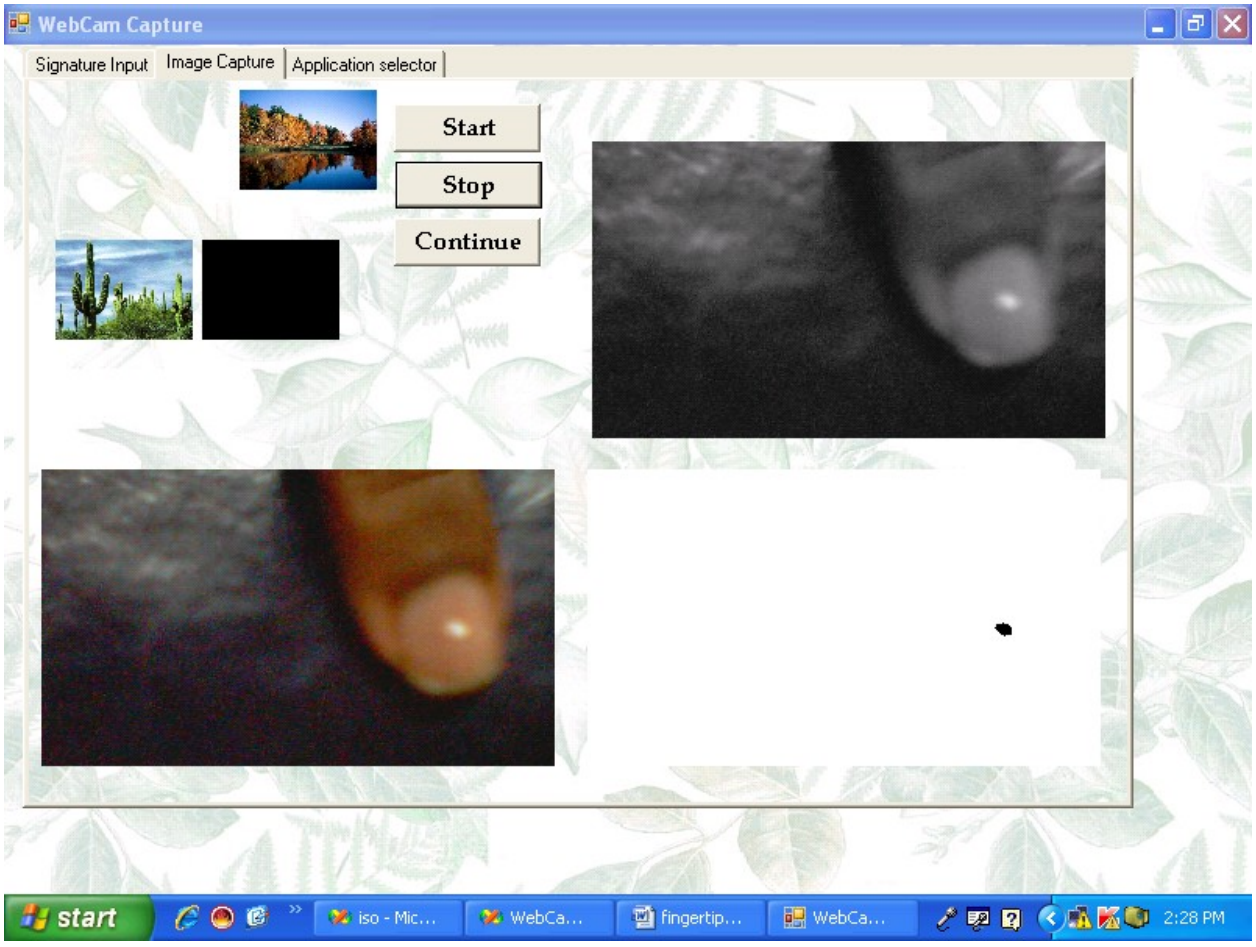
- Processor : Pentium III
- Clock speed : 550MHz
- Hard Disk : 20GB
- RAM : 128MB
- Cache Memory : 512KB
- Operating System : windows xp
- Monitor : Color Monitor
- Keyboard : 104Keys
- WebCam

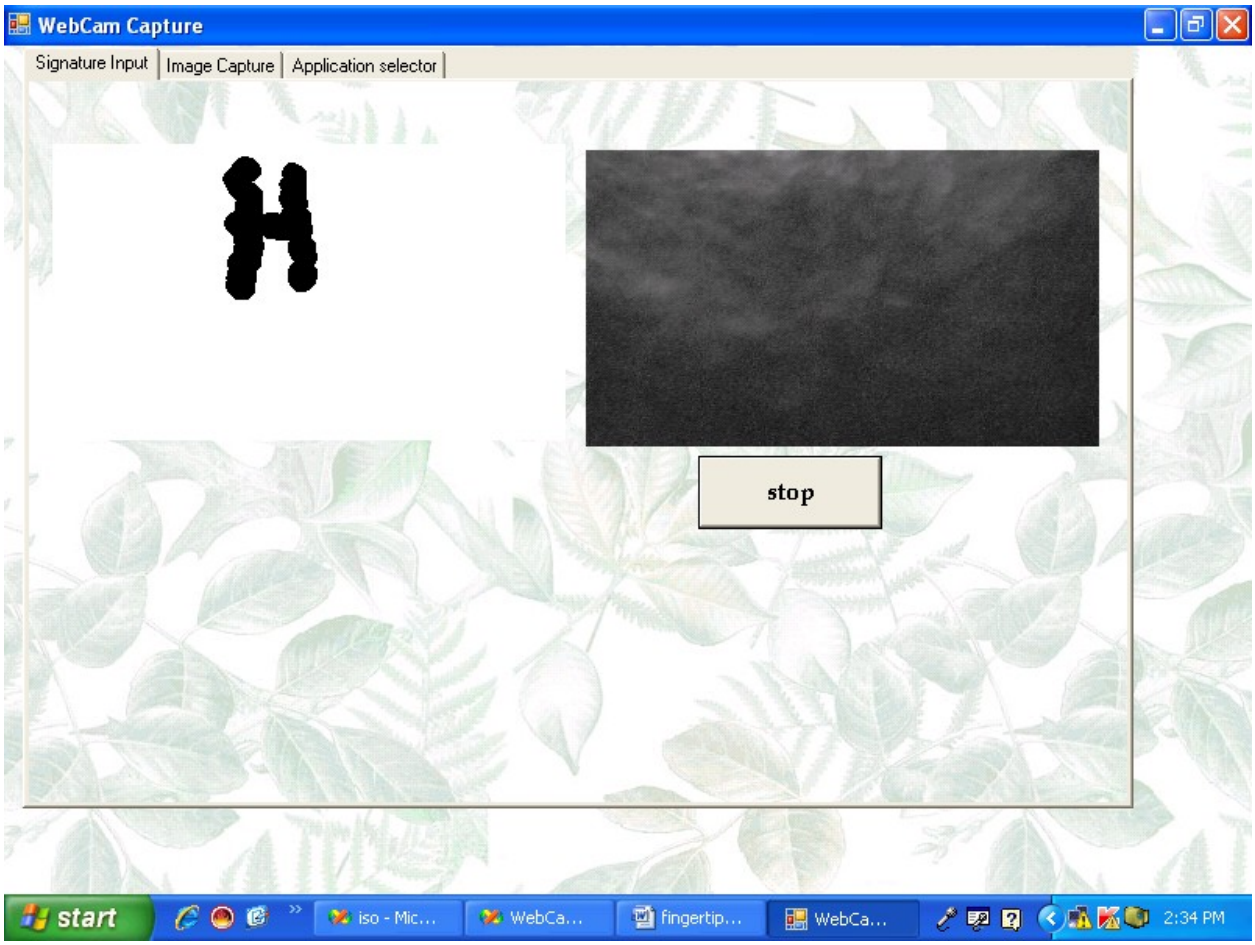
- LED Array

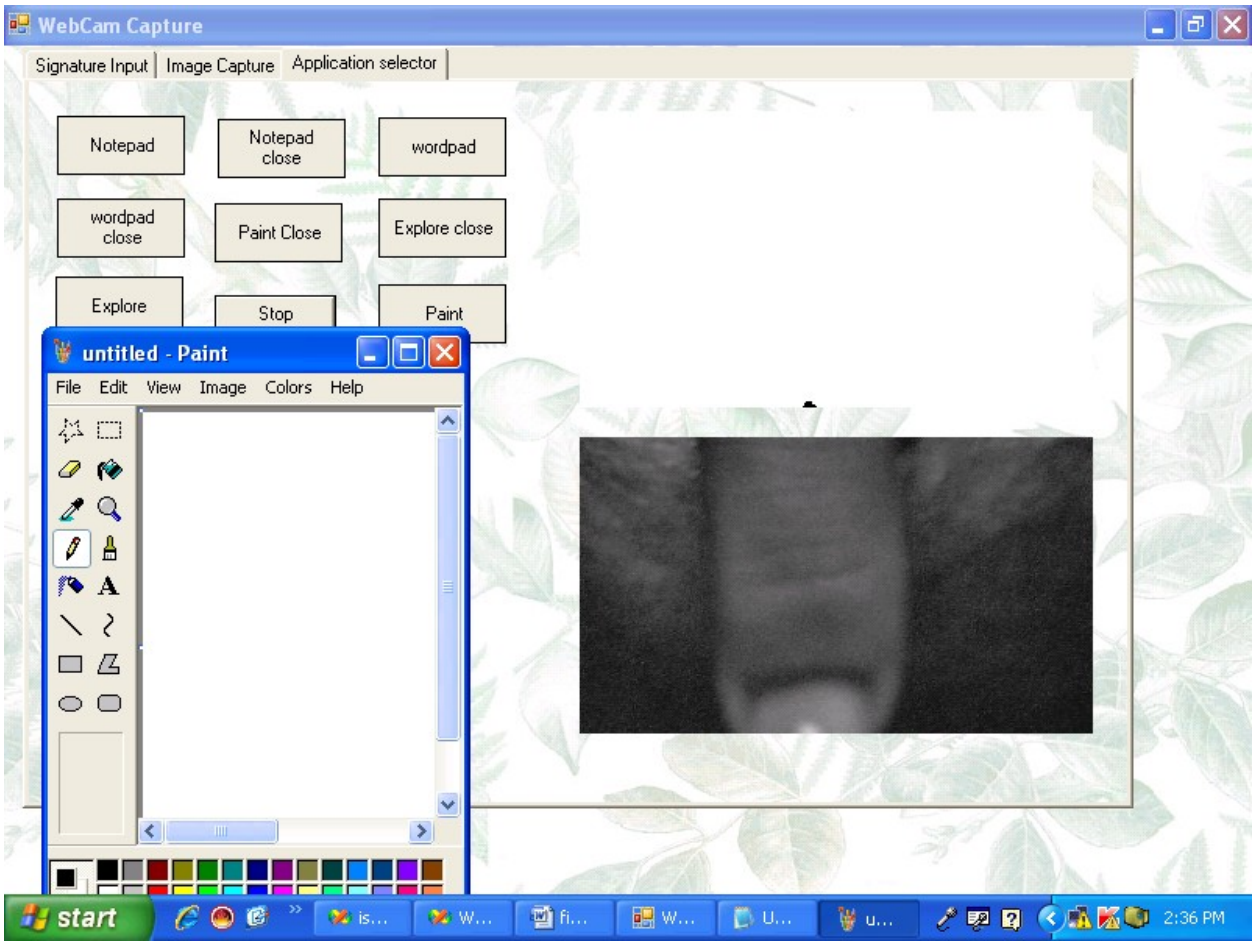
4.3 Software Requirements

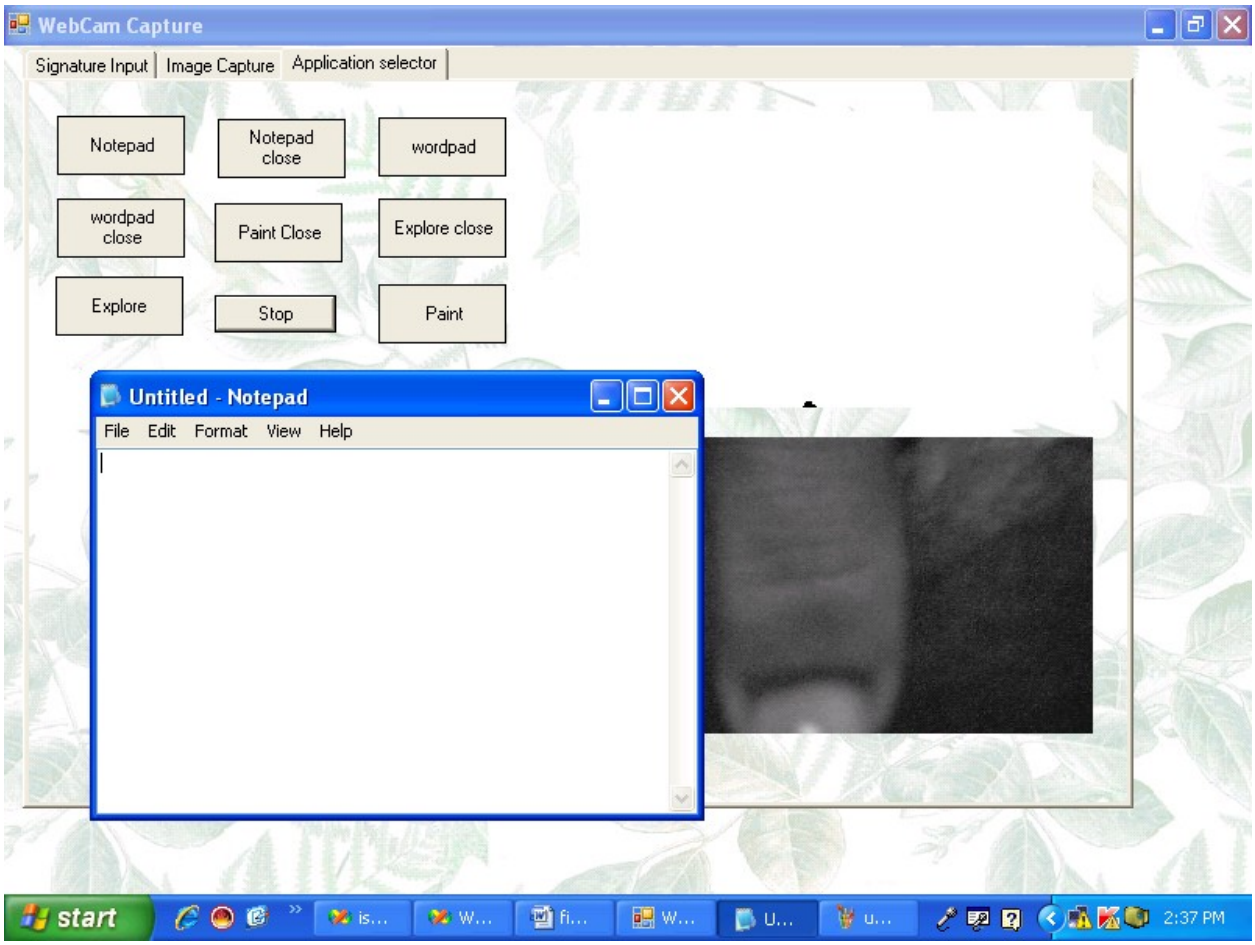
- Microsoft XP
- VB.NET

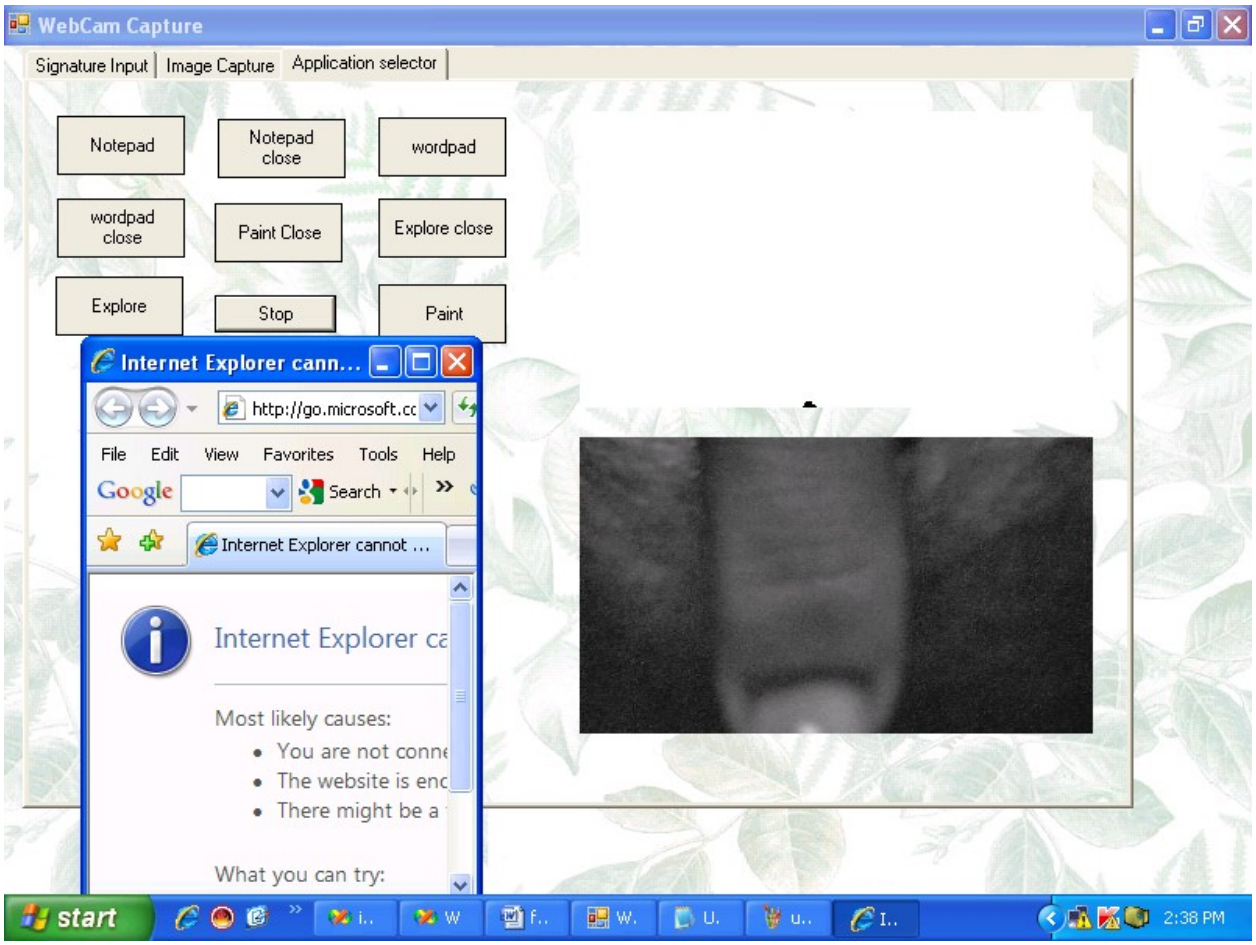
4.4 Screen shots

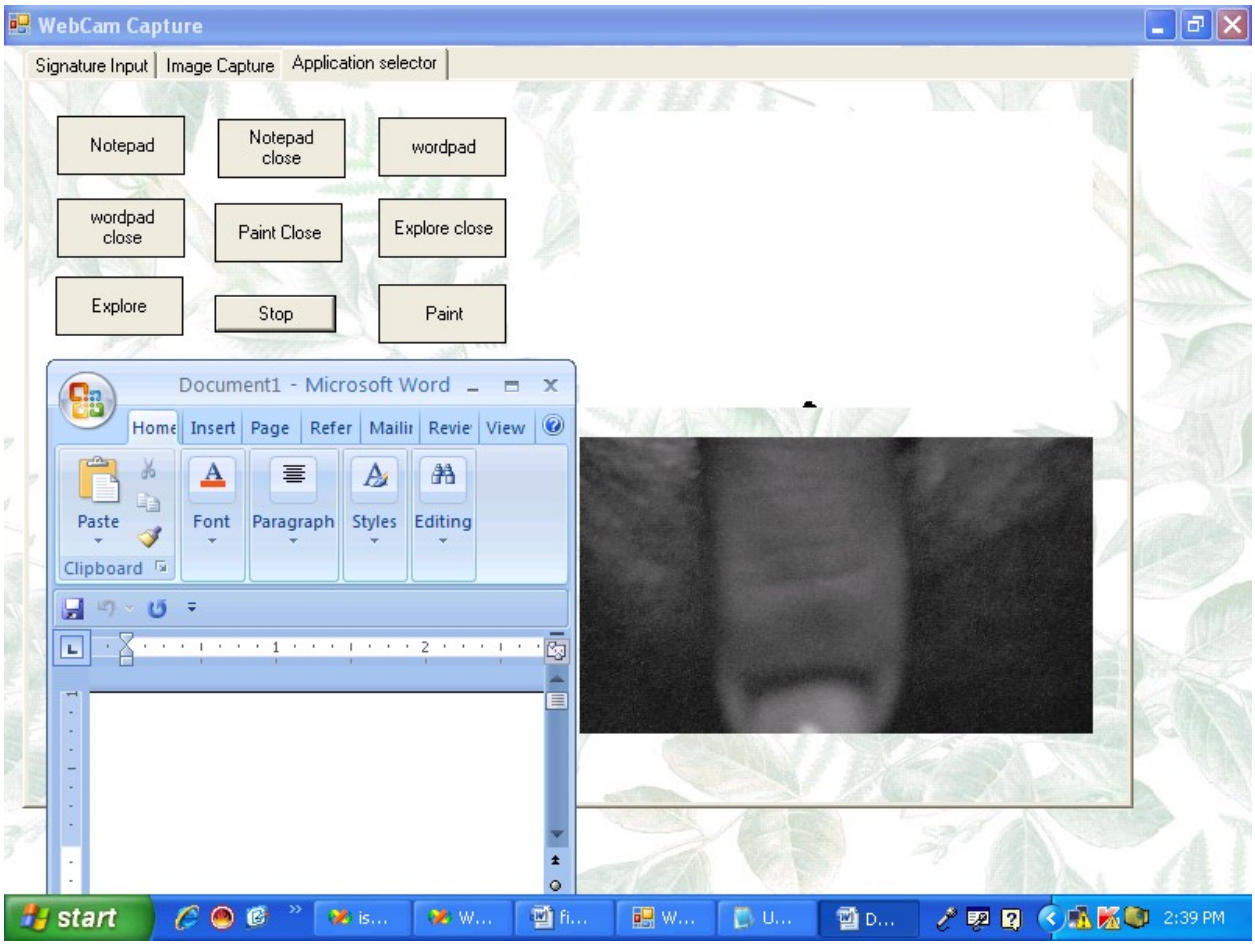



WebCam Capture
Signature Input
Image Capture
Application selector
stop
start
iso - Mic...
WebCa...
fingertip...
WebCa...
2:34 PM


WebCam Capture
Signature Input
Image Capture
Application selector
Notepad
Notepad close
wordpad
wordpad close
Paint Close
Explore close
Explore
Stop
Paint
untitled - Paint
File
Edit
View
Image
Colors
Help
start
2:36 PM


WebCam Capture
Signature Input
Image Capture
Application selector
Notepad
Notepad close
wordpad
wordpad close
Paint Close
Explore close
Explore
Stop
Paint
Untitled - Notepad
File Edit Format View Help
start
2:37 PM


WebCam Capture
Signature Input
Image Capture
Application selector
Notepad
Notepad close
wordpad
wordpad close
Paint Close
Explore close
Explore
Stop
Paint
Internet Explorer cann...
http://go.microsoft.cc
File Edit View Favorites Tools Help
Google
Search
Internet Explorer cannot ...
Internet Explorer ca
Most likely causes:
You are not conne
The website is enc
There might be a
What you can try:
start
2:38 PM


WebCam Capture
Signature Input
Image Capture
Application selector
Notepad
Notepad close
wordpad
wordpad close
Paint Close
Explore close
Explore
Stop
Paint
Document1 - Microsoft Word
Home
Insert
Page
Refer
Mailir
Revie
View
Paste
Clipboard
Font
Paragraph
Styles
Editing
start
2:39 PM

5. Coding & Debugging

5.1 Module Description:

The System as a Virtual Mouse

The new coordinates of the tip location are compared with the previous coordinates, and the difference between the coordinates is algebraically added to the previous coordinates. This results in mouse motion. The speed of the mouse can be controlled by using a multiplying factor M to multiply the difference between coordinates before addition to previous coordinates. Mathematically:

$Px, t = M*(Cx, t – Cx, t-1) + Px, t-1$ (1)

$Py, t = M*(Cy, t – Cy, t-1) + Py, t-1$ (2)

Where, ‘P’ denotes the pointer location on screen,

‘C’ denotes the coordinates of tip,

‘M’ denotes the multiplying factor,

‘x’ & ‘y’ denote the rectangular x & y axes respectively,

‘t’ denotes the present time.

Our present implementation supports single clicking mode. The clicking mode is simulated by holding the tip stationary at a location for a short duration of time, say 2 seconds. This simulates the left click of the mouse. The position of the pointer may not remain constant due to noise and hence if the fluctuation is within 1 grid pixel, the tip is considered immobile.

The System as a Signature Input Device

The Signature Capture ON instruction is generated by holding the tip pointer stationary at a location for a short duration of time, say 2 seconds. When the application is activated, the user is asked to input his/her signature. The tip pointer coordinates obtained from successive images are joined by straight lines as shown in Fig. 5. The Signature Capture OFF instruction is generated by holding the tip pointer stationary at a location for a short duration of time after completing the signature, say 2 seconds. This application provides an efficient and economical way of obtaining digital images of signatures.

Fig. 5 Virtual image during the signature input process

The System as an Application Selector

In this application, the panel acts like a touch pad to initiate different commands/programs/applications. We implemented a touch-pad that provides 4 options of applications to the user. The A4 sheet is divided into blocks each specifying a particular application (as shown in Fig. 6), and the commands that trigger the application corresponding to each block are pre-stored in the memory. The paper orientation is fixed. After panel detection, the coordinates of the four vertices of the A4 sheet are stored at a memory location, and with their help, the equations of the diagonals of the quadrilateral panel are determined. Let the equations of the two diagonals be:

$Ax + By + C = 0$ (3)

$Dx + Ey + F = 0$ (4)

Where A, B, C, D, E and F are constants; x and y are variables of the Cartesian coordinate system.

The 4 blocks, as shown in Fig. 6, can be specified in terms of the equations of the diagonals as:

Block 1: Ax + By + C < 0 and Dx + Ey + F < 0 (5)

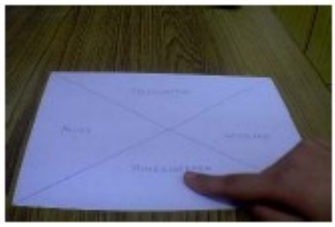
Block 2: Ax + By + C < 0 and Dx + Ey + F > 0 (6)

Block 3: Ax + By + C > 0 and Dx + Ey + F < 0 (7)

Block 4: Ax + By + C > 0 and Dx + Ey + F > 0 (8)

An application is selected by holding the finger stationary on the corresponding block for a short duration of time, say 2 seconds. The system determines the block in which the tip pointer is held constant with the help of (5) – (8) and activates the pre-stored commands corresponding to that block.

Fig. 6 The panel for application selector



5.2 About the Software

Visual Basic.NET: A New Framework

Microsoft .Net Framework is an integral Windows component that supports building and running the next generation of applications and XML Web services. The key components of the .NET Framework are the common language runtime and the .NET Framework class library, which includes VB.NET,ADO.NET, ASP.NET AND Windows Forms. The .NET framework provides a managed execution environment simplified development and a deployment and integration with a wide variety of programming languages. The .NET framework is a new computing platform that simplifies application

development in the highly distributed environment of the Internet. The .NET framework is designed to fulfill the following objectives

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally but internet-distributed or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code, created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments
- To make the developer experience consistent across widely varying types of applications and web based applications
- To build all communication on industry standards to ensure that code based on the .NET framework can integrate with any other code.

As you already know, .NET is a name for a new strategy: a blueprint for building applications for the next decade. It's actually even more than that. It's Microsoft's commitment to remain at the top of a rapidly changing world and give us the tools to address the needs of tomorrow's computing.

Visual Basic .NET is a language for creating .NET applications, like many others. It also happens that Visual Basic is the easiest to learn, most productive language (but you already know that).Visual Basic .NET is released shortly after the tenth anniversary of the first version of VB. The original language that changed the landscape of computing has lasted for 10 years and has enabled more programmers to write Windows application than any other language. Programmers who invested in Visual Basic 10 years ago are in demand today.

In the world of computing, however, things change very fast, including languages. At some point, they either die, or they evolve into something new. Visual Basic was a language designed primarily for developing Windows applications. It was a simple language, because it managed to hide many of the low-level details of the operating system. Those who wanted to do more with Visual Basic had to resort to Windows API.

In a way, earlier versions of Visual Basic were 'sandboxed' to protect developers from scary details.

Microsoft had to redesign Visual Basic. The old language just didn't belong in the .NET picture (at least, it wouldn't integrate very well into the picture). Visual Basic .NET is not VB7; it's a drastic departure from VB6, but a necessary departure. Visual Basic .NET was designed to take us through the next decade of computing, and if you want to stay ahead, you will have to invest the time and effort to learn it. The most fundamental component of the .NET initiative is the .NET Framework, or simply the Framework. You can think of the Framework as an enormous collection of functions for just about any programming task. All drawing methods, for example, are part of the System.Drawingclass. To draw a rectangle, you call the DrawRectangle method, passing the appropriate arguments. To create a new folder, you call the CreateDirectory method of the Directory class; to retrieve the files in a folder, you call the GetFiles method of the same object. The Framework contains all the functionality of the operating system and makes it available to your application through numerous methods.

VB was such a success because it was a very simple language. You didn't have to learn a lot before you could start using the language. Being able to access the Framework's objects means that you're no longer limited by the language. The new version of the language unlocks the full potential of .NET; now there's hardly anything you can do with another language but can't do with Visual Basic. This makes the language as powerful as any other language, but it also makes the learning curve steeper.

Development environments:

- Visual Studio .NET (VS .NET)
- Visual Web Developer

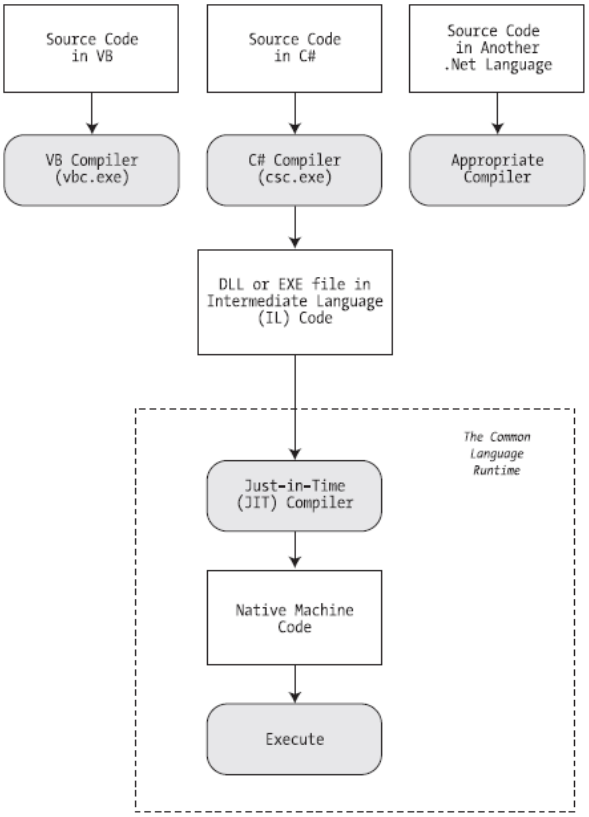
Intermediate Language:

All the .NET languages are compiled into another lower-level language before the code is executed. This lower-level language is the Common Intermediate Language (CIL, or just IL). The CLR, the engine of .NET, uses only IL code. Because all .NET languages are designed based on IL, they all have profound similarities. This is the reason that the

VB and C# languages provide essentially the same features and performance. In fact, the languages are so compatible that a web page written with C# can use a VB component in the same way it uses a C# component, and vice versa.

The .NET Framework formalizes this compatibility with something called the Common Language Specification (CLS). Essentially, the CLS is a contract that, if respected, guarantees that a component written in one .NET language can be used in all the others. One part of the CLS is the common type system (CTS), which defines the rules for data types such as strings, numbers, and arrays that are shared in all .NET languages. The CLS also defines object oriented ingredients such as classes, methods, events, and quite a bit more. For the most part, .NET developers don't need to think about how the CLS works, even though they rely on it every day.

Figure shows how the .NET languages are compiled to IL. Every EXE or DLL file that you build with a .NET language contains IL code. This is the file you deploy to other computers. In the case of a web application, you deploy your compiled code to a live web server.



Other .NET Languages:

VB and C# aren't the only choices for ASP.NET development. Developers can also use J# (a language with Java-like syntax). You can even use a .NET language provided by a third-party developer, such as a .NET version of Eiffel or even COBOL. This increasing range of language choices is possible thanks to the CLS and CTS, which

define basic requirements and standards that allow other companies to write languages that can be compiled to IL. Although you can use any .NET language to create an ASP.NET web application, some of them do not provide the same level of design support in Visual Studio, and virtually all ASP.NET developers use VB and C#. For more information about third-party .NET languages.

The Common Language Runtime:

The CLR is the engine that supports all the .NET languages. Many modern languages use runtimes. In VB 6, the runtime logic is contained in a DLL file named msvbvm60.dll. In C++, many applications link to a file named mscrt40.dll to gain common functionality. These runtimes may provide libraries used by the language, or they may have the additional responsibility of executing the code (as with Java). Runtimes are nothing new, but the CLR is Microsoft's most ambitious runtime to date. Not only does the CLR execute code, it also provides a whole set of related services such as code verification, optimization, and object management.

All .NET code runs inside the CLR. This is true whether you're running a Windows application or a web service. For example, when a client requests an ASP.NET web page, the ASP.NET service runs inside the CLR environment, executes your code, and creates a final HTML page to send to the client.

The implications of the CLR are wide-ranging:

***Deep language integration*:** VB and C#, like all .NET languages, compile to IL. In other words, the CLR makes no distinction between different languages—in fact, it has no way of knowing what language was used to create an executable. This is far more than mere language compatibility; it's language *integration*.

***Side-by-side execution*:** The CLR also has the ability to load more than one version of a component at a time. In other words, you can update a component many times, and the correct version will be

loaded and used for each application. As a side effect, multiple versions of the .NET Framework can be installed, meaning that you're able to upgrade to new versions of ASP.NET without replacing the current version or needing to rewrite your applications.

***Fewer errors*:** Whole categories of errors are impossible with the CLR. For example, the CLR prevents many memory mistakes that are possible with lower-level languages such as C++. Along with these truly revolutionary benefits, the CLR has some potential drawbacks. Here are three issues that are often raised by new developers but aren't always answered:

***Performance*:** A typical ASP.NET application is much faster than a comparable ASP application, because ASP.NET code is compiled to machine code before it's executed. However, processor-crunching algorithms still can't match the blinding speed of well-written C++ code, because the CLR imposes some additional overhead. Generally, this is a factor only in a few performance-critical high-workload applications (such as real-time games). With high-volume web applications, the potential bottlenecks are rarely processor-related but are usually tied to the speed of an external resource such as a database or the web server's file system. With ASP.NET caching and some well-written database code, you can ensure excellent performance for any web application.

***Code transparency*:** IL is much easier to disassemble, meaning that if you distribute a compiled application or component, other programmers may have an easier time determining how your code works. This isn't much of an issue for ASP.NET applications, which aren't distributed but are hosted on a secure web server.

***Questionable cross-platform support*:** No one is entirely sure whether .NET will ever be adopted for use on other operating systems and platforms. Ambitious projects such as Mono (a free implementation of .NET on Linux, Unix, and Windows) are currently underway. However, .NET will probably never have the wide reach of a language such as Java because it incorporates too many different platform-specific and operating system–specific technologies and features.

The .NET Class Library:

The .NET class library is a giant repository of classes that provide prefabricated functionality for everything from reading an XML file to sending an e-mail message. If you've had any exposure to Java, you may already be familiar with the idea of a class library. However, the .NET class library is more ambitious and comprehensive than just about any other programming framework. Any .NET language can use the .NET class library's features by interacting with the right objects. This helps encourage consistency among different .NET languages and removes the need to install numerous components on your computer or web server.

Some parts of the class library include features you'll never need to use in web applications (such as the classes used to create desktop applications with the Windows interface). Other parts of the class library are targeted directly at web development. Still more classes can be used in various programming scenarios and aren't specific to web or Windows development. These include the base set of classes that define common variable types and the classes for data access, to name just a few. You'll explore the .NET Framework throughout this book.

.NET Framework deals with thorny issues like database transactions and concurrency, making sure that hundreds or thousands

of simultaneous users can request the same web page at once. You just add the logic needed for your specific application.

Visual Studio:

The last part of .NET is the Visual Studio development tool, which provides a rich environment where you can rapidly create advanced applications. Although in theory you could create an ASP.NET application without Visual Studio (for example, by writing all the source code in a text editor and compiling it with .Net's command-line compilers), this task would be tedious, painful, and prone to error. For that reason, all professional ASP.NET developers use a design tool like Visual Studio

.

Some of the features of Visual Studio include the following:

***Page design*:** You can create an attractive page with drag-and-drop ease using Visual Studio's integrated web form designer. You don't need to understand HTML.

***Automatic error detection*:** You could save hours of work when Visual Studio detects and

reports an error before you run your application. Potential problems are underlined, just like the "spell-as-you-go" feature found in many word processors.

***Debugging tools*:** Visual Studio retains its legendary debugging tools, which allow you to watch your code in action and track the contents of variables. And you can test web applications just as easily as any other application type, because Visual Studio has a built-in web server that works just for debugging.

***IntelliSense*:** Visual Studio provides statement completion for recognized objects and automatically lists information such as function parameters in helpful tool tips. You don't need to use Visual Studio to create web applications. In fact, you might be tempted to use the freely downloadable .NET Framework and a simple text editor to create ASP.NET web pages and web services.

Visual Studio is available in several editions. The Standard Edition has all the features you need to build any type of application

(Windows or web). The Professional Edition and the Team Edition increase the cost and pile on more tools and frills (which aren't discussed in this book). For example, they incorporate features for managing source code that's edited by multiple people on a development team and running automated tests.

The scaled-down Visual Web Developer Express Edition is a completely free version of Visual Studio that's surprising capable, but it has a few significant limitations. Visual Web Developer Express Edition gives you full support for developing web applications, but it doesn't support any other type of application. This means you can't use it to develop separate components for use in your applications or to develop Windows applications. However, rest assured that Visual Web Developer Express Edition is still a bona fide version of Visual Studio, with a similar set of features and development interface.

The .NET Languages

The .NET Framework ships with three core languages that are commonly used for building ASP.NET applications: C#, VB, and J#. These languages are, to a large degree, functionally equivalent. Microsoft has worked hard to eliminate language conflicts in the .NET Framework. These battles slow down adoption, distract from the core framework features, and make it difficult for the developer community to solve problems together and share solutions. According to Microsoft, choosing to program in C# instead of VB is just a lifestyle choice and won't affect the performance, interoperability, feature set, or development time of your applications. Surprisingly, this ambitious claim is essentially true.

.NET also allows other third-party developers to release languages that are just as feature rich as C# or VB. These languages (which include Eiffel, Pascal, Python, and even COBOL) "snap in" to the .NET Framework effortlessly. In fact, if you want to install other .NET languages, all you need to do is copy the compiler to your computer and add a line to register it in the computer's machine.config configuration file. Typically, a setup program would perform these

steps for you automatically. Once installed, the new compiler can transform your code creations into a sequence of Intermediate Language (IL) instructions, just like the VB and C# compilers do with VB and C# code. IL is the only language that the Common Language Runtime (CLR) recognizes. When you create the code for an ASP.NET web form, it's changed into IL using the C# compiler (csc.exe), the VB compiler (vbc.exe), or the J# compiler (vjc.exe). Although you can perform the compilation manually, you're more likely to let ASP.NET handle it automatically when a web page is requested.

6. System Testing and Implementation

6.1 System Testing

Any software has to be tested with pre-planned strategies. As Roger Pressmen states, the preparation for testing should start as soon as the design of system starts, to carry out the testing in an efficient manner certain amount of strategic planning has to be done. Any testing strategy must incorporate test planning, test case design, test execution and the resultant data collection and evaluation.

6.2 Unit Testing:

In the lines of this strategy all, the individual functions and modules were put to the test independently. By following this strategy all, the errors in coding were identified and corrupted. This method was applied in combination with the white and black box testing techniques to find the errors in each module.

6.3 Integration Testing:

Again this software testing strategy has different approach in which integration is carried out from the top level module to the bottom and the bottom up approach in which integration is carried out from the low level module to the top.

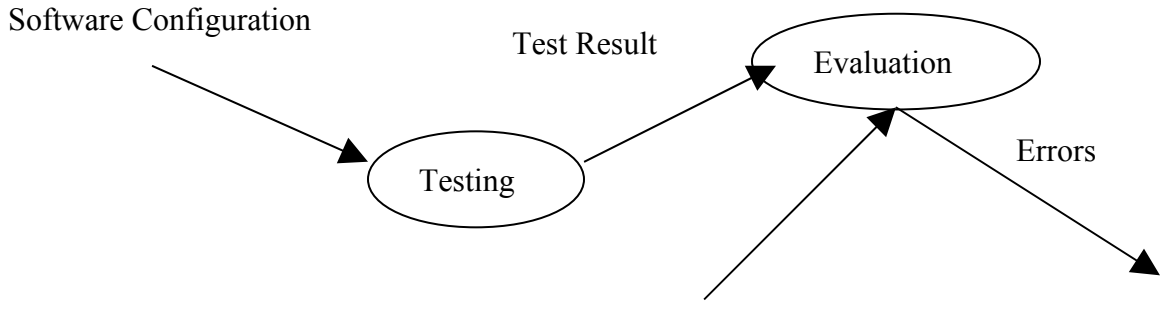
- The modules are tested using the bottom up approach by introducing stumps for the top level functions.

- This test used to identify the errors in the interfaces, the errors in passing the parameters between the functions and corrects them.

6.4 Validation Testing:

Validation testing is done to validate the inputs given by the user. The user inputs are checked for their correctness and range. If there are errors, the error message is given and the user is prompted again to enter the new value. If the user types some characters in the numeric field an error message and it is demonstrated in the following figure.

6.5 Testing Information Flow:



7. User Manual

7.1 System Overview

Firstly, we define a virtual image, which is of the same size as that of the captured image and is black in color, i.e. the value of all pixels is taken to be 0 (on a scale of 0 to 1). System analysis can be divided into three parts: Panel detection, hand detection, and fingertip detection and tracking.

Panel Detection

For a fast implementation with high-resolution images, we use the method of grid sampling as described in [2]. The captured image is divided into grids of block size 10 pixel x 10 pixels. Each block is referred to as a grid pixel. Hence, an image of size 640x480 pixels is transformed to an image of size 64x48 grid pixels. Each captured image is separated into its 3 component images corresponding to red, green and blue.

In each of the three images, each grid pixel is analyzed. If the intensity of 95% pixels, i.e. 95 pixels out of 100 in a grid pixel, is greater than 0.85 (an empirical value on a scale of 0 to 1 for a white A4 sheet), then the corresponding grid pixel in the virtual image is set to 0.5. Hence, the panel in the captured image corresponds to a gray region in the virtual image. A number of consecutive captured images are analyzed to check whether the panel is static or not. This is done by comparing the position of the gray pixels in the consecutive virtual images. When it is ascertained that the panel is static, the virtual image is saved in a permanent location and the panel is said to be successfully detected. The actual intensity of the panel is also stored at a permanent location. A command is outputted by the system that the fingertip can be brought on the panel. An example of panel detection is shown in Fig. 2.

Fig. 2(a) Captured image of panel

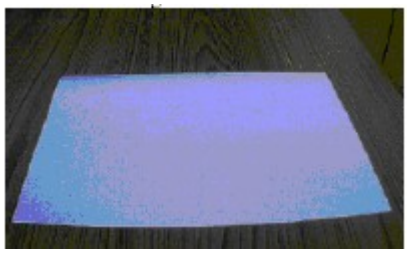


Fig. 2(b) Corresponding virtual image of panel

Hand Detection

When the hand is brought on the panel, the grid pixels of the captured image are analyzed again. If 50% of all pixels of a grid pixel lying on the panel have intensity less than 0.65 (an empirical value on a scale of 0 to 1 for a hand, taking values between 0.65 and 0.85 as a buffer), then the corresponding grid pixel in the virtual image is set to 1.Hence the hand in the captured image corresponds to a white region in the virtual image. An example of hand detection is shown in Fig. 3.

Fig. 3(a) Captured image of hand on panel

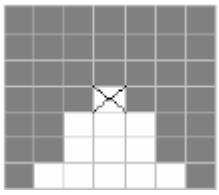
Fig. 3(b) Fingertip in the corresponding virtual image

Fingertip Detection and Tracking

For fingertip detection, a square of 7 pixel x 7 pixels with a white grid pixel at its center is considered, for every white grid pixel present on the panel. The number of white grid pixels in this square is calculated. The white grid pixel whose corresponding square has the least number of white grid pixels is the required position of fingertip. The coordinates of this tip position are calculated and returned to the system. This tip position is continuously updated for incoming images, and by this means the tracking of fingertip

is achieved. The method is graphically shown in Fig. 4, where the square corresponding to the X marked grid pixel is constructed, and it consists of 12 white grid pixels. A buffer region of 3 pixels is left on the edge of the detected panel for an efficient algorithm. An example of fingertip detection is shown in Fig. 3.

Fig. 4 A square corresponding to the X marked white grid pixel



7.2 System Implementation:

Implementation is that stage in the project where the theoretical design is turned into a working system. The most crucial stage in achieving a new system and effectively. The first step in implementing the system is in getting approval from the sample user. This is done in view of any last minute changes that will be necessary in the formats. When the sample users are satisfied, finally the site is implemented in .Net. The more complex the system being implemented, the more involved will be system's analysis and design effort required for implementation.

7.3 Installation Procedure

If you have been directed to do a server installation, you must have the following software installed in addition to the typical installation requirements: ASP.NET is supported only on the following platforms: Microsoft Windows 2000 Professional (Service Pack 3 recommended), Microsoft Windows 2000 Server (Service

Pack 3 recommended), Microsoft Windows XP Professional, and Microsoft Windows Server 2003 family.

- Any of the Operating System Mentioned above can be installed.
- Internet Information Services (IIS) version 5.0 or later. To access the features of ASP.NET, IIS with the latest security updates must be installed prior to installing the .NET Framework.

8. Conclusion

We have developed a prototype of vision-based fingertip detection and tracking system on a 2D plane based on the intensity of captured images. This approach combined with the method of grid sampling results in speeds up to 30 frames per second with images of resolution 640x480 pixels. The system uses an arbitrary shaped panel of uniform color (e.g. an A4 sheet) and a web-cam, objects that are inexpensive and easily available in present day organizations. The fingertip can be substituted by a tip pointer such as a stylus or a pen. This makes the device convenient and user-friendly. Three applications have been developed based on the fingertip detection and tracking system. The virtual mouse is an economical substitute of the hardware mouse. The signature input device eliminates the complexity involved in devices such as digital pens or touch-sensitive pads, and can be used in daily life to store digital images of signatures. This can find

application in organizations such as banks. The application selector provides an inexpensive alternative to touch-sensitive pads and screens.

9. Future Enhancement

Future work on the system can include the integration of a character-recognition system with this system. Various extensions and improvements in the fingertip detection and tracking system can be explored.

10. References

- Ying Wu, Ying Shan, Zhengyou Zhangy, Steven Shafer, "Visual Panel: From an ordinary paper to a wireless and mobile input device," Technical Report, MSR-TR-2000 Microsoft Research Corporation, http://www.research.microsoft.com, October 2000.
- Duan-Duan Yang, Lain-Wen Jin, Jun-Xun Yin, Li-Xin Zhen, Jian-Cheng Huang, "An effective robust fingertip detection method for finger writing character recognition system," Proceedings of the Fourth Int'l Conference on Machine Learning and Cybernetics, Guangzhou, pp. 4191 – 4196, August 2005.
- J. L. Crowley, F. Berard and J. Coutaz, "Finger tracking as an input device for augmented reality," Proceedings of Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 195-200, June 1995.

- F. K. H. Quek, T. Mysliwiec and M. Zhao, “Finger Mouse: A freehand pointing computer interface,” Proceedings of Int’l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 372-377, June 1995.
- Christian. V. H, François. Bm., “Bare-hand human computer interaction,” Proceedings of the 2001 workshop on Perceptive user interfaces, Orlando, Florida, USA, pp. 1-8, Nov 2001.
- Oka. K, Sato. Y, Koike. H, “Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface system,” Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 411 – 416, May 2002.
- R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd ed, Prentice Hall, 2002.