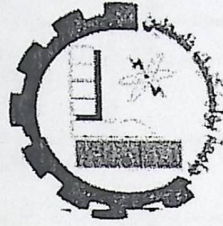


Palestine Polytechnic University
Collage of Administrative Sciences and Informatics
Department of Information Technology



Using Augmented Reality in Education

Project team:

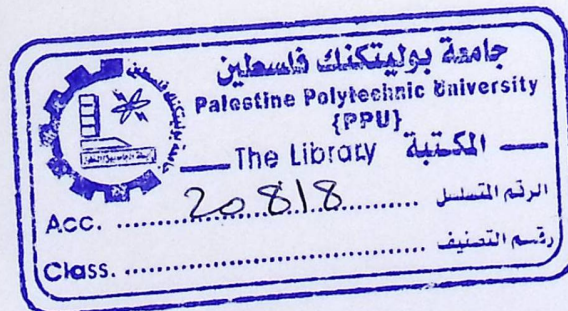
Ashwaq ALshrabati
Maha Safi
Sawsan ALbakri

Supervisor:

Dr. Hashem Tamimi

A final project submitted in partial fulfillment of the requirements for the degree of
B.Sc. in Information Technology

2007



List of Figures:

Figure name	Page number
Figure 4.1 ARToolKit general work	13
Figure 4.2 Context diagram	14
Figure 4.3 DFD level 1	15
Figure 4.4 DFD level 2	16
Figure 4.5 Initialize video capture flow chart	17
Figure 4.6 Detect marker process flow chart	19
Figure 4.7 Draw process flow chart	20
Figure 4.8 clean process flow chart	21
Figure 4.9 Demo for AR using ARToolKit	22
Figure 5.1 Motherboard	25
Figure 5.2 CPU	25
Figure 5.3 RAM	25
Figure 5.4 Context diagram for interactive Augmented Reality in computer architecture	27
Figure 5.5 Motherboard with marker	28
Figure 5.6 MainChip before and after adding virtual object	
Figure 5.7 RAM before and after adding virtual object	29
Figure 5.8 RGB Model	30
Figure 5.9 HSV Model	31
Algorithm 5.1 RGB to HSV	33
Figure 5.10 DFD level 1 interactive Augmented Reality in computer architecture	35
Figure 7.1 Data flow from mainchip to RAM	49
Figure 7.2 CPU send request to mainchip	50
Figure 7.3 interrupt for CPU	50
Figure 7.4 Detect pen using RGB	51

List of Tables:

Table Name	Page number
Table 3.1 requirement cost table	11

Abstract:

In this project we will demonstrate the using of Augmented Reality (AR) in education, in order to move the trend of education methodology from theoretical way to practical way, and to decrease learning time. We developed an educational experiment which is teaching computer architecture to apply the idea of augmented reality; our experiment illustrates the flow of data and show the component of computer. This prototype enhances understanding process of computer architecture by observe the data movement in computer.

Interactive education is one of recent trend that followed by many universities in the world cause of the benefit that can be taken through interacting with specific content and support the education process. To make it powerful, more interesting and more understandable, to achieve that we apply this idea by developing interactive computer architecture course using augmented reality. This done by using image processing abilities to detect where the user point on a given virtual object on the marker, and then make the action.

We believe that student's understandings of spatial content change through their physical interactions with virtual objects. Also interactive learning which include, seeing, and touching things, increases student's perception for specific subject. The results of this project will demonstrate the potential benefits of using AR visualization interfaces in education and training.

CHAPTER One

Introduction

Introduction: Using Augmented Reality in Education:

There are some difficulties which the students face in understanding certain topics, recognize how it work so to enhance learnability, increasing their interactivity with lesson and motivate creativity in different educational level and making education process more interesting. For this reason we used augmented reality concept which has been applied on wide variety of science like chemistry, medicine, computer science and others.

Augmented Reality (AR): is the ability to overlay computer graphics onto the real world¹. It is also a field of computer research which deals with the combination of real world and computer generated data. At present, most Augmented Reality research is concerned with the use of live video imagery which is digitally processed and "augmented" by the addition of computer generated graphics.

The objective we want to achieve in this project is applying new idea in the education to enhance learnability and the creativity of the students; real world is enhanced or augmented with additional information generated from a computer model.

We developed an application to apply the idea of Augmented Reality on computer architecture, we illustrate in our experiment internal computer architecture.

In order to develop one of the applications of Augmented Reality we used ARToolKit because it deals with video in real time also it deal with camera parameter and video configuration, it operate under multiple platform, also it can operate with virtual modeling language(VRML), and it build for Augmented Reality application.

¹ <http://www.newhorizons.org/strategies/technology/billinghurst.htm>

Project outline:

Our project is divided into the following chapters:

Chapter 1: Include an introduction about the whole system; it talks generally about Augmented Reality, and using it in education.

Chapter 2: Includes a preview about previous studies for some researchers, it talks about some researches was made on the concept of Augmented Reality, and some projects that have been implemented.

Chapter 3: Includes software and hardware requirements that are important for Developer and user to make and operate Augmented Reality applications, it also contains some functional and non functional requirements with the cost of these requirements.

Chapter 4: Explains how to design Augmented Reality applications using ARToolKit, in this chapter we talks about ARToolKit general work and how these tools help in developing Augmented Reality applications, and preview to demo's developed by using ARToolKit.

Chapter 5: Talks about using Augmented Reality in education and, especially in learning computer architecture, this chapter about the experiment that we has made, also it explain how to design interactive Augmented Reality.

Chapter 6: Describes in details the functions used in programming, what is the basic functions that should be in any Augmented Reality application that use ARToolKit, and what these functions do.

Chapter 7: It Includes results that we reach after working in the project, and results.

CHAPTER Two

Literature Review

Introduction:

Researchers are interested in the Augmented Reality, so in their research and papers they discuss the concept of Augmented Reality what is it? Why we use it? In what areas can we use it? In order to clarify some scientist opinion and some projects that have been made we will make a preview for some previous studies.

Ronald T. Azuma defined Augmented Reality as a system that has three characteristics:

Combines real and virtual, Interactive in real time, and registered in 3-D [1], he also talked about using AR in different areas, for example he said that *doctors* can use AR as a visualization and training aid for surgery, by displaying where to drill a hole into the skull for brain surgery or where to perform a needle biopsy of a tiny tumor. The information will be directly display on the patient, showing exactly where to perform the operation also virtual instructions can remind a novice surgeon of the required steps, without the need to look away from a patient to consult a manual. Virtual objects could also identify organs and specify locations to avoid disturbing. Also that we can use Augmented Reality in *manufacturing and repair* for complex machine, he said that instructions might be easier to understand if they are available, not as manuals with text and pictures, but rather as 3-D drawings superimposed upon the actual equipment, showing step-by-step the tasks that need to be done and how to do them, and we can point to any of the machine or engine parts and the name of the pointed part will appear on it. [1]

Other researchers [2] talked about using Augmented Reality in military area, the concept of building such a system depend on that mobile AR take the position and orientation of worriers and give him information about enemy forces in form of drawing special mark on it, and friendly forces position in form of drawing another mark on it, also the paper clarify how mobile augmented reality help worriers in training in similar condition.

Also entertainment and games has been taken in Augmented Reality applications such as **The Invisible Train [3]** which is a mobile, collaborative multi-user Augmented Reality (AR) game, in which players control virtual trains on a real wooden miniature railroad track. These virtual trains are only visible to players through their PDA's video see-through display as they don't exist in the physical world. Players can interact with the game environment by operating track switches and adjusting the speed of their virtual trains. The current state of the game is synchronized between all participants via wireless networking. The common goal of the game is to prevent the virtual train from colliding.

Single user Augmented Reality interfaces has been developed for computer aided instructions [4], using this approach in area like manufacturing [5], and medical visualization [6], enable users to interact with the real world, and make work done easier. In manufacture area AR can used in order to teach the user how to repair things [4], such as a laser printer, user can see virtual objects and notation in order to make practical explanation for repairing the printer, also in medical area doctors can see virtual objects on the patient's body allowing them to have "X-Ray" vision on a needle biopsy task [6].

Also Augmented Reality have been used in collaborative work, a lot of projects is made to enhance collaborative work such as StudierStube project of Schmalsteig [7] in this project they use see-through head mounted displays to enable the users to collaboratively view 3D model of scientific data superimposed the real world. Another work for collaborative work AR2Hokey [8] in this application players wear see-through head mounted displays and play with virtual buck by using real table and mallet, they send the virtual buck to each other goal.

In computer-based education researchers have found that even though every student has his own computer they not perform good as when they all around one single machines [9], also when students work on individual machines they will spontaneously cluster around machines in pairs and trios [10,11], they also have found that physical objects is commonly used to convey meaning and it has the ability to focus attention. Also enhance the physical objects, For example, in the Shared Space interface users could manipulate three-dimensional virtual objects simply by moving real cards that the virtual models appeared attached to [12].

There was no mouse or keyboard in sight. This property enables even very young children to have a rich educational experience.

When using Augmented Reality interface with students who set on around table, that will be more comfortable than setting all together in front of the same machine; and they can see each other and make good discussion about the virtual object that appears in the middle of the table [13].

MagicBook [14] project is an educational project for children, in this project the user can read in a normal book without using any technology, and when using a handheld AR display he will see 3D virtual objects appear out from the real page, and the user can see the object from different view, by moving the book.

CHAPTER Three

Requirements

Requirements:

- **Hardware requirement:**

The basic hardware requirement we need to our experiment USB camera, personal computer and special pattern card.

1. Camera:

The most important part in hardware requirement choosing camera, there are some consideration that must be taken when camera has been chosen such as **resolution**², **frame-rate**³, **optics**⁴ **distortion**⁵. On the other hand supported platforms, parameters of the driver also influence the choice of the selected camera.

The traditional choice is USB or Firewire camera. The frame rate or color palettes depend mainly on the bandwidth of the technology. Firewire Cameras offers better solution, but camera with full RGB color palette is generally more expensive than USB therefore we will use USB camera.

2. Pattern:

This is a unique card we use it to apply our experiment that can be recognized by camera in order to add virtual object on it, also we can build our special pattern.

² Typically refers to the number of distinct pixels in each dimension that can be displayed.

http://en.wikipedia.org/wiki/Display_resolution

³ Is most often expressed in frames per second, or simply hertz (Hz).

http://en.wikipedia.org/wiki/Display_resolution

⁴ The branch of physics that deals with light and vision, chiefly the generation, propagation, and detection of electromagnetic radiation having wavelengths greater than x-rays and shorter than microwaves.

<http://www.answers.com/topic/optics>

⁵ A **distortion** is the alteration of the original shape (or other characteristic) of an object, image, sound, waveform or other form of information or representation.

<http://en.wikipedia.org/wiki/Distortion>

Software requirements divided into two parts:

- **Developer requirements:**

1. **Windows operating system, Windows XP, or Windows 2000, or Windows NT [15].**

2. **ARToolKit** which is a collection of libraries, utilities applications, and documentation and sample code. The libraries provide the user with a means to capture images from video sources, process those images to optically track markers in the images, and to allow compositing of computer-generated content with the real-world images and display the result using OpenGL (Phillip Lamb, 2004). ARToolKit is designed to build on Windows, Linux, SGI Irix, and Macintosh OS X platforms.

The latest version is available from <http://sourceforge.net/projects/artoolkit> and includes the following files that should be downloaded with ARToolKit:

- ARToolkit-2.70.tgzplatformindependent.
- DsVideoLib-0.0.4-win32.zipi386.
- OpenVRML-0.14.3-win32.zip i386.

ARToolKit is made available under a dual-license model. As it has been since the first public release of version 1.0, ARToolKit is freely available for non-commercial use under the terms of the GNU General Public License.

3. **Development environment [16]:** To execute ARToolKit examples we need to download an environment that ARToolKit can work under it, such as Visual Studio 6 and Microsoft Visual Studio.NET 2003. Visual Studio.NET is the only development environment built from the ground up to enable integration through XML Web services. By allowing applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

VS2003 can be operated on: Windows 2000; Windows NT; Windows Server 2003; Windows XP.

4. **OPEN Graphics Language (OpenGL):** A 3D graphics language developed by SGI (Silicon Graphics, Inc.), which has become a de facto standard supported in UNIX, Windows and Macintosh computers. OpenGL can be implemented as an extension to an operating system or to a windowing system such as X Window. All high-end 3D display adapters include OpenGL drivers. OpenGL and Direct3D (Microsoft's DirectX) are the major 3D graphics languages in use today.⁶

OpenGL makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps. And it provides a set of commands that allow the specification of geometric objects in two or three dimensions. The OpenGL API was designed for use with the C and C++ programming languages but there are also bindings for a number of other programming languages such as Java, and FORTRAN.

5. **Virtual Reality Modeling Language (VRML):** specification for displaying 3-dimensional objects on the World Wide Web. We can think of it as the 3-D equivalent of HTML. Files written in VRML have wrl extension (short for world). VRML produces a hyperspace (or a *world*), a 3-dimensional space that appears on your display screen. And you can figuratively move within this space. That is, as you press keys to turn left, right, up or down, or go forwards or backwards, the images on your screen will change to give the impression that you are moving through a real space.⁷ VRML is not a general purpose programming language like C++, a script language like JavaScript or a page specification language like HTML. It's a scene description language that describes the geometry and behavior of a 3D scene or "world".

⁶http://www.pcmag.com/encyclopedia_term/0,2542,t=OpenGL&i=48490,00.asp

⁷<http://isp.webopedia.com/TERM/V/VRML.html>

6. **DSVideoLib:** is used to handle communication with the camera driver. DSVideoLib-0.0.8b or later is required for ARToolKit.⁸ It needs Operating System 32-bit MS Windows (95/98), it was programmed by using C++ programming language.

7. **OpenVRML:** is a free cross-platform runtime for VRML and X3D available under the GNU Lesser General Public License. The OpenVRML distribution includes libraries you can use to add VRML/X3D support to an application. OpenVRML also provides a plug-in to render VRML/X3D worlds in Web browsers.⁹The existence of this library is optional in ARToolKit applications because it is required to operate some examples, like SimpleVrml example.

8. **OpenGL Utility Toolkit (GLUT):** is a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. GLUT makes it considerably easier to learn about and explore OpenGL programming. GLUT provides a portable API so you can write a single OpenGL program that works across all PC and workstation OS platforms.
 GLUT is designed for constructing small to medium sized OpenGL programs. While GLUT is well-suited to learning OpenGL and developing simple OpenGL applications, GLUT is not a full-featured toolkit so large applications requiring sophisticated user interfaces are better off using native window system toolkits. GLUT is simple, easy, and small.¹⁰

⁸ http://www.hitl.washington.edu/artoolkit/documentation/usersetup.htm#comp_windows

you can download Dsvideolib from <http://sf.net/projects/artoolkit>

⁹ <http://openvrml.sourceforge.net/>

You can download OpenVrml from this site: <http://openvrml.sourceforge.net/>.

¹⁰ <http://www.opengl.org/resources/libraries/glut/>

User requirements:

- The executable files exe and dll files that needed in order to run the program.
- Webcam to run the program.
- Special card that determined by developer.
- Windows XP as operating system.

Now we can summaries the **prerequisite** software as following:

- Microsoft Visual Studio.NET 2003 or Visual Studio 6.
- Windows XP.
- DSVideoLib-0.0.4-win32.
- GLUT.
- DirectX 9.0b or later SDK.
- (Optional, for VRML rendered only) OpenVRML-0.14.3-win32

Table 3.1 shows the cost of the requirements needed in order to Build AR applications:

Component	Cost
HP Compaq d230 Microtower Desktop PC	\$1000
USP webcam	\$50
Windows XP	\$299
Visual studio 2003	\$892.95.
VRML	\$214
Total cost	\$2415.95

Table3.1 Requirement cost table

User requirements:

- The executable files exe and dll files that needed in order to run the program.
- Webcam to run the program.
- Special card that determined by developer.
- Windows XP as operating system.

Now we can summaries the **prerequisite** software as following:

- Microsoft Visual Studio.NET 2003 or Visual Studio 6.
- Windows XP.
- DSVideoLib-0.0.4-win32.
- GLUT.
- DirectX 9.0b or later SDK.
- (Optional, for VRML rendered only) OpenVRML-0.14.3-win32

Table 3.1 shows the cost of the requirements needed in order to Build AR applications:

Component	Cost
HP Compaq d230 Microtower Desktop PC	\$1000
USP webcam	\$50
Windows XP	\$299
Visual studio 2003	\$892.95.
VRML	\$214
Total cost	\$2415.95

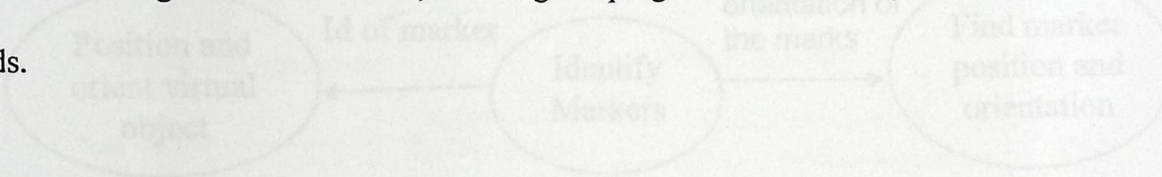
Table3.1 Requirement cost table

CHAPTER Four

Design Augmented Reality Application Using ARToolKit

ARToolKit background:

ARToolKit [17] is a C/C++ language software library that allow programmer to develop augmented reality applications, the most important point of developing Augmented Reality application is precisely calculating the camera position and orientation in real time so that the virtual images are exactly associated with real world objects in the right position. ARToolKit uses computer vision algorithm in order to calculate the position and orientation of the real camera relative to a given marked cards, allowing the programmer to overlay virtual objects onto cards.



History of ARToolKit:

ARToolKit was developed at 1999 by Hirokazo Kato when he arrived at the HITLab.

ARToolKit appeared at SIGGRAPH (short for Special Interest Group for Computer GRAPHics) is the name of the annual conference on computer graphics (CG) convened by the ACM SIGGRAPH organization and ARToolKit was used for shared project space.

ARToolKit support windows direct show video help to enlarge number of users uses this tool. Large modification has been done which include multiplatform, better tracking algorithm.

1. Initialize the application for tracking video capture and read camera and marker parameter.
2. Camera start capturing images of the real world object and send it to the computer.
3. The program searches every video frame for any squares when the position of the camera is known the virtual object is drawn from the same position.
4. If a square is found the software uses some mathematics to calculate the position of the camera relative to the square.

How ARToolKit work?

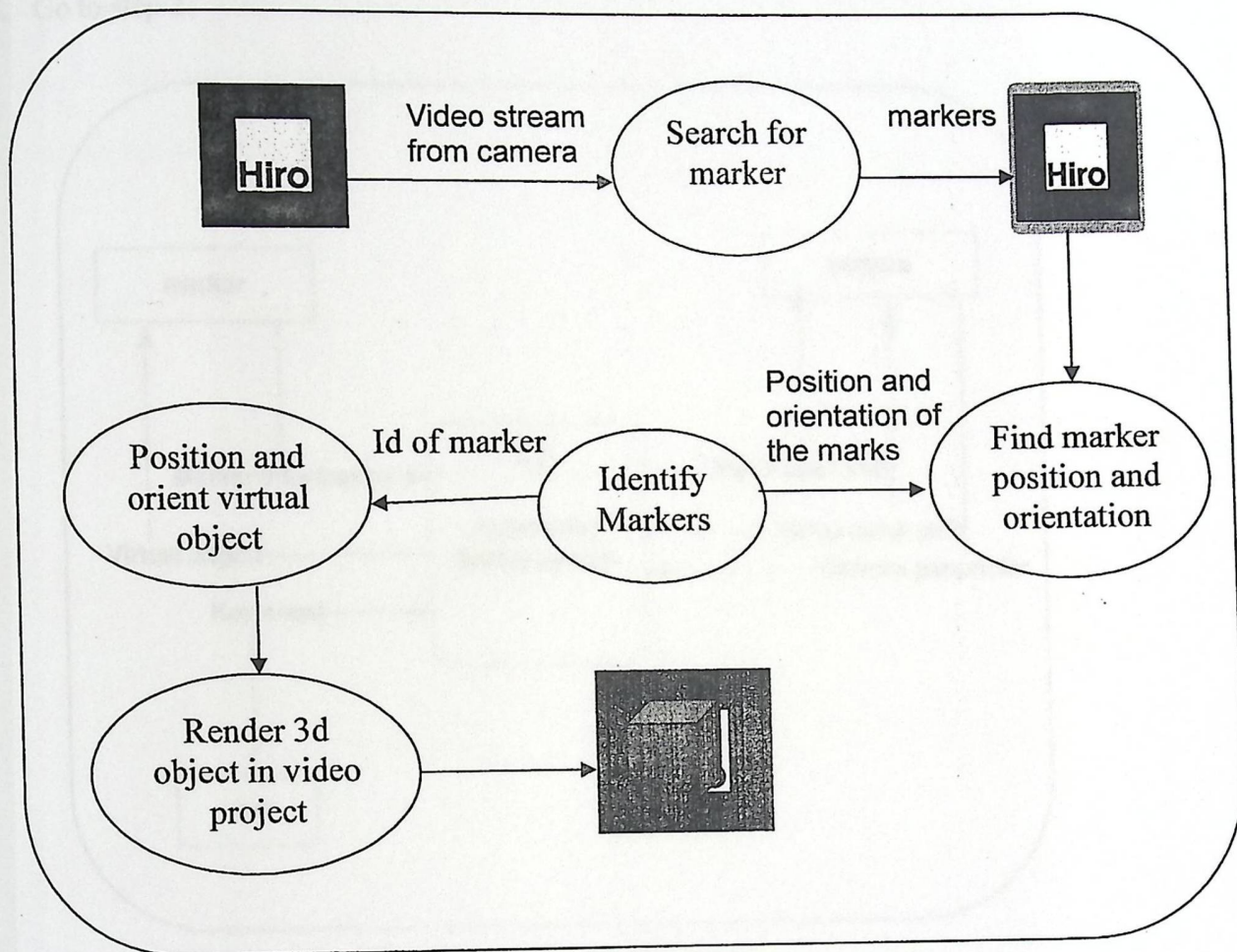


Figure 4.1 ARoolkit general work

In order to overlay virtual object on real time video ARToolkit use tracking marker representative in Black Square, we can clarify this process by steps:

1. Initialize the application for starting video capture and read camera and marker parameter.
2. Camera start capturing images of the real world object and send it to the computer.
3. The program searches every video frame for any squares when the position of the camera is known the virtual object is drawn from the same position.
4. If a square is found the software uses some mathematics to calculate the position of the camera relative to the square.

5. Draws virtual object in video frame.
6. Go to step 2.

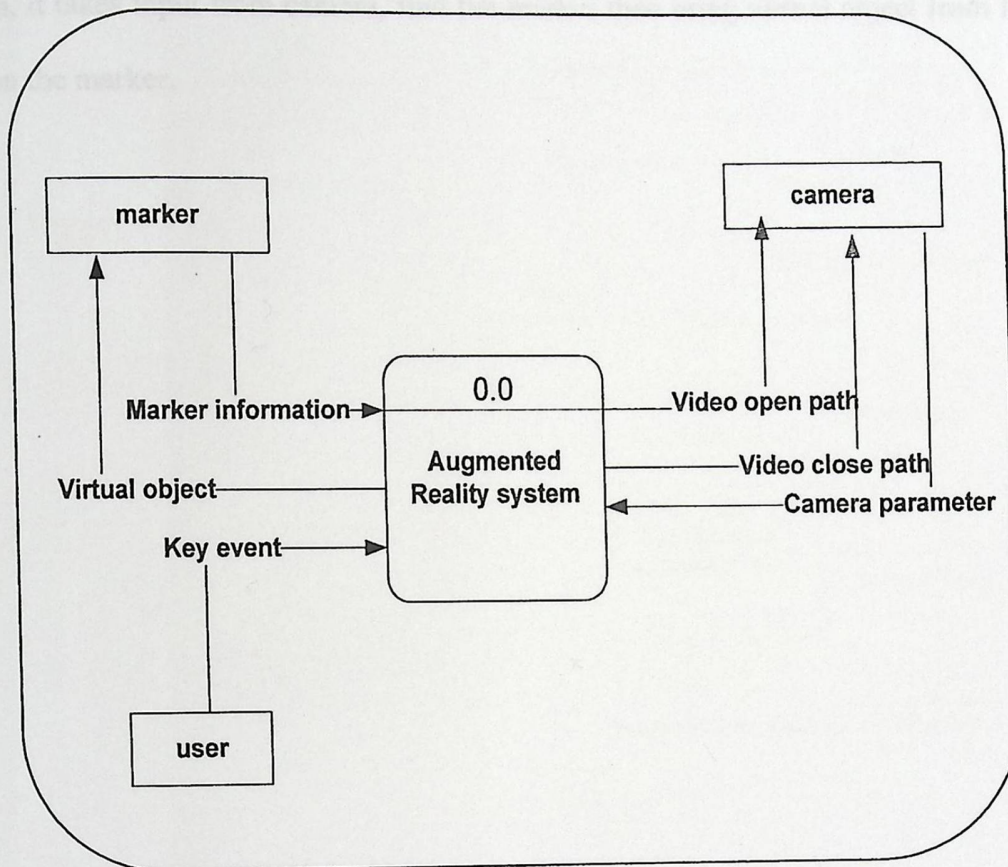


Figure 4.2 Context Diagram

As you can see in Figure 4.2 it is the context diagram with the main functions of our system is to add graphics to real time video, our system deals with user in by taking keyboard event from him, it takes input from camera, find the marker then bring virtual object from file and draw it on the marker.

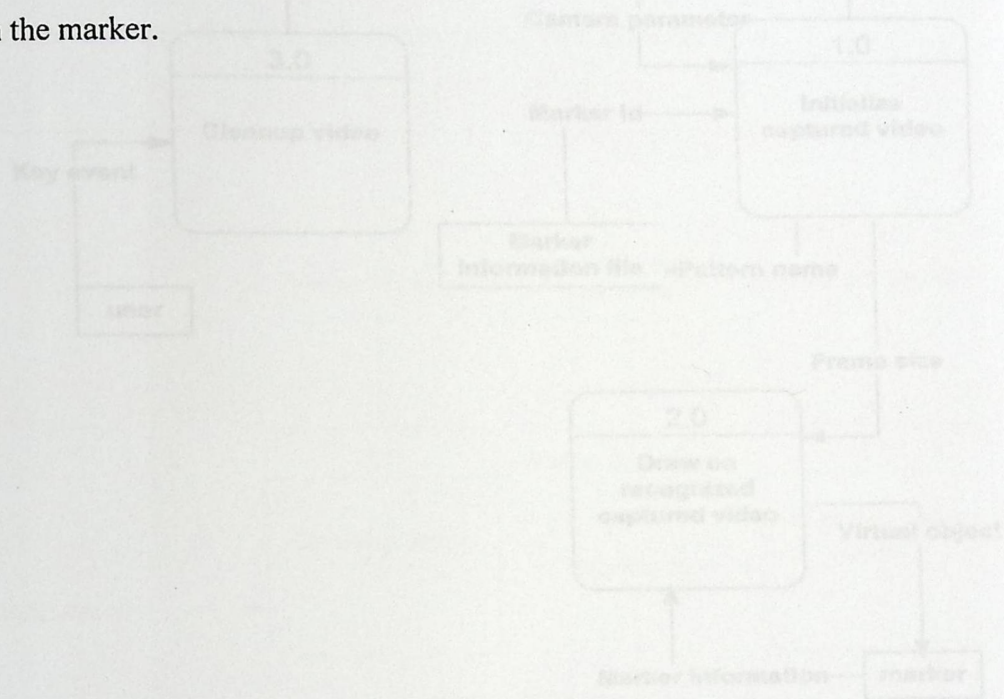


Figure 4.3 DFD level one

The system is divided into subroutines each of them will do specific action, Figure 4.3 illustrates three main subroutines, the first one initializing video capture which opens the video path in order to start capturing the second one is drawing on recognized captured video detect a square and then draw the virtual object on it, the third one is stopping video capturing and end the program.

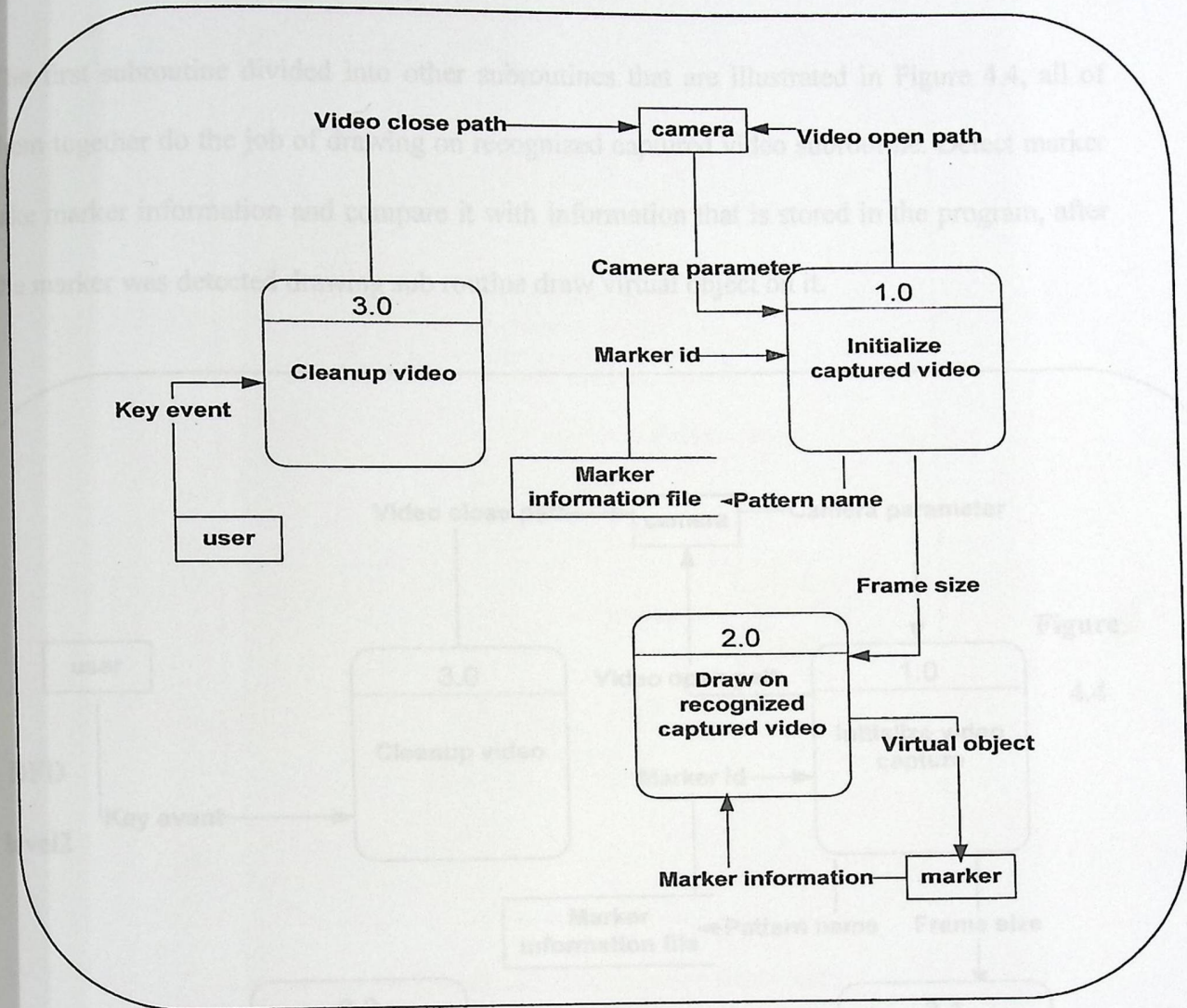


Figure 4.3 DFD level one

The system is divided into subroutine each of them will do specific action; Figure 4.3 illustrates three main subroutines, the first one initializing video capture which opens the video path in order to start capturing, the second one is drawing on recognized captured video detect a square and then draw the virtual object on it, the third one is stopping video capturing and end the program.

The first subroutine divided into other subroutines that are illustrated in Figure 4.4, all of them together do the job of drawing on recognized captured video subroutine. Detect marker take marker information and compare it with information that is stored in the program, after the marker was detected drawing sub routine draw virtual object on it.

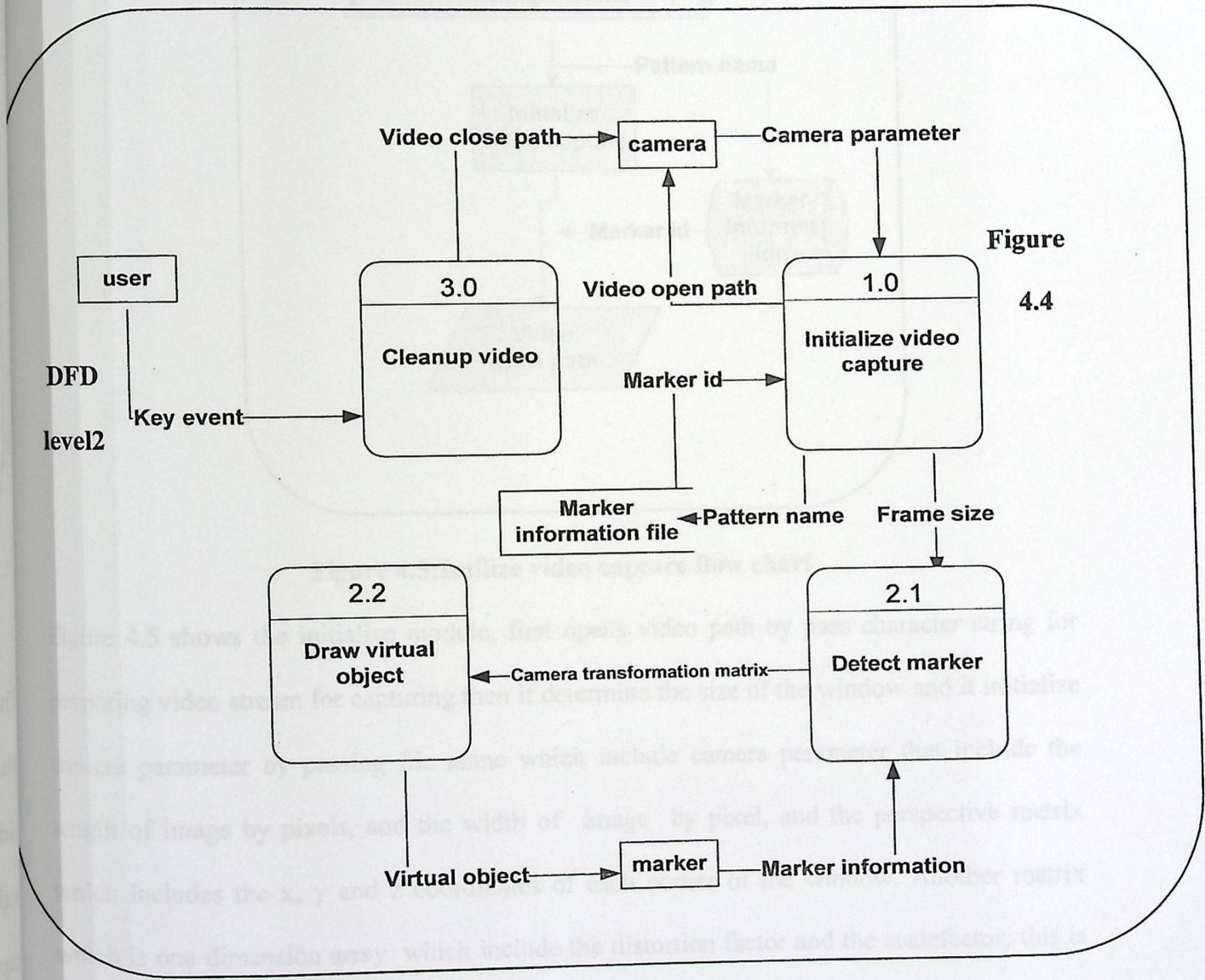


Figure 4.4

Figure 4.4 DFD level2

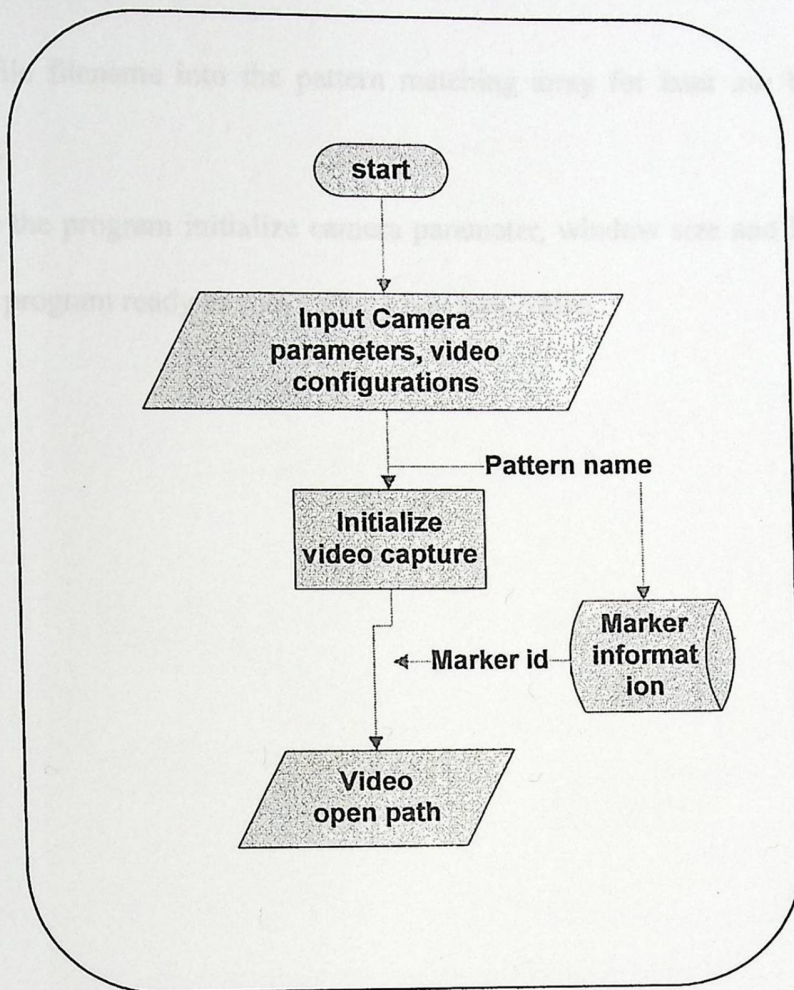


Figure 4.5: Intilize video capture flow chart

Figure 4.5 shows the initialize module, first opens video path by pass character string for preparing video stream for capturing then it determine the size of the window and it initialize camera parameter by passing file name which include camera parameter that include the length of image by pixels, and the width of image by pixel, and the perspective matrix which includes the x, y and z coordinates of each corner of the window. Another matrix which is one dimension array which include the distortion factor and the scalefactor, this is the standard parameter of the camera that must get from the camera parameter file. Next the parameters are transformed for the current image size, because camera parameters change depending on the image size, even if same camera is used then it load the bitmap pattern

specified in the file filename into the pattern matching array for later use by the marker detection routines.

By initial module the program initialize camera parameter, window size and load pattern in this condition the program ready to start capture real time video.

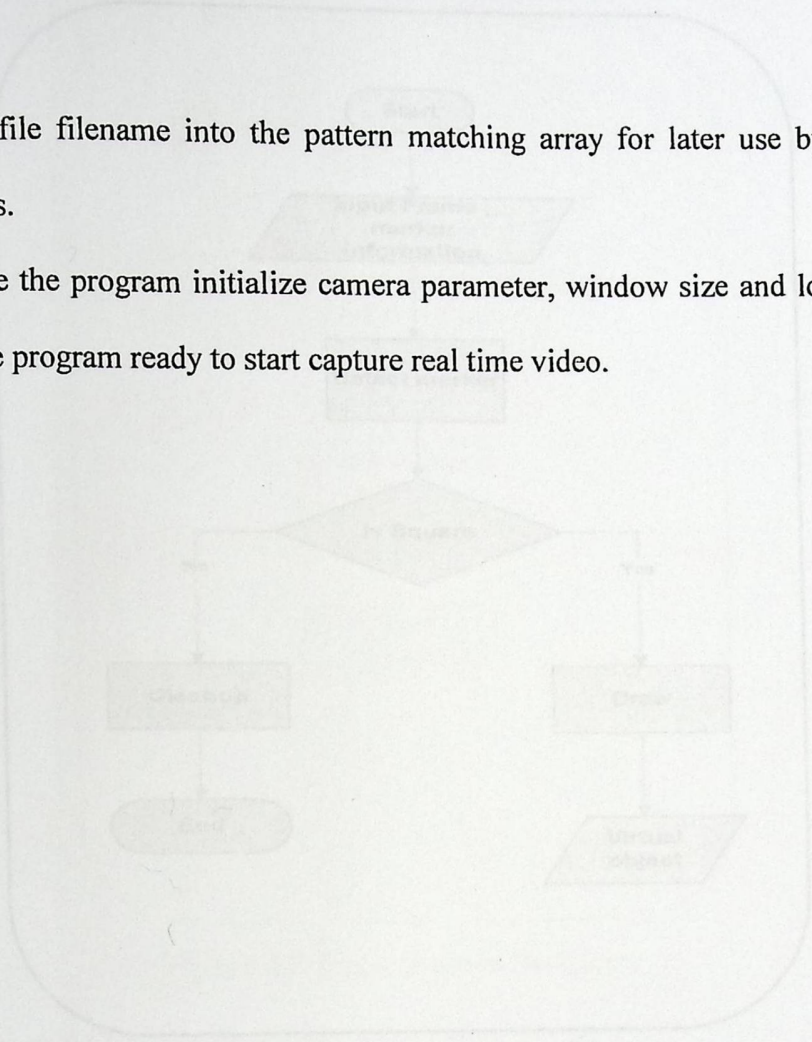


Figure 4.6: Detect marker process flowchart

Figure 4.6 shows detection process flowchart. After capturing real time video the program get video frame and return its pixels in buffer, camera also take marker information from real time images and compare it with the information that exist in the system, after converting real time images into binary images, then the program calculates the right position in the current real video image to determine where the virtual object will draw.

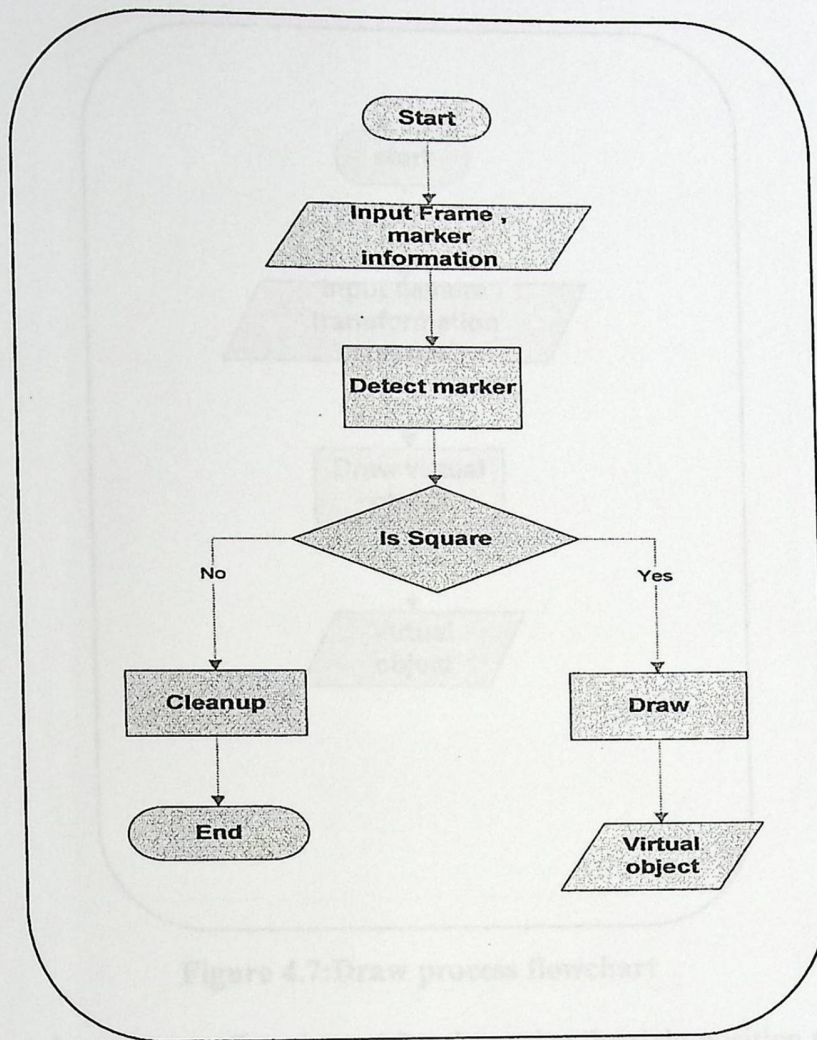


Figure 4.6: Detect marker process flowchart

Figure 4.6 shows detection process flowchart. After capturing real time video the program grab video frame and return its pixels in buffer, camera also take marker information from real time images and compare it with the information that exist in the system, after converting real time images into binary images, then the program calculates the right position to the current real video image to determine where the virtual object will draw.

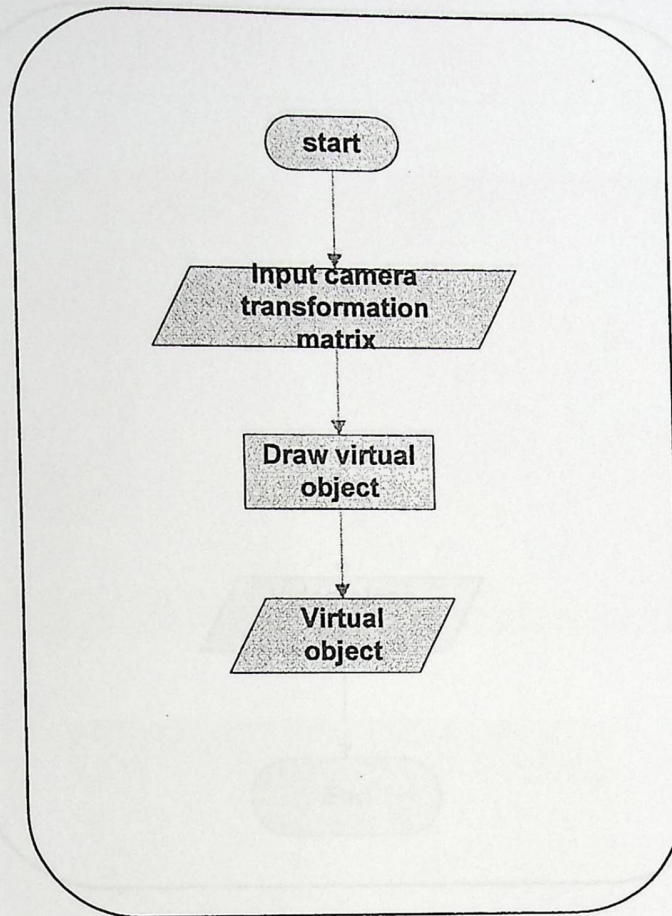


Figure 4.7:Draw process flowchart

Figure 4.7 shows draw process flowchart. After determine the right position the program call draw module to draw the virtual object, which firstly initialize a 3D rendering after that the module will convert the matrix to OpenGL format then the module will render 3D object

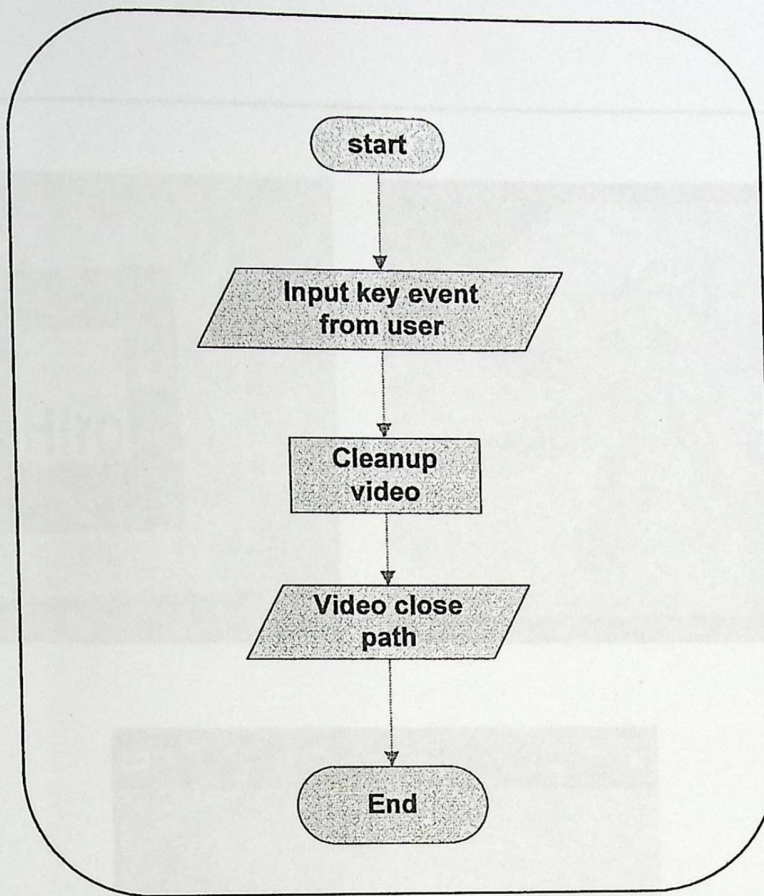


Figure 4.8: Cleanup process flowchart

Figure 4.8 shows the final process in the system, when the user press esc button at this moment the program must stop video capturing, video processing and clean the buffer by close the video path to free it up for other application.

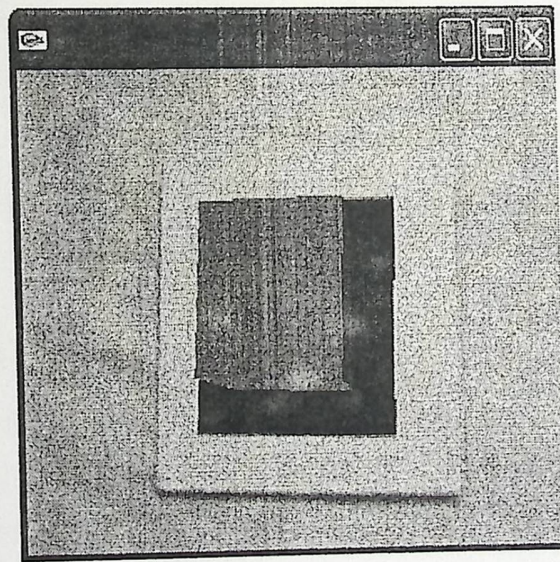
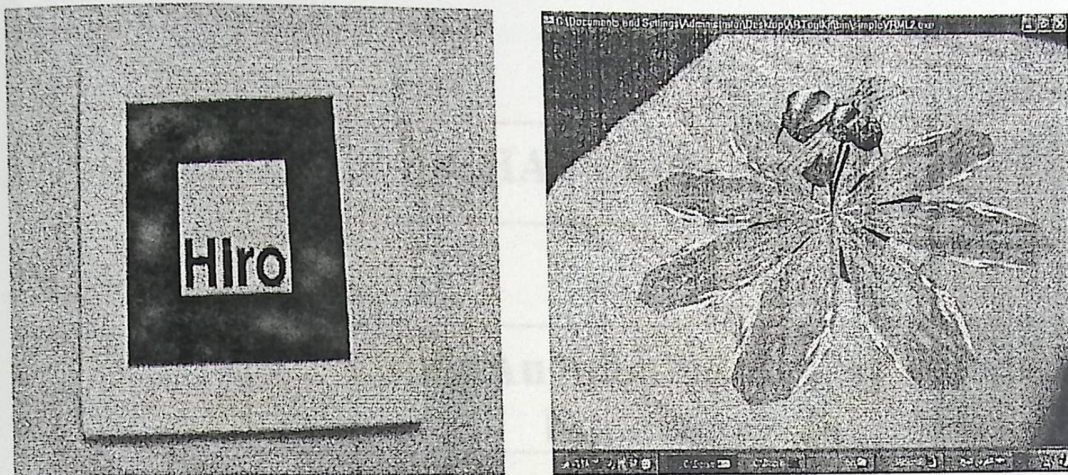


Figure 4.9: Demo for AR using ARToolKit

Figure 4.9 demonstrates two experiments has been made by using ARToolKit, adding a virtual moving bud, and still cub, the figure shows the demo picture before and after overlay the virtual object which appears stuck on it.

CHAPTER Five

Using Augmented Reality in Education

Introduction:

Computer enters our life day by day and become essential part in wide variety area such as: medicine, production, military and education.

Education one of the fields that worth with technology, many researches have been done to support and enhance educational process using different computing paradigm such as: PC era, ubiquitous computing and augmented reality.

Many projects have been done in education applied AR concept cause of students difficulty face in learning complex concepts and phenomena, as a results of this difficulty instructor challenged to find new way for representing the information that beneficial the student in order to examine the advantages of the Augmented Reality interface for viewing and manipulating 3D objects in many application area such as: earth-sun relationships, the Classroom of the Future, Augmented Reality for Geographic Visualization, all the result of these project conclude that using Augmented Reality enhance understanding and increase learnability.

Using Augmented Reality for Teaching Earth-Sun Relationships [22] the experiment was to teach undergraduate geography students about earth-sun relationships applied on thirty students using model designed to teach concepts of Rotation/revolution, solstice/equinox, and seasonal variation of light and temperature.

The result of the experiment instructor found improvement in significant overall improvement in student understanding after the Augmented Reality exercise, as well as a reduction in student misunderstandings. Further analysis implies that learning complex spatial phenomena is depend on how students manipulated with 3D objects, manipulating these 3D objects provide the necessary information needed to make the connections of how earth-sun relationships operate. By bringing Augmented Reality interfaces into classrooms there may be significant benefits to the quality in which curriculum involving complex 3D spatial phenomena and concepts are taught in geography, astronomy and other disciplines.

Also Augmented Reality in education is applied in the classroom of the Future [23] technology promise to enhance learning but not fully exploited Whereas fifty years ago, the only concern a teacher had was running out of chalk, faculty now struggle to perform relatively simple tasks, such as connecting their computer output to the projector, switching the display to show a video tape, and even turning on the lights! Technology potential remains largely untapped.

In response to exploit technology, assist teacher and increase understanding process classroom technology have augmented with various sensors and computer processing The room activates and configures the appropriate equipment in response to instructor activity without the need for manual control thereby freeing the instructor to concentrate on the lecture itself, rather than the user interface.

This project which is attempting to immerse both teacher and learner in a lecture-room environment. The purpose of classroom 2000 is to use automated tools to capture different streams of classroom activities, such as prepared lecture materials, audio, video, and handwritten notes on an electronic surface. The captured material is then integrated together and made accessible via the Web to provide a facsimile of the actual classroom experience.

One of the greatest benefits of Augmented Reality interfaces is that they can be integrated into the existing workplace and combined with other more traditional interface technology as the use of Augmented Reality for Geographic Visualization [24]. Hybrid interfaces that combine technologies such as Augmented Reality, immersive virtual reality and computer-vision methods for natural user input. Project Prototype interfaces are designed to allow multiple users to view and interact with geo-spatial data. Combining Augmented Reality and VR techniques interface have been created where users can seamlessly move between the real and immersive virtual worlds. Combining computer vision hand and object tracking with Augmented Reality techniques. Adding support for pen-based input and 2D image viewing has enabled the user to make annotations on the 3D virtual models.

Augmented Reality one of the field interesting scientists, by day the importance of Augmented Reality has been growing in many fields and in different scope where it can be effectively enhance learning and practical experience and improve interactivity with real

world especially in education. The educational experience offered by Augmented Reality is different for a number of reasons, including:

- Support of seamless interaction between real and virtual environments.
- The use of a tangible interface metaphor for object manipulation.
- The ability to transition smoothly between reality and virtuality.

Augmented Reality in Computer Architecture Course:

Computer architecture is field of study that show the architecture of computer, its component, the processes made inside, and clarify how data move from one place to another through buses, the difficulty of this subject that although the student see component physically as shown in Figures 5.1, 5.2, and 5.3, but they can't realize the processes that have been made inside on the other hand they can't see the flow of data between these component.

In order to facilitate the study, increase learnability, and enhance the reality for computer architecture we used AR in learning computer architecture.

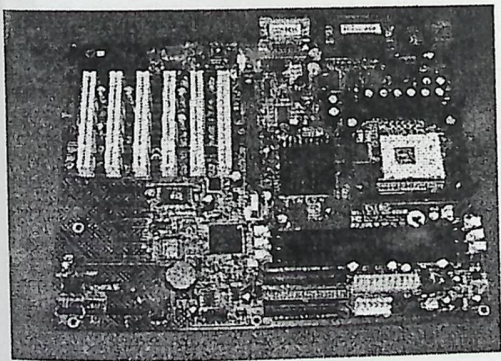


Figure 5.1 Motherboard

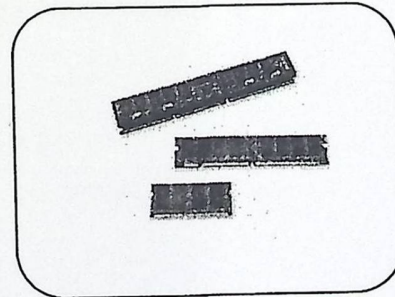


Figure 5.2 RAM

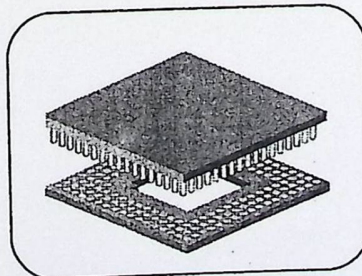


Figure 5.3 CPU

Description of our experiment:

Our project examines the advantages of the AR interface for viewing and manipulating 3D objects. We want to know how students' understandings of spatial content change through their physical interactions with virtual objects. Our hypothesis is that AR changes the way students come to understand certain concepts. The results will demonstrate the potential benefits of using AR visualization interfaces in education and training.

We used VRML to build our virtual object which can be used to clarify component in the computer architecture in order to overlay these virtual objects onto the real objects in real time. We used pattern that the program can recognize it and put on it the virtual object.

We pasted our natural pattern which is the real component with white paper around and inside it, on the component we wanted to clarify it, and Videotaping the computer from inside via webcam in order to display real component stuck with virtual object that clarify it.

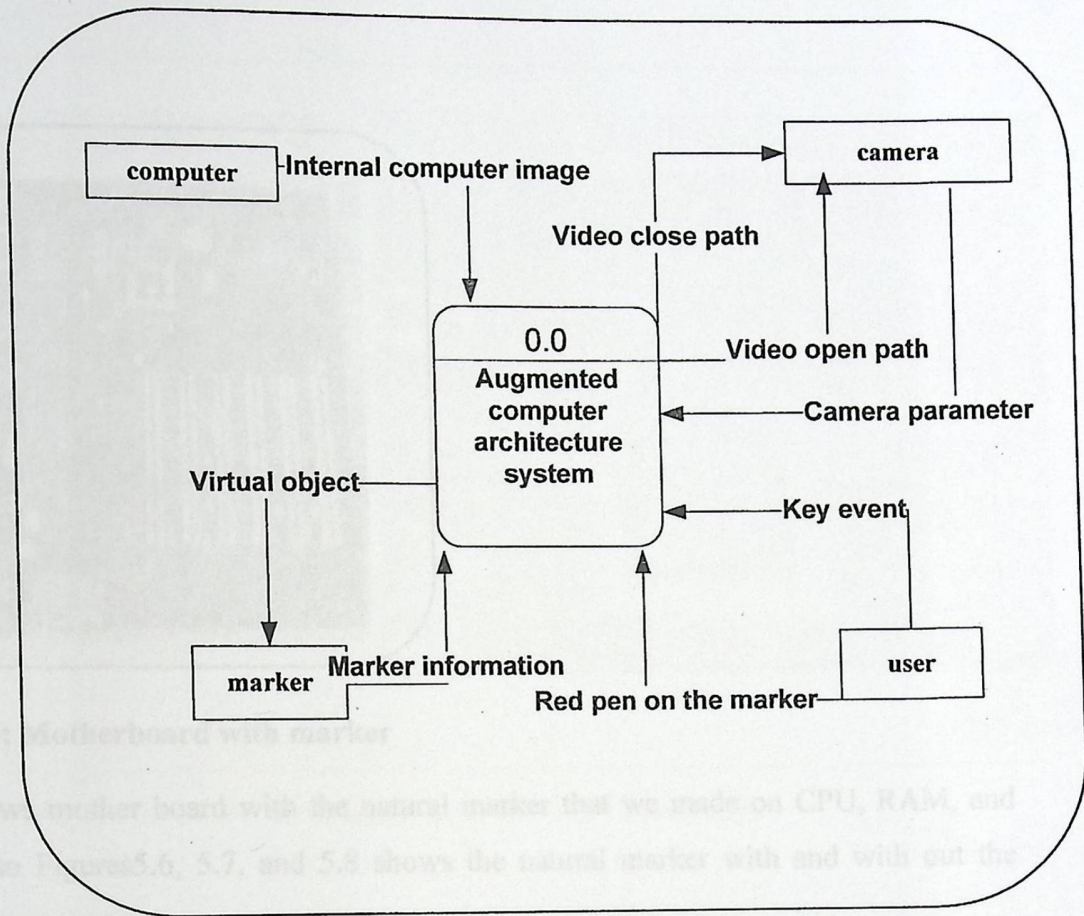


Figure 5.4: Context diagram for Interactive Augmented Reality in Computer Architecture

As shown in Figure 5:4 the program take camera parameter and internal image of computer that the experiment done on it then the augmented reality program put the VRML object on it.

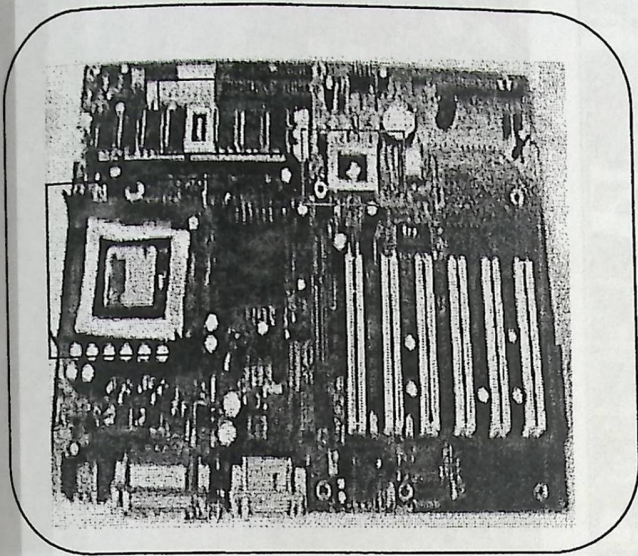


Figure 5.5: Motherboard with marker

Figure 5.5 shows mother board with the natural marker that we made on CPU, RAM, and MainChip. Also Figures 5.6, 5.7, and 5.8 shows the natural marker with and with out the virtual object.

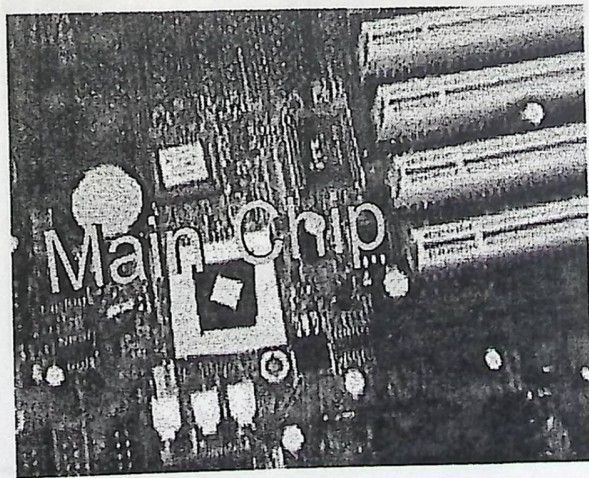
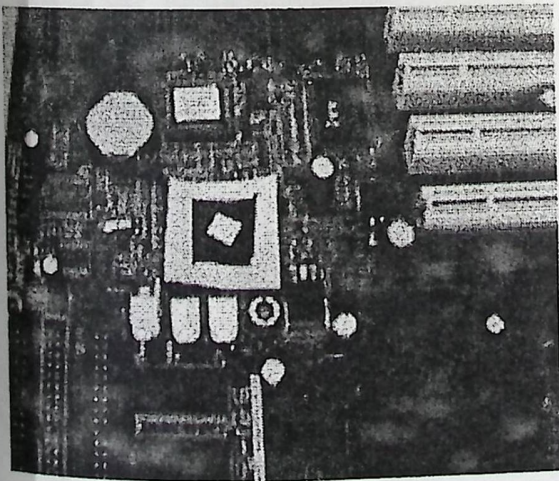


Figure 5.6: MainChip before and after adding virtual object.

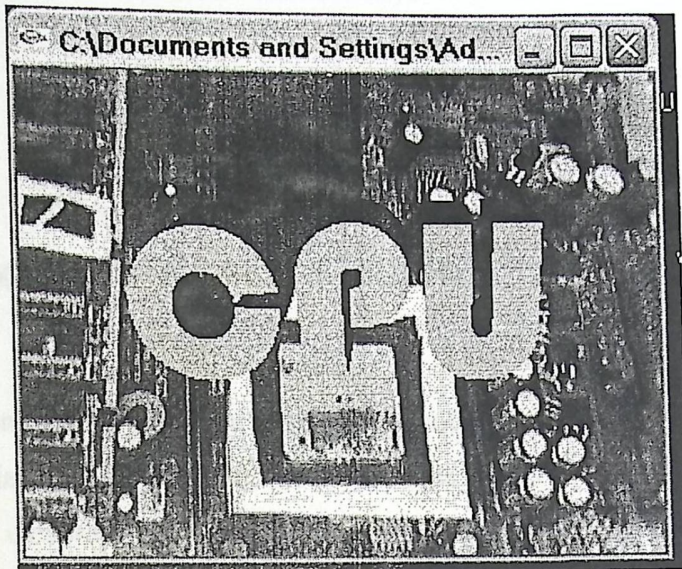
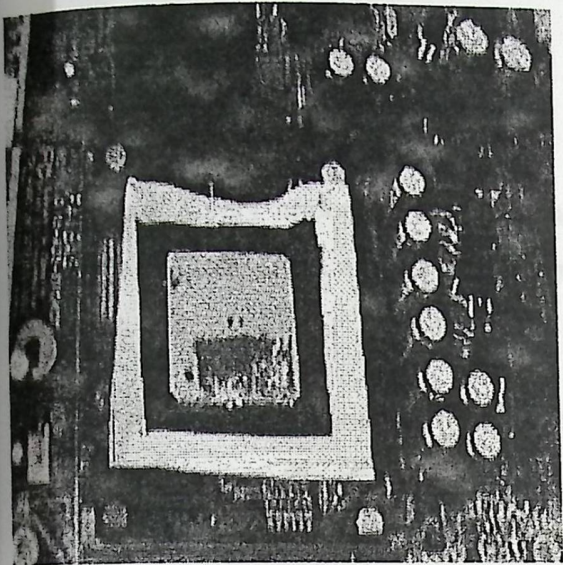


Figure 5.7: CPU before and after adding virtual object

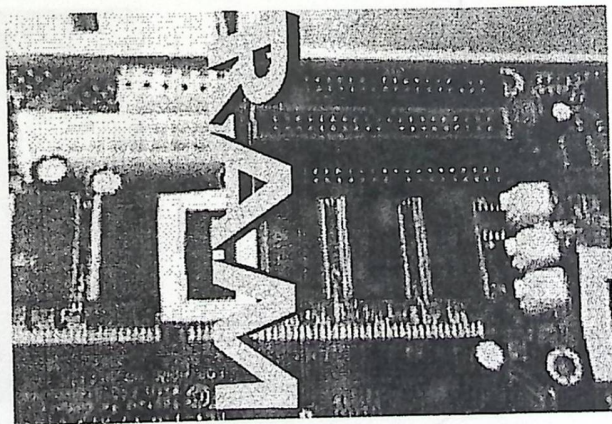
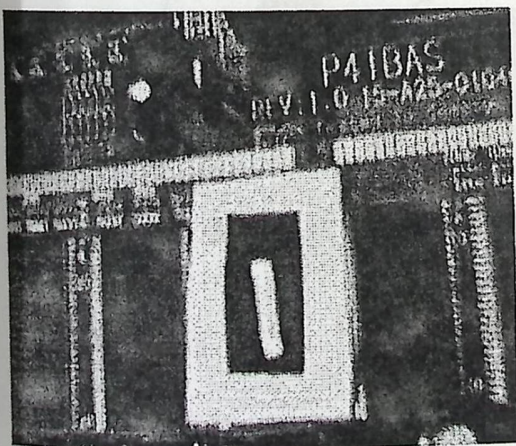


Figure 5.8: RAM marker before and after adding virtual object

Interactive Augmented Reality:

Color Models:

Color plays a very important role in image analysis when used as a descriptor. A full color image, also known as a True Color image, has three different bands (R=red, G=green, B=blue), one per color.

A color model is a 3D unique representation of a color. There are different color models and the use of one over the other is problem oriented. For instance, the color model RGB is used in hardware applications like PC monitors, cameras and scanners, the CMYK color model is used in color printers, and the YIQ model in television broadcast. In color image manipulation the two models widely used are HSI and HSV.

RGB

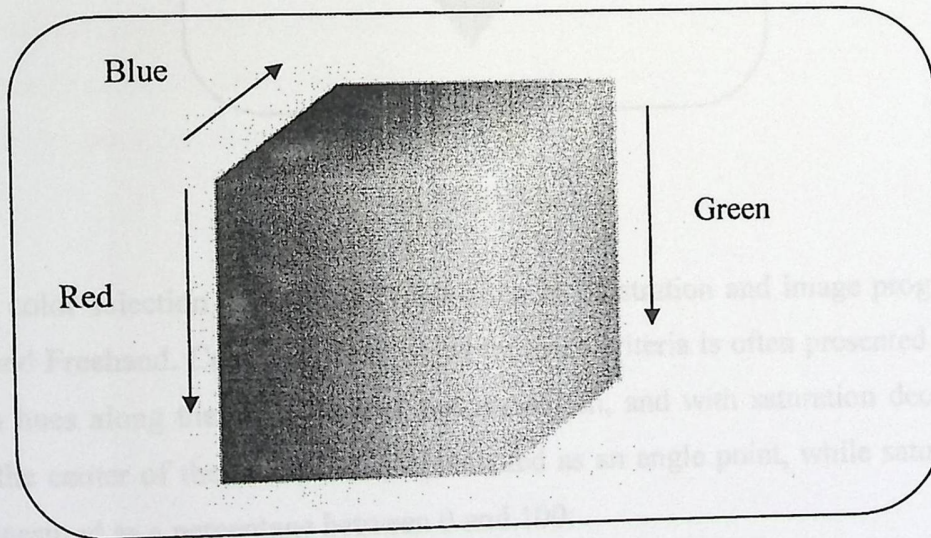


Figure 5.8 RGB Model

Figure 5.8 shows RGB model which stands for red, green, blue—the three colors that make up an image on a monitor. The RGB color model is an additive model used for displaying images on a computer monitor or other screen device. When the three primary colors, red, green, and blue, are combined they make a white light. When all are absent, there is black. Projected colors often feel much brighter than their subtractive model counterparts.

RGB drawbacks that we face it in our project:

The biggest problem with RGB its sensitivity to different lighting condition and that cause the detection process very difficult.

HSV Model

Figure 5.9 shows hue, saturation, and value combine to form the HSV color model.

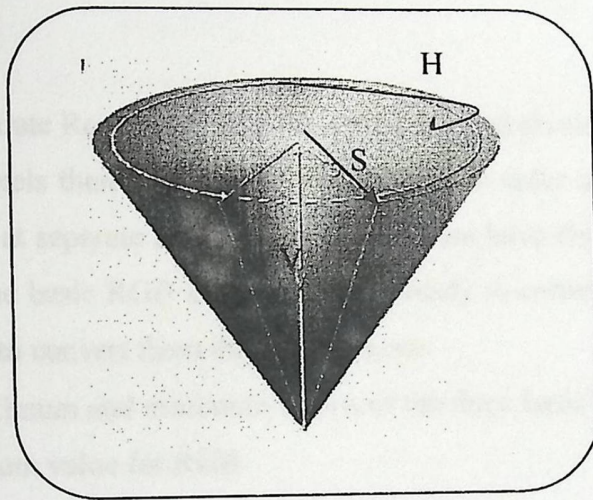


Figure5.9HSVModel

HSV is the color selection model used most often in illustration and image programs, like Fireworks and Freehand. Color selection based on these criteria is often presented as a color wheel, with hues along the outer edge at full saturation, and with saturation decreasing as moving to the center of the circle. Hue is presented as an angle point, while saturation and value are measured as a percentage between 0 and 100:

To create HSV colors on the computer you select a hue and saturation on the color cone. Hues are around the edge of the cone, saturation increases with the distance from the center.

Until recently Augmented Reality presented a passive interface to its users, who merely just a viewer of the visual representation, now a days Augmented Reality provide us with an interactive environment.

In our project we used our abilities in image processing to build our interactive application; firstly we capture real time images by using USB camera, then convert images from RGB model into HSV model then we take colors in specific range (threshold), in order to take the red color only and make detection based on the center of the red pen, then make the action that supposed to be done.

To do that we firstly separate Red, Green and Blue pixel by read pixels and store first pixel at separate array for red pixels then read second pixel for green color and read third pixel for blue color each of them at separate array at this moment we have three arrays each of them represent one color of the basic RGB color model are ready to convert them to HSV model for that we use equation to convert them this equation are:

Firstly we must find maximum and minimum values of the three basic color values by:

Calculate maximum value for RGB

Calculate minimum value for RGB

Secondly we compute the Hue, Saturation and Value values see Algorithm 5.1.

```
V=Max
S= (Max-Min)/Max
Loop for Hue value
  Max equal Min
  Undefined
Else If Max equal R and G more than or equal B
  H=60*(G-B)/ (Max-Min)
Else If Max equal R and G grater than B then
  H=60*(G-B)/ (Max-Min) +360
  Else If Max equal
  H=60*(B-R)/ (Max-Min) +120
  Else If Max equal B then
  H=60*(R-G)/ (Max-Min) +240
```

Algorithm 5.1 RGB to HSV

After that we are ready to get Red color by take color in specific range which will give us each object have red color to do that we threshold each Hue between 0.3 and 0.8 value and Saturation between 0.95 and 1 then each value of them locate in it's range we give it zero value then we emerge Hue and Saturation value in each pixel when both of them have zero value then each object have red color will appears in black color and another color will be appear in white color

Designing interactive augmented reality program:

To interact with virtual object we use image processing abilities to detect the red pen that used to interact with virtual object. HSV color model used to detect red pen.

Hence the processing takes time to convert color to HSV model, also slow the program and the result of this problems influence the education process and immersion in the system we construct lookup table to overcome these problems.

Lookup table is three dimensional arrays that contain all potential value for R, G and B value then calculate hue and saturation for each potential then store each of them in specific array or pointer.

Instead of calculate hue and saturation value in the real time the program search lookup table for the RGB color that has been taken in real time for the hue and saturation value corresponding to it, so the result of this operation speed up the program and make the experiment more realistic and as a result increase immersion of the learner in the system.

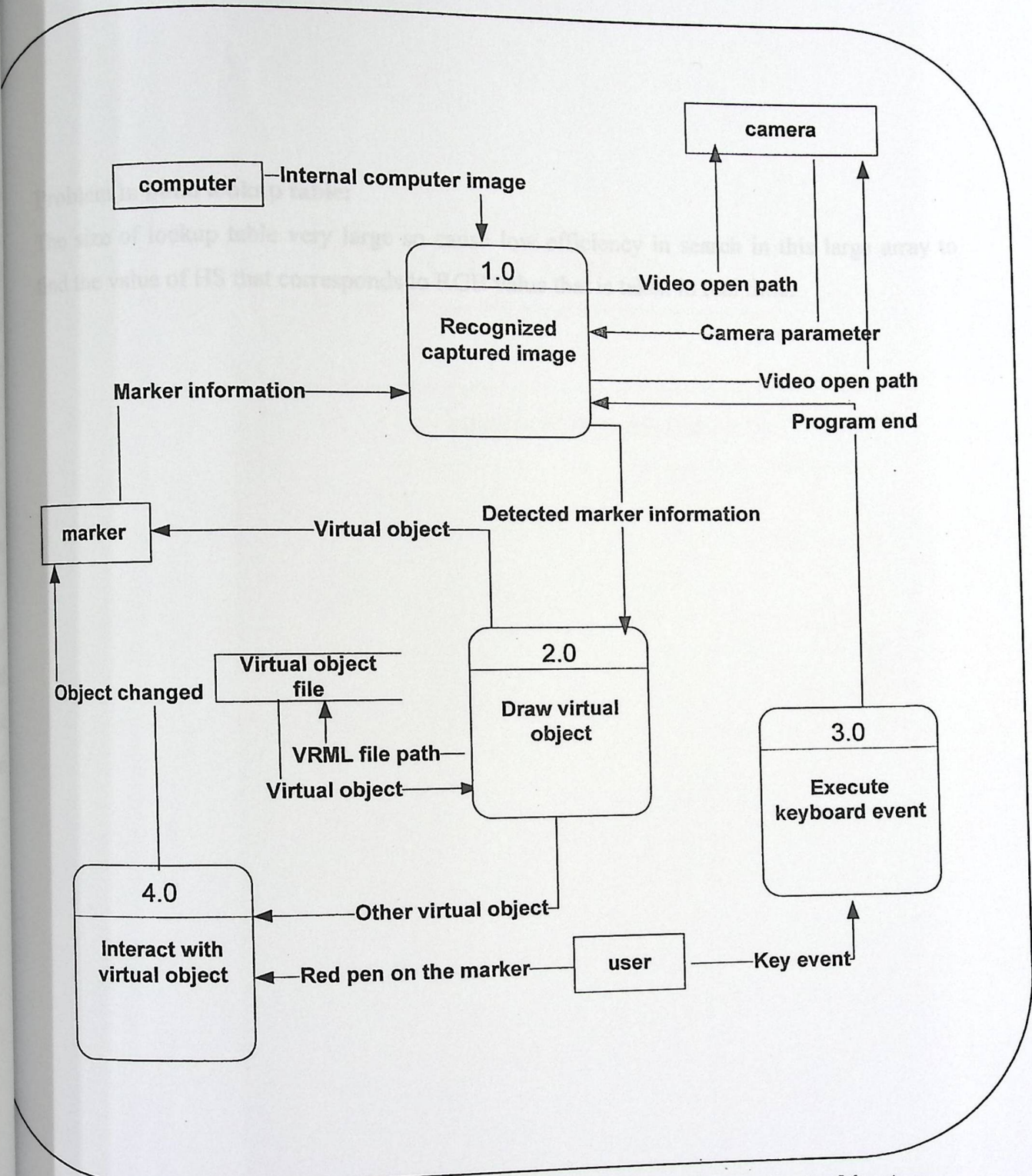


Figure 5.11: DFD level 1 Interactive Augmented Reality in Computer Architecture

As shown in Figure 5.11 the program the user interact with the virtual object via red pen then the program make action such as render another virtual object.

Problem in build lookup table:

The size of lookup table very large so cause low efficiency in search in this large array to find the value of HS that corresponds to RGB value that is taken in real time.

CHAPTER Six

Implementation

In this chapter we will classify each function and explain what each function does with
in general AR Toolkit it work as these steps:

1. Initialize the application for starting video capture and read camera and marker parameter.
2. Camera start to capture video and send it to the computer.
3. The program searches every video frame for the position of the camera is known the virtual object is drawn from the same position.
4. If the square is found the software uses some mathematics to calculate the position of the camera relative to the black square.
5. Draw virtual object in video frame.
6. Close the video capture.

The previous process give general idea how AR Toolkit work, but in details there are several
routine in ARToolkit but the most important functions are main, init, mainloop,
draw and cleanup.

We will clarify these function and subroutine included in every function by walkthrough of
the SimpleTest example.

```
main :  
  
The main routine of simple is shown below:  
  
main(int argc, char *argv[])  
{  
    init();  
    VideoCapStart();  
    ArgMainLoop(NULL, keyEvent, mainloop);  
}
```

In this chapter we will classify each function and explain what each function does with details. In general ARToolkit work at these steps:

1. Initialize the application for starting video capture and read camera and marker parameter.
2. Camera start to capture images of the real world object and send it to the computer,
3. The program searches every video frame for any square when the position of the camera is known the virtual object is drawn from the same position.
4. If the square is found the software uses some mathematics to calculate the position of the camera relative to the black square.
5. Draw virtual object in video frame.
6. Close the video capture

The previous process give general idea how ARToolkit work, but in details there are several routine in ARToolkit but the most important functions are `main`, `init`, `mainloop`, `draw` and `cleanup`.

We will clarify these function and subroutine included in every function by walkthrough of the SimpleTest example.

Main :

The main routine of simple is shown below:

```
main (int argc, char *argv[])
{
    init ();
    arVideoCapStart ();
    argMainLoop ( NULL, keyEvent, mainLoop );
}
```


The main steps of the main routine:

`Init ()` is the initialize function for initialize the camera parameter and loading marker file to start video capture.

`arVideoCapStart ()` the main purpose from this function to capture video at the real time.

`argMainLoop ()` start the main function of the program by called the mainloop function.

Initialization function:

```
static void init( void )
{
    ARParam wparam;

    /* open the video path */
    if( arVideoOpen( vconf ) < 0 ) exit(0);
    /* find the size of the window */
    if( arVideoInqSize(&xsize, &ysize) < 0 ) exit(0);
    printf("Image size (x, y) = (%d, %d)\n", xsize, ysize);

    /* set the initial camera parameters */
    if( arParamLoad(cparam_name, 1, &wparam) < 0 ) {
        printf("Camera parameter load error !!\n");
        exit(0);
    }
    arParamChangeSize( &wparam, xsize, ysize, &cparam );
    arInitCparam( &cparam );
    printf("*** Camera Parameter ***\n");
    arParamDisp( &cparam );

    if( (patt_id=arLoadPatt(patt_name)) < 0 ) {
        printf("pattern load error !!\n");
        exit(0);
    }

    /* open the graphics window */
    argInit( &cparam, 1.0, 0, 0, 0, 0 );
}
```

Initialize function starts capturing process, reading in the marker and camera parameters, and setup of the graphics window and each task do by specific subroutines.

Firstly open video path by passing variable name `vconf` refer to character string encapsulates operating-system dependent directions as to how to find, open and configure the video stream if it is successful return 0, if can't open the path return -1 `arVideoOpen(vconf)` but the capture will not start until called the `arVideoCapStart`.

the program must find the size of window to the captured video frame by pixel where is the `xsize` pointer to the length of the window while the `ysize` is pointer to the width of frame, the function return 0 if the size of the frame determined otherwise returned -1 this subroutine may have `arVideoInqSize()` name.

After that the program must have function which load camera parameter by pass the name of camera parameter file which loaded that may have `cparam_name` name and pass integer number of parameter that pass by function without the first parameter and the number itself, `wparam` pointer to the camera parameter struct which include `xsize` parameter is the length of image by pixels, `ysize` parameter is the width of image by pixel, both `xsize` and `ysize` are integer type, another parameter in the struct is the perspective matrix is two dimension which include the `x`, `y` and `z` coordinates in each corner of the window and it have another matrix which called `dist_factor` it is one dimension array which include the distortion factor and the scale factor, this is the standard parameter of the camera that must get from the camera parameter file, the function return 0 if success and -1 If failed (`arParamLoad`).

Next the parameters are transformed for the current image size, because camera parameters change depending on the image size, even if same camera is used through function (`arParamChangeSize()`) to do that we must pass the source struct of camera parameter which called `cparam_name`, `xsize` and `ysize` that are integer type and they represent the width and height of the window and another parameter, will be pass to the previous function, have the same type of source camera parameter struct to hold the new parameter, next initialize camera parameters by setting the camera parameters specified in the camera parameters structure to static memory in the AR library these camera parameters are typically read from a data file at program startup this function may called `arInitCparam`, after that, program load the pattern by pass pattern name as parameter to function may called `arLoadPatt`, this parameter is the name of the file that containing the pattern bitmap to be loaded and search for a square and compare it with pattern id this function return integer value which represent id number of the pattern if it load pattern successfully else return -1, finally the function open graphics window (`argInit()`) this function performs required initialisation of the gsub library and have six arguments which are struct of the camera parameter, a zoom parameter for the result that is double type, flag have integer value determine if full screen mode enable or disable, `xsize` and `ysize` of the window finally argument represent the flag to enable stereo display mode or disable.

arVideoCapStart function:

This function start capturing routine for video processing, so this function start thread that begin capturing process by calling it and is coupled with `arVideoCapStop` that can be called

many times in program return zero if start capturing successfully else will return -1 if the capturing could'n start.

argMainloop function:

This function start main loop with specified callback function, it take three argument mouse function can be null, key function which can be also null and main user update can be null. The argMainLoop function is called which starts the main program loop and associates the function key Event with any keyboard events and main Loop with the main graphics rendering loop, the we will clarify each argument with details:

keyEvent function:

```
static void keyEvent( unsigned char key, int x, int y)
{
    /* quit if the ESC key is pressed */
    if( key == 0x1b ) {
        printf("*** %f (frame/sec)\n", (double)count/arUtilTimer());
        cleanup();
        exit(0);
    }
}
```

This function have three argument firstly key argument is unsigned char that represent the button that pressed by user by take the value that represent the key of keyboard then the function compare this aegument by value equal ESC button if both are equale then the program will quit and call cleanup function.

Mainloop function:

```
static void mainLoop(void)
{
```

```

ARUint8      *dataPtr;
ARMarkerInfo *marker_info;
int          marker_num;
int          j, k;

/* grab a vide frame */
if( dataPtr = (ARUint8 *)arVideoGetImage() == NULL ) {
    arUtilSleep(2);
    return;
}
if( count == 0 ) arUtilTimerReset();
count++;

argDrawMode2D();
argDisplImage( dataPtr, 0, 0 );

/* detect the markers in the video frame */
if( arDetectMarker(dataPtr, thresh, &marker_info, &marker_num) < 0 ) {
    cleanup();
    exit(0);
}

arVideoCapNext();

/* check for object visibility */
k = -1;
for( j = 0; j < marker_num; j++ ) {
    if( patt_id == marker_info[j].id ) {
        if( k == -1 ) k = j;
        else if( marker_info[k].cf < marker_info[j].cf ) k = j;
    }
}

if( k == -1 ) {
    argSwapBuffers();
    return;
}

/* get the transformation between the marker and the real camera */
arGetTransMat(&marker_info[k], patt_center, patt_width, patt_trans);

printf("%f %f %f\n", patt_trans[0][3], patt_trans[1][3], patt_trans[2][3]);

```

```

draw();

argSwapBuffers();
}

```

Main loop is the most important part of the program because in this part we will find the main steps of the program. Firstly, there is some important declaration like `dataPtr` which is pointer to the captured video, another important variable is `struct` that hold the detected marker information that is `ARMarkerInfo` struct which include `area` which is integer value that represent number of pixels in the labeled region, `id` which is integer value that hold specific identity number for each marker, `dir` Direction is integer value that tells about the rotation about the marker, `cf` confidence is double value that determine how it probably to be a marker, `pos` is array of double that hold center of marker in ideal screen coordinates, `line` is two dimensional array of double type and is equations for four side of the marker (in ideal screen coordinates). Last argument, `vertex` is two dimensional array of double type edge points of the marker (in ideal screen coordinates).

First image captured through the function `arVideoGetImage ()` This function returns a buffer with a captured video image. The returned data consists of a tightly-packed array of pixels, beginning with the first component of the leftmost pixel of the topmost row, and continuing with the remaining components of that pixel, followed by the remaining pixels in the topmost row, followed by the leftmost pixel of the second row. The arrangement of components of the pixels in the buffer is determined by the configuration string passed in to the driver at the time the video stream was opened. If no pixel format was specified in the configuration string, then an operating system dependent default, defined in `<AR/config.h>` is used.

The memory occupied by the pixel data is owned by the video driver and should not be freed by your program. The pixels in the buffer remain valid until the next call to `arVideoCapNext`, or the next call to `arVideoGetImage` which returns a non-NULL pointer, or any call to `arVideoCapStop` or `arVideoClose`. If the result equal null that mean that there is no image captured so call function `arUtilSleep(2)` to wait 2 milisecond as you detrmine, then if count equal zero reset internal timer of ARToolkit and increse count by one, after that `argDrawMode2D()` is called to switch the rendering context for 2D rendering mode by update current camera parameter for rendering 2D or 3D objects in the view plane, next `argDispImage()` called to Display in the back-buffer the video image in argument which mean `dataptr`. This function must be called before any rendering of 3D object follow this detection process done by `arDetectMarker(dataPtr, thresh, &marker_info, &marker_num)` this is the main function to detect the square markers in the video input frame, it take four parameter `dataptr` a pointer to the color image which is to be searched for square markers, threshold specifies the threshold value (between 0-255) to be used to convert the input image into a binary image, `marker_info` a pointer to an array of `ARMarkerInfo` structures returned which contain all the information about the detected squares in the image, param `marker_num` the number of detected markers in the image, the function return 0 when the function completes normally, -1 otherwise if there is no square marker is found and call cleanup function, `arVideoCapNext()` function that called at least of each frame It has several purposes, depending on the operating system. It allows the video driver to perform housekeeping task and also signals to the video grabber that your code has finished using the most recent video frame returned by `arVideoGetImage()`, and that the video driver may re-use the memory occupied by the frame, then check the

object visibility by entering loop to compare the confidence value of the marker found with another confidence value of another marker also found to determine the best marker, the meaning of confidence value is the probability to be marker, next if marker found `arGetTransMat(&marker_info[k], patt_center, patt_width, patt_trans)` called to compute camera position in function of detected markers thus to calculate the transformation between a detected marker and the real camera, parameter `marker_info` the structure containing the parameters for the marker for which the camera position and orientation is to be found relative to. This structure is found using `arDetectMarker`. parameter `center` the physical center of the marker. `arGetTransMat` assumes that the marker is in x-y flat surface, and z axis is pointing downwards from marker flat surface. So vertex positions can be represented in 2D coordinates by ignoring the z axis information. The marker vertices are specified in order of clockwise. parameter `width` the size of the marker (in mm). parameter `conv` the transformation matrix from the marker coordinates to camera coordinate frame, that is the relative position of real camera to the real marker return always 0 then draw function is called to draw virtual object on the card.

Draw Function :

```
static void draw( void )
{
    double gl_para[16];
    GLfloat mat_ambient[] = {0.0, 0.0, 1.0, 1.0}; [18]
    GLfloat mat_flash[] = {0.0, 0.0, 1.0, 1.0};
    GLfloat mat_flash_shiny[] = {50.0};
    GLfloat light_position[] = {100.0, -200.0, 200.0, 0.0}; [19]
    GLfloat ambi[] = {0.1, 0.1, 0.1, 0.1};
    GLfloat lightZeroColor[] = {0.9, 0.9, 0.9, 0.1}; [20]

    argDrawMode3D();
    argDraw3dCamera( 0, 0 );
    glClearDepth( 1.0 );
    glClear(GL_DEPTH_BUFFER_BIT);
```



```

glEnable(GL_DEPTH_TEST);
glDepthFunc(GL_LEQUAL);

/* load the camera transformation matrix */
argConvGpara(patt_trans, gl_para);
glMatrixMode(GL_MODELVIEW);
glLoadMatrixd( gl_para );

glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glLightfv(GL_LIGHT0, GL_AMBIENT, ambi);
glLightfv(GL_LIGHT0, GL_DIFFUSE, lightZeroColor);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_flash);
glMaterialfv(GL_FRONT, GL_SHININESS, mat_flash_shiny);
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
glMatrixMode(GL_MODELVIEW);
glTranslatef( 0.0, 0.0, 25.0 );
glutSolidCube(50.0);
glDisable( GL_LIGHTING );

glDisable( GL_DEPTH_TEST );
}

```

The draw functions is divided in initialize rendering, setup matrix, render object firstly, it have important variables like `gl_para` it is one dimension array of double have sixteen values are the position and orientation values of the real camera and there are another arrays that specific for display mode like shiny, ambient and colors for OpenGL functions after that the program must Update curent camera parameters for rendering in 3D space. Generally call to reinializing model view matrix. To complement rendering view for 3D rendering mode the program pass `xwin` and `ywin`, both have integer value for rendering window, to Update curent internal camera parameters by `argDraw3dCamera()`. Then to specifies the clear value for the depth buffer we will pass Depth parameter which Specifies the depth value used when the depth buffer is cleared. The default value is 0 (zero) for `glClearDepth()`. then to sets the bitplane area of the window to values previously selected by `glClearDepth()` it

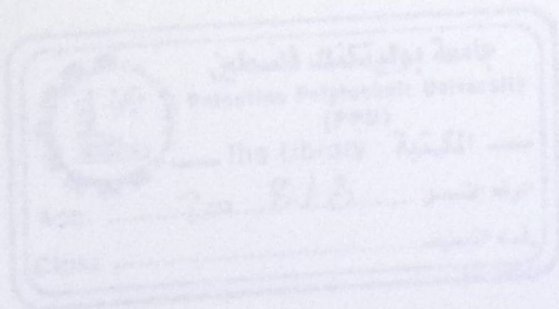
pass mask, which Bitwise OR of masks that indicate the buffers to be cleared this mask is `GL_DEPTH_BUFFER_BIT` which Indicates the depth buffer, to `glClear()`. Then to enable server-side GL capabilities we pass `cap` parameter, which Specifies a symbolic constant indicating a GL capability in specific the capability is `GL_DEPTH_TEST` If enabled, do depth comparisons and update the depth buffer. Note that even if the depth buffer exists and the depth mask is non-zero, the depth buffer is not updated if the depth test is disabled, to `glEnable()`. then we must specify the value used for depth buffer comparisons to do that we must pass `GL_LEQUAL` paramter, which Passes if the incoming depth value is less than or equal to the stored depth value, to `glDepthFunc()` so now we can compare each incoming pixel depth value with the depth value present in the depth buffer, to `glDepthFunc` The comparison is performed only if depth testing is enabled that what we are do in the previous function, after that we must transform ARToolkit matrix format to an openGL matrix format to do that we call `argConvGlpara()` function with param `para` the ARToolkit matrix param `gl_para` the resulted openGL matrix. that `argSwapBuffers()` called swap the rendering buffer, Swap the back-buffer to the front-buffer. The pre-condition is that all the rendering functions have been called.

Cleanup function:

```
static void cleanup(void)
{
    arVideoCapStop();
    arVideoClose();
    argCleanup();
}
```

Cleanup function called when program exit or when key event is pressed, or when there is no marker found by detect function, the main task cleanup function do first stops the video

capture routine by calling `arVideoCapStop()` which return 0 if successful, -1 if the capture couldn't be stopped., then `arVideoClose()` called to close the video source. After your application has finished using a video stream, this function must be called to close the link to the input source, and free resources associated with the capture operation so the function will return 0 if shut down successfully, otherwise -1. At the last of the program `argCleanup()` is called to close gsub library and clean the rendering context (GLUT and OpenGL) thus Called in the exit of the program.



CHAPTER Seven

Results, discussion and conclusion

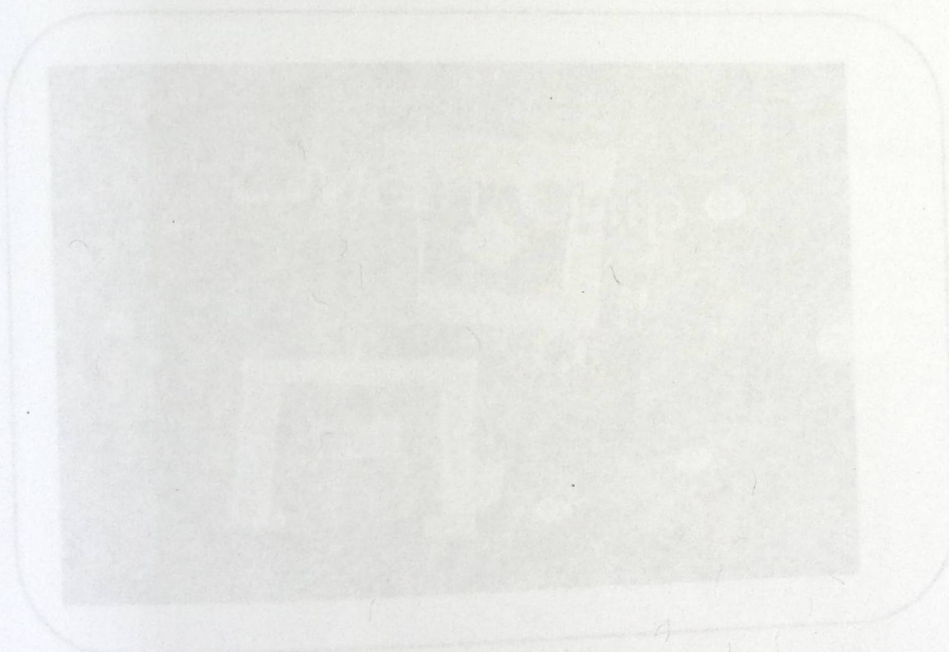
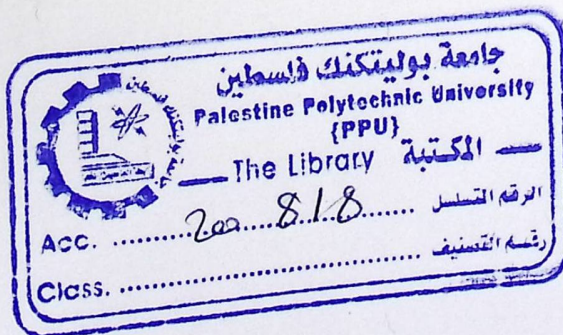


Figure 7.1 data flow from Main Chip to the RAM



results:

We found that the results will demonstrate the potential benefits of using AR visualization interfaces in education and training.

AR make students immerse in the system, because they could touch and see things and that help them to understand how the data really transfer in the computer, also it helps teachers to explain their ideas clearly.

Figure 7.1 show one of our experiment pictures about the flow of data from MainChip to the RAM.

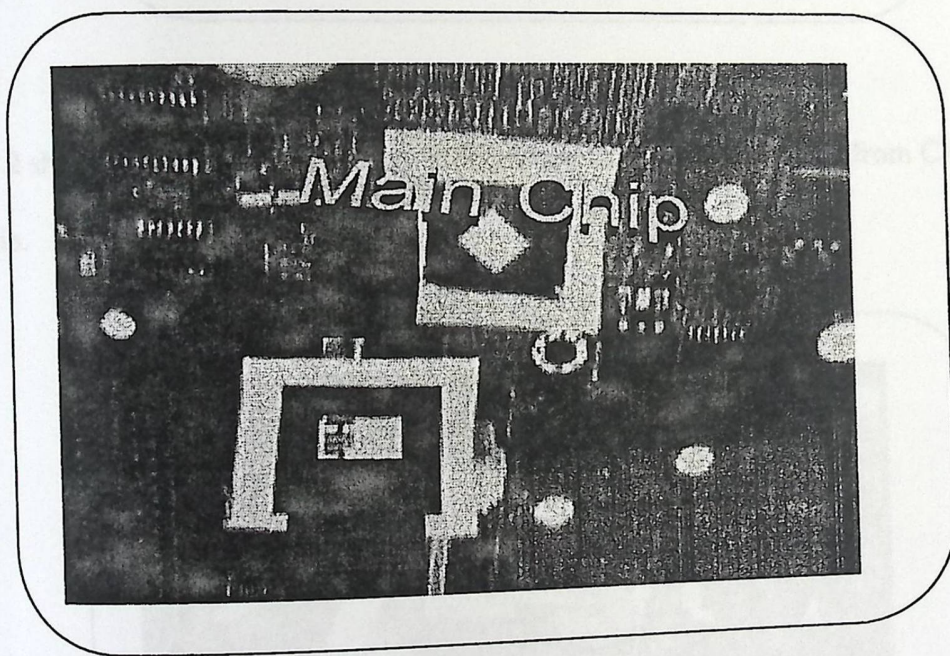


Figure 7.1 data flow from MainChip to the RAM

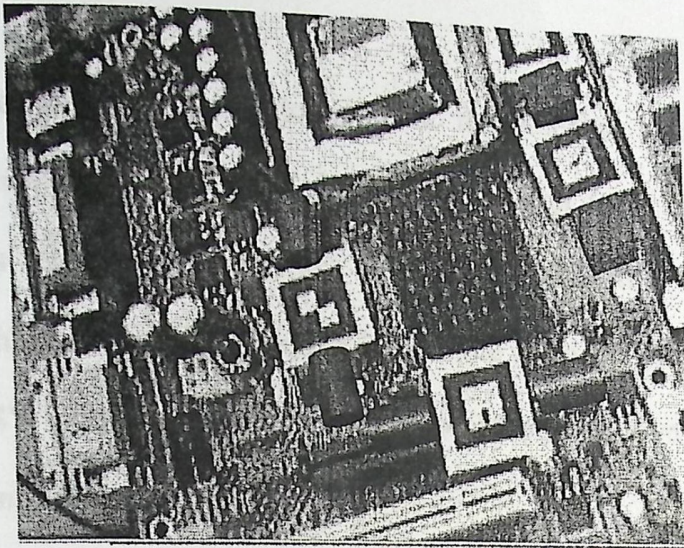


Figure 7.2: CPU send request to the MainChip

Figure 7.2 shows the virtual cylinders which presents the data that flows from CPU to the MainChip.

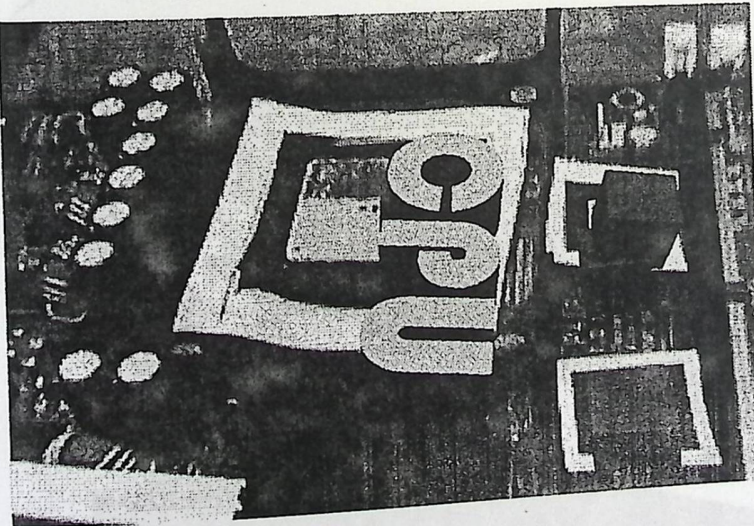


Figure 7.3: Interrupt for CPU

Figure 7.3 shows virtual cubes presented as interrupts comes from the MainChip to the CPU.

Interactive Augmented Reality applications, make the students more interested in the application, but when making this kind of applications with RGB model cause a lot of problems because its sensitivity to different lighting conditions that cause problems with detecting red pen, which make the interaction impossible in poor lighting conditions, on the other hand using HSV model resolve this kind of problems and give a good results when detecting the read pen.

Figure 7.4 shows how RGB Model work while trying to detect a red pen, we get a good results when we used it under very good lighting conditions.

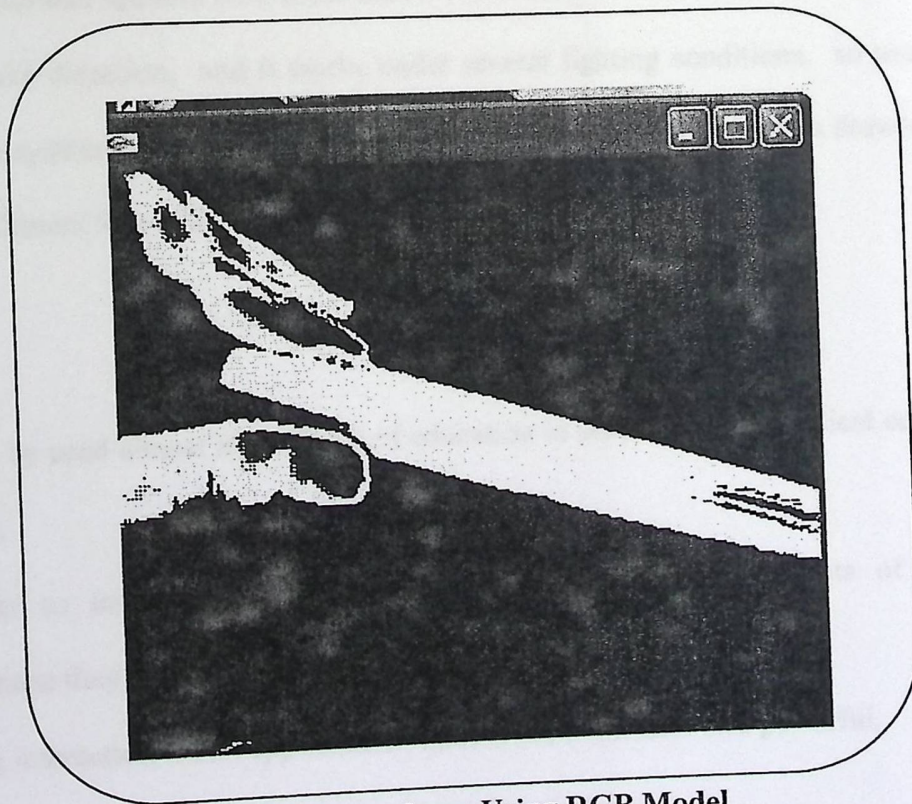


Figure 7.4: Detect red pen Using RGB Model

Discussion:

In the traditional way of learning architecture course, teacher brings some computer components in order to make the students understand what he say, but there still a problem in understanding all things that he say.

With our experiment the student can see, touch and interact with the component and some times they think that is real, so they will understand what the teacher say.

To make that researchers use RGB model at the beginning of there experiment it was easy to implement and fast in processing but it has some problems related with the light conditions, because RGB model need a good lighting conditions, so in a poor lighting it hard and almost impossible to make detection so user in this condition will not be able to interact with the system.

To resolve the problem appears with RGB model researchers use HSV model, it give better results when make detection, and it works under several lighting conditions, so users can interact with the system all the time, on the other hand this method has its own drawbacks it is harder to implement than RGB, and it takes time in processing.

Conclusion:

- AR can be used almost in all fields of education in order to make practical education process.
- In order to increase understanding and perception of the students of special phenomena they must see, touch and expertise it.
- Adding interaction to AR applications make it more effective and powerful.

- It is efficient to use AR applications in different areas cause it reduce time of learning and decrease effort that paid by instructor to clarify lesson.

Recommendation:

After finishing work in this project we have some recommendations to:

University:

To use Augmented Reality as a new concept for practical education.

Next generation of graduated students:

To work in this subject in order to expand it to include other educational applications, such as learning how to maintain specific objects.

Developer of AR applications:

To use ARToolKit in order to make such applications.

- Copy the files DSVM.dll and DSVM.dll from [ARToolKit]\DSVM into [ARToolKit]\bin.
- Update glui library, copy glui_32.dll from glui library folder into windows\system32, and include glui.h, LIBglui32.lib into visual studio SDK folder.
- In order to operate simplified examples you need to extract OpenVRML-0.14.3 in ARToolKit directory.

Problems while operating some examples:

We found some problems while trying to operate some examples especially in simplified example.

At the beginning we used VisualStudio2005 as a development environment, with ARToolKit 2.2.1 and that caused us a lot of problems. While compiling simplified example we needed some dll files such as libAR.dll, libARgui.dll, libARmath.dll, libARvideo.dll, libARvrml.dll, libARvrmlutil.dll, libARvrmlutil.dll, libARvrmlutil.dll, after installing these files from different web sites, then the problem was resolved. In simplified there were different kinds of errors, the first problem appears was in libvrml.dll, and this library was not available on

Appendices

Appendix 1:

Build steps for ARToolKit:

- Unzip the ARToolKit folder, this folder name should be ARToolKit, and it does not matter where you extract the folder.
- Unzip DSVideoLib directory in ARToolKit folder, and be sure that the directory name is DSVL.
- Copy the files DSVL.dll and DSVLd.dll from {ARToolKit}\DSVL\bin into {ARToolKit}\bin.
- Unzip glut library, copy glut 32.dll from glut library folder into windows\system32, and Include\gl\glut.h, Lib\glut32.lib into visual studio\SDK folder
- In order to operate simplevrml example you need to extract OpenVRML-0.14.3 in ARToolKit directory.

Problems while operating some examples:

We faced some problems while trying to operate some examples especially in simplevrml example.

At the beginning we used visualstudio2005 as a development environment, with ARToolKit-2.72.1 and that caused us a lot of problems. While compiling simpletest example we needed some dll files such as libARd.dll, libARgsubd.dll, libARmultid.dll, libARvideod.dll, libARgsub_lited.dll, libARgsubUtild.dll, libARvideod.dll, after installing these files from different web sites, then the problem was resolved. In simplevrml there were different kinds of errors. The first problem appears was in libarvrml.lib, and this library was not available on

the internet so in order to resolve this problem we used openvrml-0.14.3 and replace the files in lib/SRC/ARvrml with the file from ARToolKit-2.71 version. And build it to see the library (libarvrml.lib) in ARToolKit\lib, and that happened.

Another error was appeared and other libraries was required to make the example work successfully such as libbng.lib, libjbg.lib and Zlib.lib libraries even though libraries was downloaded from Zlib.org, but the example still not working.

After that we littered ARToolKit team in the forum of HitLab [21], and they answered us that the problem may be in OpenVrml library, and may be it does not work under VS2005, and they told us to use VS2003, also they told us that we can run Simplevrml.exe under VS2005 by doing these steps:

- Downloaded msvc71d.dll from <http://www.dll-files.com/dllindex/dll-files.shtml?msvc71d> and placed into {ARToolKit}\bin
- Downloaded msucr71d.dll from <http://www.dll-files.com/dllindex/dll-files.shtml?msucr71d> and placed into {ARToolKit}\bin
this was done to get {ARToolKit}\bin\videotestd.exe to work. Then rebuilt with Release Configuration and successfully executed {ARToolKit}\bin\graphicsTest.exe, and simpleTest.exe
- **Make the following changes to the OpenVRML file:**

```
Code changes to {ARToolKit}\OpenVRML\src\openvrml0.14.3\lib\regex\regerror.c
// regerror(errcode, preg, errbuf, errbuf_size)
regerror(int errcode, const regex_t *preg, char *errbuf, size_t errbuf_size)
// int errcode;
// const regex_t *preg;
// char *errbuf;
// size_t errbuf_size;
```

```
Code changes to {ARToolKit}\OpenVRML\src\openvrml
0.14.3\lib\antlr\antlr\AST.hpp
// #ifdef NEEDS_OPERATOR_LESS_THAN
```

```
// inline operator< (RefAST l, RefAST r); // {return true ;}
// #endif
#ifdef NEEDS_OPERATOR_LESS_THAN
int inline operator< (RefAST l, RefAST r); // {return true;}
#endif
```

Code changes

```
to {ARToolKit}\OpenVRML\src\openvrml0.14.3\lib\antlr\antlr\Token.hpp
#ifdef NEEDS_OPERATOR_LESS_THAN
//inline operator< (RefToken l, RefToken r); //{return true ;}
//endif
#ifdef NEEDS_OPERATOR_LESS_THAN
int inline operator< (RefToken l, RefToken r); //{return true;}
#endif
```

- From the Visual Studio 2005 OpenVRML Solution at {ARToolKit}\OpenVRML\src\openvrml-0.14.3\ide-projects\Windows\VisualC7\OpenVRML build the OpenVRML Solution under the Debug Configuration (see Configuration Manager). Then rebuild with antlr as Release, lookat as Debug, openvrml as Release, openvrml-gl as Release, regex as Release Configuration.
- Copy {ARToolKit}\OpenVRML\src\openvrml-0.14.3\ide-projects\Windows\VisualC7\OpenVRML\antlr\Release\antlr.lib to {ARToolKit}\OpenVRML\lib
- Copy {ARToolKit}\OpenVRML\src\openvrml-0.14.3\ide-projects\Windows\VisualC7\OpenVRML\openvrml\Release\openvrml.lib to {ARToolKit}\OpenVRML\lib
- Copy {ARToolKit}\OpenVRML\src\openvrml-0.14.3\ide-projects\Windows\VisualC7\OpenVRML\openvrml-gl\Release\openvrml-gl.lib to {ARToolKit}\OpenVRML\lib

- Copy {ARToolKit}\OpenVRML\src\openvrml-0.14.3\ide-projects\Windows\VisualC7\OpenVRML\regex\Release\regex.lib to {ARToolKit}\OpenVRML\lib
- Also, copy {ARToolKit}\OpenVRML\src\openvrml-0.14.3\ide-projects\Windows\VisualC7\OpenVRML\openvrml\Release\openvrml.lib to {ARToolKit}\lib
- Download the following file: libpng-1.2.8.exe 859791 13453 i386.exe (32-bit Windows) from: <http://gnuwin32.sourceforge.net/packages/jpeg.htm>
Delete {ARToolKit}\OpenVRML\dependencies\lib\libpng.lib
Install libpng-1.2.8.exe and copy C:\Program Files\GnuWin32\lib\libpng.lib to {ARToolKit}\OpenVRML\dependencies\lib download it
- Download jpeg-6b-4.exe
Delete {ARToolKit}\OpenVRML\dependencies\lib\libjpeg.lib
Install jpeg-6b-4.exe and copy C:\Program Files\GnuWin32\lib\jpeg.lib to {ARToolKit}\OpenVRML\dependencies\lib, and rename it from jpeg.lib to libjpeg.lib also, copy jpeg62.dll, and libpng13.dll, and zlib1.dll from C:\Program Files\GnuWin32\bin to {ARToolKit}\bin {ARToolKit}\ARToolKit.sln successfully built under Visual Studio 2005 with Release Configuration; actually had to rebuild twice for dependencies of two files.

These steps were made and saw the final out put of this example, there were two virtual objects appeared on two markers bud and snow man.

Finally we resolved all the problems that we faced it by downloading another version of ARToolK which is the latest version of ARToolKit(ARToolKit_2.72.1_bin_win32, use it with OpenVRML-0.14.3-win32 after, under VS2003) it is include every library you need it and every dll files there is nothing you will need to be downloaded.

Appendix 2

Paper for our project:

Abstract:

In this project we will demonstrate the ability of Augmented Reality by applying it in education, to bring the educational process to practical issue, and to reduce learning time of the student by adding virtual object to real world. We develop augmented computer architecture experiment to apply the idea of augmented reality; we illustrate in our experiment the flow of data in computer and show the component of computer. This prototype enhances understanding process of computer architecture course student by seeing the data movement in computer.

Interactive education one of recent trend that followed by many universities in the world cause of the benefit that can be taken through interacting with specific content and support the education process, make it powerful, more interesting and more understandable, to achieve that we apply this idea by developing interactive computer architecture using Augmented Reality .This done by using image processing abilities to detect where the user point on the marker, then make the action.

Researchers suppose that the person who will use this system has a good background about using computers programs it is not necessary that he will be an expert, and there educational application will make faster and easier understanding in architecture course, so it will be a good helper for teacher, and that AR changes the way students come to understand certain concepts.

After working in this project we fined that student's understandings of spatial content change through their physical interactions with virtual objects. Also interactive learning increase imagination in students mind, because they see, touch and expertise things. The results will demonstrate the potential benefits of using AR visualization interfaces in education and training.

Key words: augmented reality, virtual object, ARToolkit, marker, VRML, pattern

Introduction:

There are some difficulties which the student face in understanding certain topics, and they take time to learn it and discover how it work, so in order to reduce learning time of the student, increase their creativity in different educational level and on the other hand make education process more interesting. To achieve this we used Augmented Reality concept which has been applied on wide variety of science like chemistry, medicine, computer science etc...

In the traditional way of learning the teacher was using chalk to write the lesson, and use pictures as the only way to help him in order to explain his idea, now computers enters our lives and make work easier, even though teachers struggle to perform relatively simple tasks, such as connecting their computer output to the projector, switching the display to show a video tape, and even turning on the lights, and that takes time from the teacher.

AR is a recent trend appears and makes the educational process easier, AR defined as the ability to overlay computer graphics onto the real world is commonly called Augmented Reality¹¹. At present, most AR research is concerned with the use of live video imagery which is digitally processed and "augmented" by the addition of computer generated graphics.

Related work:

AR technology can be used in wide range application domains, especially in educational area; it can add good additional value to educational process, even though for single or collaborating use.

Single user AR interfaces have been developed for computer aided instructions [Feiner 93], using this approach in area like manufacturing [Caudell 92], and medical visualization [Bajura 92], enable users to interact with the real world, and make work done easier. In manufacture area AR can used in order to teach the user how to repair things [Feiner 93],

¹¹ <http://www.newhorizons.org/strategies/technology/billinghurst.htm>

such as a laser printer, user can see virtual objects and notation in order to make practical explanation for repairing the printer, also in medical area doctors can see virtual objects on the patient's body allowing them to have "X-Ray" vision on a needle biopsy task [Bajura 92].

Also augmented reality have been used in collaborative work, a lot of projects is made to enhance collaborative work such as StudierStube project of Schmalsteig [Schmalsteig 96] in this project they use see-through head mounted displays to enable the users to collaboratively view 3D model of scientific data superimposed the real world. Another work for collaborative work AR2Hokey [Ohshima 98] in this application players wear see-through head mounted displays and play with virtual buck by using real table and mallet, they send the virtual buck to each other goal.

In computer-based education researchers have found that even though every student has his own computer they not perform good as when they all around one single machines [Inkpen 97], also when students work on individual machines they will spontaneously cluster around machines in pairs and trios [Watson 91, Strommen 93], they also have found that physical objects is commonly used to convey meaning and it has the ability to focus attention [Gav 97], also enhance the physical objects, For example, in the Shared Space interface users could manipulate three-dimensional virtual objects simply by moving real cards that the virtual models appeared attached to [Poupyrev 2000]. There was no mouse or keyboard in sight. This property enables even very young children to have a rich educational experience.

When using AR interface with students who set on around table, that will be more comfortable than setting all together in front of the same machine; and they can see each other and make good discussion about the virtual object that appears in the middle of the table [Kiyokawa 2002].

MagicBook [Billingham 2001] project is an educational project for children, in this project the user can read in a normal book without using any technology, and when using a handheld AR display he will see 3D virtual objects appear out from the real page, and the user can see the object from different view, by moving the book.

Proposed work:

Learning computer architecture using augmented reality:

Computer architecture is field of study that show the architecture of computer, its component, the processes made inside, and clarify how data move from one place to another through buses, the difficulty of this subject that although the student see component physically but they can't realize the processes that have been made inside on the other hand they can't see the flow of data between these component.

Our project examines the advantages of the AR interface for viewing and manipulating 3D objects. We want to know how students' understandings of spatial content change through their physical interactions with virtual objects. Our hypothesis is that AR changes the way students come to understand certain concepts. The results will demonstrate the potential benefits of using AR visualization interfaces in education and training.

We used VRML to build our virtual object which can be used to clarify component in the computer architecture in order to overlay these virtual objects onto the real objects in real time. We used pattern that the program can recognize it and put on it the virtual object. We pasted pattern on the component we wanted to clarify it, and videotaping the computer from inside via webcam in order to display real component stuck with virtual object that clarify it.

Making interactive AR application provide the user with all potential benefits from these kinds of systems, in our project we add interaction with the virtual objects that appears on the component of the computer when the user touch the marker on the object the parts of these objects, and brief description about the object that the user touch it will appear on the screen, that make the user immerge in the experiment and enlarge the imagination in his mind.

Design interactive AR application:

Until recently AR presented a passive interface to its users, who merely just a viewer of the visual representation by AR, now a days AR provide us with an interactive environment. In our project we used our abilities in image processing to build our interactive application; firstly we capture real time images by using USB camera, then convert images from RGB model into HSV model and convert it from RGB model to HSV model, then we take colors in specific range (threshold), in order to take the red color only and make detection based on the center of the red pen, then make the action that supposed to be done.

Experiment and results:

We use in our experiment a webcam camera with good resolution, and ARToolKit which is a C and C++ language software library that allow programmer to develop augmented reality applications, Figure 1.1 show the general ARToolKit work.

Camera start to capture images of the real world object and send it to the computer, The program searches every video frame for any square when the position of the camera is known the virtual object is drawn from the same position. If the square is found the software uses some mathematics to calculate the position of the camera relative to the black square, and Draw virtual object in video frame.

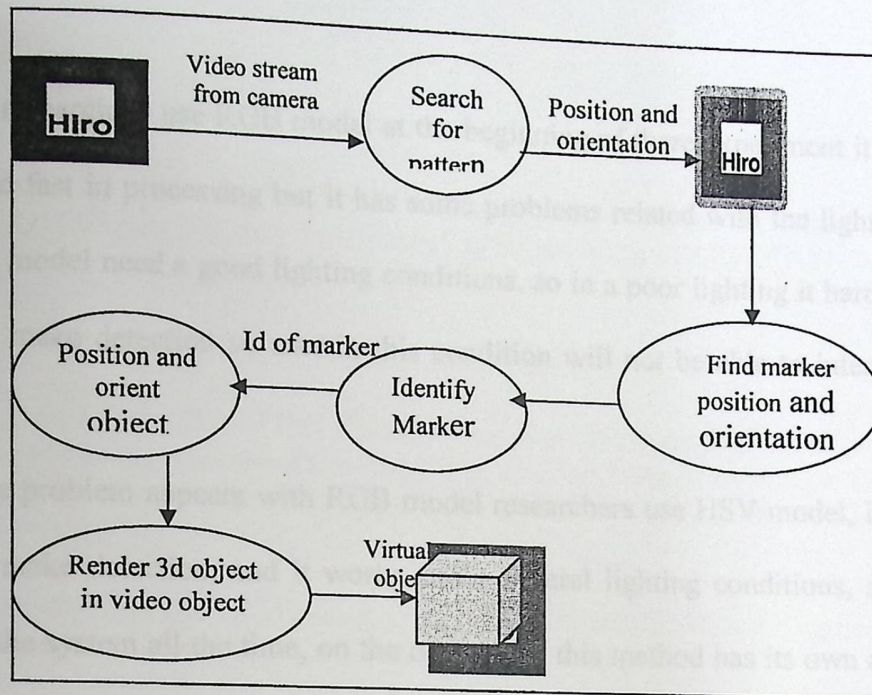


Figure 1.1 ARToolKit

When students saw the system they think that that the objects was real and they try to touch it to see what will happened, so researchers find that it is necessary to add interaction in order to meet the expectations of the users.

Interaction increase user emerge in the system, so researchers try to add interaction by using RGB model, but they faced some problems when using it, and discover that it will not give accurate results, so they work with HSV model because it give a good results when detecting the read pen.

Also our project needs good lightning conditions.

Discussion and conclusion:

In the traditional way of learning architecture course, teacher brings some computer components in order to make the students understand what he say, but there still a problem in understanding all things that he say.

With our experiment the student can see, touch and interact with the component and some times they think that is real, so they will understand what the teacher say.

To make that researchers use RGB model at the beginning of there experiment it was easy to implement and fast in processing but it has some problems related with the light conditions, because RGB model need a good lighting conditions, so in a poor lighting it hard and almost impossible to make detection so user in this condition will not be able to interact with the system.

To resolve the problem appears with RGB model researchers use HSV model, it give better results when make detection, and it works under several lighting conditions, so users can interact with the system all the time, on the other hand this method has its own drawbacks it is harder to implement than RGB, and it takes time in processing.

Johnson, K. (1997) Adapting the Human-Computer Interface to Support Collaborative Learning Environments for Children. PhD Dissertation, Dept. of Computer Science, University of British Columbia.

K. Nishikawa, M. Billingham, S. Hayes, A. Gupta, Y. Sanohe, H. Kato (2002) "Communication Behaviors of Co-located Users in Collaborative AR Interfaces." In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002), 30 Sept. - 1 Oct., 2002, Darmstadt, Germany, IEEE Press, Los Alamitos, CA, pp. 179-182.

Gleason, T., Satoh, K., Yamamoto, H. and Tamura, H. (1998) "AR2 Hockey: A Case Study of Collaborative Augmented Reality," Proc. IEEE VRAIS '98, pp.265-275.

Preyner, L., Billingham, M. Kato, H., May, K.(2000) "Integrating Real and Virtual Worlds in Shared Space." In Proceedings of the 5th International Symposium on Artificial Life and Robotics (AROB 5th/00), Oita, Japan, 26-28 January 2000, Vol. 1, pp. 22-25.

Stankovic, D., Fuhrmann, A., Szalay, Z., Gervautz, M. (1996) "Studierstube - An Environment for Collaboration in Augmented Reality." In CVE '96 Workshop Proceedings, 19-20 September 1996, Nottingham, Great Britain.

Bransford, J.F. (1993) "Does your oak leaves? Cooperative learning in an educational activity task." Journal of Computing in Childhood Education, 4(1), 45-56.

Winters, J. (1991) "Cooperative learning and computers: One way to address student differences." The Computing Teacher, 14(4), pp. 9-15.

References for the paper:

Bajura, M., Fuchs, H., Ohbuchi, R. (1992) "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery Within the Patient." In Proceedings of SIGGRAPH '92, New York: ACM Press, pp. 203-210.

MagicBook website <http://www.hitl.washington.edu/magicbook/>

Caudell, T.P., and Mizell, D.W. (1992) "Augmented Reality: an application of heads-up display technology to manual manufacturing processes." In Proceedings of the Twenty-Fifth Hawaii International Conference on Systems Science, Kauai, Hawaii, 7th-10th Jan. 1992, Vol. 2, pp. 659-669.

Feiner, S., MacIntyre, B., and Seligmann, D. (1993) "Knowledge-Based Augmented Reality." Communications of the ACM, Vol. 36(7), pp. 53-62.

Gav, G., Lentini, M. "Use of Communication Resources in a Networked Collaborative Design Environment." http://www.osu.edu/units/jcmc/IMG_JCMC/ResourceUse.html.

Inkpen, K. (1997) Adapting the Human Computer Interface to Support Collaborative Learning Environments for Children. PhD Dissertation, Dept. of Computer Science, University of British Columbia.

K. Kiyokawa, M. Billingham, S. Hayes, A. Gupta, Y. Sannohe, H. Kato. (2002) "Communication Behaviors of Co-Located Users in Collaborative AR Interfaces." In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002), 30 Sept. - 1 Oct., 2002, Darmstadt, Germany, IEEE Press, Los Alamitos, CA, pp. 139-148.

Ohshima, T., Satoh, K., Yamamoto, H. and Tamura, H. (1998) "AR2 Hockey: A Case Study of Collaborative Augmented Reality," Proc. IEEE VRAIS '98, pp.268-275.

Poupyrev, I., Billingham, M. Kato, H., May, R. (2000) "Integrating Real and Virtual Worlds in Shared Space." In Proceedings of the 5th International Symposium on Artificial Life and Robotics (AROB 5th'00), Oita, Japan, 26-28 January 2000, Vol. 1, pp. 22-25.

Schmalstieg, D., Fuhrmann, A., Szalavari, Z., Gervautz, M., (1996) "Studierstube - An Environment for Collaboration in Augmented Reality." In CVE '96 Workshop Proceedings, 19-20th September 1996, Nottingham, Great Britain.

Strommen, E.F. (1993) "Does yours eat leaves?" Cooperative learning in an educational software task." Journal of Computing in Childhood Education, 4(1), 45-56.

Watson, J. (1991) "Cooperative learning and computers: One way to address student differences." The Computing Teacher, 18(4), pp. 9-15.

References:

- [1] Ronald T. Azuma (August 1995), A Survey of Augmented Reality.
- [2] Mark A. Livingston, Ph.D., Dennis G. Brown, Simon J. Julier, Ph.D., and Greg S.Schmidt, PhD (2006), Mobile Augmented Reality: Applications and Human Factors Evaluations, Washington, USA
- [3] Invisible train project http://studierstube.icg.tu-graz.ac.at/invisible_train
- [4] Feiner, S., MacIntyre, B., and Seligmann, D. (1993) "Knowledge-Based Augmented Reality." Communications of the ACM, Vol. 36(7), pp. 53-62.
- [5] Gav, G., Lentini, M. "Use of Communication Resources in a Networked Collaborative Design Environment." http://www.osu.edu/units/jcmc/IMG_JCMC/ResourceUse.html.
- [6] Bajura, M., Fuchs, H., Ohbuchi, R. (1992) "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery Within the Patient." In Proceedings of SIGGRAPH '92, New York: ACM Press, pp. 203-210.
- [7] Schmalsteig, D., Fuhrmann, A., Szalavari, Z., Gervautz, M., (1996) "Studierstube - An Environment for Collaboration in Augmented Reality." In CVE '96 Workshop Proceedings, 19-20th September 1996, Nottingham, Great Britain.
- [8] Ohshima, T., Satoh, K., Yamamoto, H. and Tamura, H.(1998) "AR2 Hockey: A Case Study of Collaborative Augmented Reality," Proc. IEEE VRAIS '98, pp.268-275 .
- [9] Inkpen, K. (1997) Adapting the Human Computer Interface to Support Collaborative Learning Environments for Children. PhD Dissertation, Dept. of Computer Science, University of British Columbia.
- [10] Watson, J. (1991)" Cooperative learning and computers: One way to address student differences." The Computing Teacher, 18(4), pp. 9-15.
- [11] Strommen, E.F. (1993) "Does yours eat leaves?" Cooperative learning in an educational software task." Journal of Computing in Childhood Education, 4(1), 45-56.
- [12] Poupyrev, I., Billighurst, M. Kato, H., May, R.(2000) "Integrating Real and Virtual Worlds in Shared Space." In Proceedings of the 5th International Symposium on Artificial Life and Robotics (AROB 5th'00), Oita, Japan, 26-28 January 2000, Vol. 1, pp. 22-25.
- [13] K. Kiyokawa, M. Billinghamurst, S. Hayes, A. Gupta, Y. Sannohe, H. Kato. (2002) "Communication Behaviors of Co-Located Users in Collaborative AR Interfaces." In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002), 30 Sept. - 1 Oct., 2002, Darmstadt, Germany, IEEE Press, Los Alamitos, CA, pp. 139-148.

- [14] **MagicBook** website <http://www.hitl.washington.edu/magicbook/>
- [15] Determining the cost of the operating system Windows XP as is bought in market at <http://labmice.techtarget.com/FAQ/winxpfaq.htm>
- [16] Determine the cost of Microsoft Visual Studio .NET Professional 2003 English as is boaght in market at the amazon site <http://www.amazon.com/Microsoft-Visual-Studio-Professional-English/dp/B00008I9K2>
- [17] ARToolkit documentation used in implementation chapter, for developing first application in augmented reality follow this link <http://www.hitl.washington.edu/artoolkit/documentation/devstartup.htm>
- [18] Information about the function in OpenGL at <http://developer.3dlabs.com/documents/GLmanpages/glenable.htm>
- [19] Information about OpenGL that used in implementation chapter at this site <http://www.eecs.tulane.edu/Terry/OpenGL/Introduction.html#Introduction>
- [20] Information about OpenGL function at <http://pyopengl.sourceforge.net/documentation/manual/glutSolidCube.3GLUT.html>
- [21] Hit lab forum that used to solve problems at attempting running simplevrml at <http://www.hitlabnz.org/forum/showthread.php?t=550>
- [22] Brett E.Shelton, December (2002), Augmented reality in education (Current project and the potential class room), Montreal, Canada.
- [23] Jeremy R.Cooperstock (2001), The Classroom of the Future (Enhancing Education Through Augmented Reality), New Orleans, USA.
- [24] Nicholas R.Hedley, Mark Billinghust, Lori Postner, Richard May, Hirokazu Kato (April 2002), Exploration in the use of Augmented Reality for Geographic Visualization, Washington, USA.
- Caudell, T.P., and Mizell, D.W. (1992)"Augmented Reality: an application of heads-up display technology to manual manufacturing processes." In Proceedings of the Twenty-Fifth Hawaii International Conference on Systems Science, Kauai, Hawaii, 7th-10th Jan. 1992, Vol. 2, pp. 659-669.

Bibliography:

Chapter 1:

Introduction:

1.1 Abstract.....1
1.2 Using Augmented Reality in Education.....2
1.3 Project outline.....3

Chapter 2

Literature review:

2.1 Introduction.....4
2.2 Literature review for previous work.....4

Chapter 3

Requirements:

3.1 Hardware requirement.....7
3.2 Software requirements8
 3.2.1 Developer requirements.....8
 3.2.2 User requirements.....11

Chapter 4

Design:

4.1 ARToolKit background12
4.2 History of ARToolKit12
4.3 How ARToolKit work?.....13

Chapter 5

Using Augmented Reality in Education:

5.1 Introduction.....23
5.2 Augmented Reality in Computer Architecture Course.....25
5.3 Description of our experiment.....26
5.5 Interactive Augmented Reality.....30
5.6 Designing interactive augmented reality program.....34

Chapter 6

Implementation:

6.1 Main.....37
6.2 Initialization Function.....38
6.3 arVideoCapStart function.....40
6.4 argMainloop function.....41
6.5 Mainloop function.....45
6.6 Draw Function.....47
6.7 Cleanup function.....47

Chapter 7:

Result, Discussion, Conclusion AND Recommendation:

7.1 Result.....	50
7.2 Discussion	53
7.3 Conclusion.....	53
7.4 Recommendation	55

Appendices:

Appendix 1:

1.1 Build steps for ARToolKit.....	56
1.2 Problems while operating some examples.....	56

Appendix 2:

2.1 Paper for our project.....	60
2.1.1 Abstract	60
2.1.2 Introduction	61
2.1.2 Related work	61
2.1.3 Proposed work	61
2.1.4 Learning computer architecture using augmented reality.....	63
2.1.5 Design interactive AR application.....	64
2.1.6 Experiment and results.....	64
2.1.7 Discussion and conclusion.....	65
2.1.8 References for the paper.....	67

References:

References.....	68
-----------------	----