Department of Computer Science
The University of Hong Kong

CSIS0801 – Final Year Project

Final Report - GUI
Intelligent Mirror for Augmented Fitting Room Using Kinect
By Ma Cheuk Kiu, Charles 2008303898

Supervisor:  Dr. Kenneth Wong
2nd Examiner: Dr. Anthony Tam
Members: Ho Chung Yin, Billy 2008196829
Ma Cheuk Kiu, Charles 2008303898
Ngai Ka Kit, Kit 2008539150

Dated as: 11th April, 2012
# Table of Contents

Table of Contents

Project Overview 3
Tasks 3
Research Findings 4
Approach for Cloth Fitting 4
Introduction to Kinect 5
  1. General components 5
  2. Hardware and Streams provided 5
  3. Nui Skeleton API 6
Phase 1 7
Phase 2 9
  1. Functionalities 12
Result in phase 2 13
Mid-term Conclusion 14
  1. Mid-term evaluation phase 14
Phase 3 16
  1. Graphical design 16
  2. User-friendliness - User Interface structure 17
  3. Easy Control - User Interface design 22
  4. Functionality 24
Conclusion 26
References 27
Project Overview

With the rapid growth of technology development, our daily lives are heavily affected. Nowadays, people are getting more used to online shopping, online auctions etc to purchase their interested products. This way of transaction has become the main trend and it does bring great convenience to customers. However, an issue for buying clothes online is that you can’t try on them before you get the product. The feeling after dressing on does affect your decision on buying the clothes. Therefore, a kind of virtual “fitting room” can be developed to simulate the visualization of dressing.

The objective of this project is to develop an intelligent mirror that can augment the image of a customer standing in front of it with different clothes fitted to his/her body. In particular, the customer can pose freely in front of the mirror, e.g., turning around to look at his/her back and side-view, and the fitted clothes will keep aligned with his/her body poses in real-time. Such an intelligent mirror can be deployed in the fitting room or advertisement of latest fashion of a garment shop, or even at the home of a customer for shopping over the Internet.

Tasks

1. To develop 3D models for clothes and human body with precise mechanical control
2. To develop a user-friendly application for users to try out different types of clothes for visualization using Kinect
3. To develop possible templates for users to design their own clothes

In this report, we would mainly focus discussion on the GUI design part.
Research Findings

Before we decide the approach, we have done a lot of research online to look for relevant materials. There were already people/teams working on similar projects to develop a virtual fitting room using Kinect. However, most of them are taking the approach to map a 2D texture to the user’s body. Hence when the user moves around you can easily see that the clothes not accurately capturing the user’s position and movement. In order to achieve a more realistic simulation on the clothes fitting process, we have decided to take an approach that requires the construction of two 3D models.

Approach for cloth-fitting

Firstly, we will create a 3D human model according to the user’s dimensions (the body shape, height, width, length of limbs etc.) from the data captured by Kinect. The whole human model will always follow the motion of the skeleton model captured by the Kinect. That is, how the user moves will be reflected by the skeleton model, and our human model will do the same movements.

Secondly, a cloth model will be created according to some mechanics such as frictional forces, gravity, elasticity etc. The cloth model is used to build various types of clothes for dressing on.
In real-time, the interaction takes place between the human model and cloth model. The clothes will be fitted on the human model, and will move in the same way as the human model moves. Hence, a realistic simulation on fitting is done by the interaction between our two models.

Introduction to Kinect

1. General component

The components of Kinect for Windows are mainly the following:
1. Kinect hardware: Including the Kinect sensor and the USB hub, through which the sensor is connected to the computer;
2. Microsoft Kinect drivers: Windows 7 drivers for the Kinect sensor;
3. NUI API: core of the Kinect for the set of Windows API, supports fundamental image and device management features like access to the Kinect sensors that are connected to the computer, access to image and depth data streams from the Kinect image sensors and delivery of a processed version of image and depth data to support skeletal tracking.

2. Hardware and streams provided

![Fig 1.1 Hardware and software interaction with an application](image)

Kinect sensor mainly provides 3 streams: Image stream, depth stream and audio stream, with detected range from 1.2 to 3.5 meters. At this stage, the first two streams would be utilized for development of human model, cloth simulation and GUI.

The middle camera is a 640×480 pixels @ 30 Hz RGB camera, providing image stream which is delivered as a succession of still-image frames for the application. The quality level of color image determines how quickly data is transferred from
the Kinect sensor array to the PC which is easy for us to optimize the program on different platform. The available color formats determine whether the image data that is returned to our application code is encoded as RGB.

The leftmost one is the IR light source with a corresponding 640×480 pixels @ 30 Hz IR depth-finding camera with standard CMOS sensor on the right, which mainly provide the depth data stream. This stream provides frames in which the high 13 bits of each pixel give the distance, in millimeters, to the nearest object at that particular x and y coordinate in the depth sensor's field of view.

3. Nui skeleton API

Among NUI API, NUI Skeleton API provides information about the location of users standing in front of the Kinect sensor array, with detailed position and orientation information. Those data are provided to application code as a set of 20 point, namely skeleton position. This skeleton represents a user’s current position and pose. Our applications can therefore utilize the skeleton data for measurement of different dimension of users’ part and control for GUI. Skeleton data are retrieved as aforementioned image retrieval method: calling a frame retrieval method and passing a buffer while our application can then use an event model by hooking an event to an event handler in order to capture the frame when a new frame of skeleton data is ready.
With the Microsoft Kinect device as a motion sensing input device, we can develop a GUI controlled by user motions in order to provide an interactive environment for better user experience.

**Phase 1:**

Human motion tracking would be our first target to be finished. By comparing different software development kit for Kinect released on network, our group have decided the official Kinect SDK - The Kinect for Windows SDK Beta v1.0 (latest version at that time) as our development platform as this seems to have more concrete support by Microsoft other than normal programmers and Kinect hackers.

Focusing on detecting user motion, the essential part of Kinect to be utilized is Skeletal tracking, i.e. to track the skeleton image of user moving within the Kinect field of view. It allows us to create gesture-driven applications.

**Implementation:**

For a WPF project, XAML is main component for creating user interfaces with code-behind in, for example using C#, xaml.cs. In this phase, skeleton stream is used for capturing user action and video stream for providing real-time image of user. For each stream an event handler is implemented to handle event changes. Smoothing on skeleton engine is also added for retrieving a more steady skeleton.

For **Skeletons** in each skeleton frame, **SkeletonData** is stored inside which provides different information of the particular **Skeletons** at **SkeletonFrame**. This includes: **Joints, Position, Quality, TrackingID, TrackingState** and **UserIndex**.

There are 3 states pre-set:

- **Tracked** – indicates success of capturing the **Skeleton**;
- **Position only** – indicates some part of **Skeleton** is chopped, out of detection field by Kinect or not 100% confirmed to be correct;
- **NotTracked** – indicates that **Skeleton** has not been tracked.

For correct skeleton tracking, the **TrackingState** of **Skeleton** should always be ensured as **Tracked**, and the user’s motions will be handled.
In this phase, skeleton is drawn out for testing purpose. It includes mainly two parts: bones and joints. By coloring with Brush using a set of colors for different parts of skeleton (i.e. body segment), they are drawn and added to the Skeleton. Parts like left arm, right arm, left leg, right leg and (head to hip) are individually drawn using Polyline. For each part of Skeleton, joints are drawn by points, which are pre-defined with different colors in order to identify the joints.

```
private void nui_SkeletonFrameReady(object sender, SkeletonFrameReadyEventArgs e)
{
    SkeletonFrame skeletonFrame = e.SkeletonFrame;
    Brush[] brushes = new Brush[6];
    brushes[0] = new SolidColorBrush(Color.FromRgb(255, 0, 0)); ...

    skeleton.Children.Clear();

    foreach (SkeletonData data in skeletonFrame.Skeletons)
    {
        if (SkeletonTrackingState.Tracked == data.TrackingState)
        {
            // Drawing bones
            Brush brush = brushes[iSkeleton % brushes.Length];
            skeleton.Children.Add(GetBodySegment(data.Joints, brush, JointID.HipCenter, JointID.Spine, JointID.ShoulderCenter, JointID.Head)); ...

            // Drawing joints
            foreach (Joint joint in data.Joints)
            {
                Point jointPos = GetDisplayPosition(joint);
                Line jointLine = new Line();
                jointLine.X1 = jointPos.X - 3;
                jointLine.X2 = jointLine.X1 + 6;
                jointLine.Y1 = jointLine.Y2 = jointPos.Y;
                jointLine.Stroke = jointColors[joint.ID];
                jointLine.StrokeThickness = 6;
                skeleton.Children.Add(jointLine);
                SetEllipse(joint.ID, jointPos, joint);
            }
        }
    }
}
```
Phase 2:

In this phase, there are 3 major objectives on user interface:
1. Graphics – design and style should be nice-looking to users;
2. Easy Control – the control method should be intuitive, e.g. users use their hands to touch the virtual buttons to access different functions;
3. User-friendliness – the location of buttons and menus should be well-presented for users’ convenience;

![Fig 1.3 GUI Design in phase 2](image)

For graphics, Photoshop is used in order to design delicate and pretty user interface with different effects. The whole design style mainly follows the rule “Simple is the best”.

For easy control, skeleton stream retrieved from Kinect can be utilized for motion capturing. The user’s hand will be shown on the screen, where left and right hand are represented by red and green hand icon respectively. Users use their hands to control buttons as the same way cursors do on GUI.
For user-friendliness, it is brought by intuitive graphical controls in which users can easily understand what the control refers to at their first sight. In traditional user interface, confirmation of selecting items and using functions requires clicking by mouse on a confirm button. This is not applicable in our GUI design as users would feel inconvenient to repeatedly move their hands onto same button in order to confirm. Therefore, we use the concept of hover button: users can confirm by ‘touching’ the item or button for few seconds with their hands for completing the action. This can also avoid the user touching a wrong control button when they move their hands across the interface. Users’ hand movement is also captured for implementation of a scrollbar. By moving the hand icon from top to bottom, user can browse the next page of the clothing catalog.

```csharp
private bool CheckTakePhoto()
{
    double leftX = Canvas.GetLeft(handRightEllipse) + 25;
    double leftY = Canvas.GetTop(handRightEllipse) + 25;

    if (leftX < 490 && leftX > 455 && leftY < 70 && leftY > 30)
    {
        if (!t3.Enabled)
        {
            t3.Enabled = true;
            t3.Elapsed += new System.Timers.ElapsedEventHandler(t_TakePhotoTimeOut);
            handRightEllipse.Hovering();
        }
        return true;
    } // remaining...
```
The picture below explains the GUI created in this phase:

![GUI explanation](image)

Fig 1.5 Tutorial page

The circular rotation disc in the top-left corner is the category button. Users can change the category shown among clothes, trousers, shoes and other accessories. The white round-cornered rectangle with scroll bar on the right is the catalog panel showing the items in selected category. Users can select their want-to-try item by the green hover button. As mentioned, the scroll bar is used to navigate through pages of the catalog.

The three buttons on the top-righthand corner are the functionalities we have completed in this phase to provide users with better user experience.

The leftmost button is the photo-taking button. Users can take photos with the chosen clothes dressed on their bodies. The screen will show a count from 5 to 0 for users to take their poses. Photo will then be saved in the folder created by the program and users can retrieve the photos there. The middle button is trash button which allows users to clear what they have worn. The rightmost button is tutorial button. Users who want to look at the instruction of this application can choose this function, and the tutorial page in Fig1.4 will be shown on the screen.
Functionalities

Photo-taking
This functionality enables user to save the screenshot image just like taking a photo. With the clothes dressed on the user’s body, the user can “press” the photo button on the panel. Then, a 5-seconds countdown will be displayed on the screen. Users can make use of the time to make their own pose. The output picture will be automatically saved to:

C:/Users/username/My Documents/KinectFitter/Photos

with the file name IMG_XXXX.jpg where XXXX is the id of the photo.

```
public static void SaveCanvasToFile(Canvas surface, string filename)
{
    Size size = new Size(surface.Width, surface.Height);
    surface.Measure(size);
    surface.Arrange(new Rect(size));

    // Create a render bitmap and push the surface to it
    RenderTargetBitmap renderBitmap = new RenderTargetBitmap((int)size.Width, (int)size.Height, 96d, 96d, PixelFormats.Pbgra32);
    renderBitmap.Render(surface);

    // Create a file stream for saving image
    try
    {
        using (FileStream outStream = new FileStream(filename, FileMode.Create))
        {
            BmpBitmapEncoder encoder = new BmpBitmapEncoder();
            // push the rendered bitmap to it
            encoder.Frames.Add(BitmapFrame.Create(renderBitmap));
            // save the data to the stream
            encoder.Save(outStream);
        }
    }
    catch (DirectoryNotFoundException)
    {
        Directory.CreateDirectory("C:\Users\" + Environment.UserName + "\My Documents\KinectFitter\Photos");
        using (FileStream outStream = new FileStream(filename, FileMode.Create))
        {
            BmpBitmapEncoder encoder = new BmpBitmapEncoder();
            encoder.Frames.Add(BitmapFrame.Create(renderBitmap));
            encoder.Save(outStream);
        }
    }
}
```
Result in phase 2

The above picture shows the implementation of our application in phase 2.

After doing the body configuration, user can choose clothes from the catalogue. The testing cloth can be successfully dressed on the user’s body. When the user slightly moves around his/her body, the cloth will do the corresponding movement as well.
Mid-term conclusion
Up to this stage, we have implemented a simple, user-friendly GUI, with a set of controls done by capturing motions of users through Kinect. We have used a combination of 3D solids to construct a fair enough human model, and also the cloth model with some physics mechanics theory on it, together doing interaction with each other to successfully simulate a cloth fitting on the user-body, and visualize it on the screen.

In the next stage, we’ll be continuing our focus on fine-tuning our two models. Both accuracy and performance are to be considered to reach an acceptable standard. Besides, more functionality will be added to make the software product more complete. The design of the GUI will be adjusted according to users’ evaluations conducted with voluntary participants as well. Cloth design functionality for users will be implemented depending on the progress.

Mid-term evaluation phase
In order to improve our performance of our application, we had invited several users to try our interim application and give us some feedback.

Focusing on the GUI, there are several drawbacks mentioned by the users that the application should be improved in the next phase:

1. The category button located on the left top corner is actually impressive. The “touch-and-roll” design is interesting, but it is quite plain that it should be redrawn to become more fashionable/ colorful.
2. The “wardrobe panel” (white round-cornered rectangle) located on right bottom corner is too “boring” and occupy nearly half of the screen. User cannot move freely on screen and their body usually is blocked by the “clothing panel”. It should be hidden/ closed in order to free the space for the users to move.
3. The control buttons like tutorial button, photo-taking button and clear button can be grouped in one panel such that user can easily found where there are on screen.
Apart from above feedbacks, we also observed that users want to have a more personalized application rather than a simple tool. The application in this phase is too simple in user interface design which makes no major difference from a real fitting room.

Based on above feedbacks given by several users, we have decided the approach of improvement on our application in next phase:

1. Animation effect should be widely used among different components/panels. These effects would help to arouse the interest of user.
2. The design of the panels and buttons should be more fashionable, more structural and in same design style.
3. The user interface should be more personalized like adding personal profiles, favorite clothes etc, with the support of additional graphical panels.
4. The positioning and presenting of controls on the screen should be designed more carefully.

On the other hand, Microsoft released the Kinect SDK Beta 2. It has a number of under-the-hood improvements. For instance:

1. Improvement in accuracy of skeleton tracing helps to trace the user motion more accurately, e.g. increase the stability of capturing each point in the skeleton. User can find it easier to control each panel on screen.
2. Improvement in speed of transferring skeleton information from Kinect device helps to reduce the latency of tracing user motion. The controlling of each panel can become more synchronous with the user motion.
3. Improvement in stability of the driver and runtime helps the application to perform stably.

After the release of Kinect SDK Beta 2, we had studied numerous of articles about upgrading the code in Beta 1 to Beta 2. We found that it is quite troublesome at first as there are lots of tiny change in codes based on the reference of Microsoft.Research.Kinect.dll and Coding4Fun Kinect Toolkit which leads to increase the work on code modification. Luckily, after several weeks, Microsoft had released newer version of the aforementioned two references which are fully compatible to Kinect SDK Beta 2, and we can completely utilized our code in the new Kinect SDK with slightly changes.
Phase 3:
After the mid-term evaluation phase, we have made several major improvements on our application in following aspects:

1) Graphical design;
2) User-friendliness - User Interface structure;
3) Easy Control - User Interface design;
4) Functionalities;

1. Graphical design
From the beginning, we always want to apply the rule “Simple is the best” to the GUI. However, based on the feedback given by users in mid-term evaluation phase, users are disappointed about the over-simplified GUI and they prefer a well-designed user interface.

Frankly, the application we have completed in phase 2 is like a prototype rather than a complete application, therefore our effort is mainly given to the design of main control function. In this phase, in order to impress our user, the design in color matching and design format should be also considered in order to give a better experience for user to use this application. Therefore, we have studied several boutiques’ designs, such as UNIQLO:

![Design of UNIQLO](http://www.uniqlo.com/us/)

After studying several boutiques’ designs, we have learnt that using a same color base with various luminosity gradients can create simple but impressive color combinations in the user interface. Therefore the graphical design user interface is mainly improved based on this design rule.

For example, the category button in phase 2 is only in black and white with simple design. In this phase, we have redrawn it to have a black color base with several luminosity changes. Also, wardrobe is also redrawn into more realistic so that user can immediately recognize what it is when user uses it.
2. User-friendliness - User Interface structure

In phase 2, we have introduced “hovering button” and “motion detection” into the application. These greatly increase the attractiveness of our application. However, an application cannot become a success with solely fancy effects, it requires a systematical structure in order to help the user to utilize the application in a much easier and more comfortable way.

Based on the feedback from users in mid-term evaluation phase, we can conclude that users prefer to have a more customized and personalized applications. Therefore we have modified the user interface to become more structural with user's personal information included.

Stage 0 and Stage 1- Front page entrance/ First experience of usage:

These stages have been already included in phase 2. Stage 0 is the dressing room with bind closing. The bind would not open until any Kinect device detected, i.e.
any skeleton frames start retrieving from the Kinect device. Once any Kinect device detected, the bind would open and enters stage 1. In stage 1, user is required to move his/her right hand to the center of the screen in order to “enter the dressing room”. User who uses this application for first time can experience how body-motion-based control works and familiar with the use of hovering button to confirm their action.

**Stage 2- User Identification:**

After getting used to the body-motion-based control and hovering button, user would then be advanced to stage 2. In this user identification stage, two types of “dialog boxes” would appear:
The first one is “registered dialog boxes”. In this box, information of previous registered user will be retrieved from the data file and be shown on the dialog boxes. User photo, gender, height and last login time are included in the data file. User can easily identify his/her user profile by the user photos shown in the boxes. On these dialog boxes, there are two buttons: remove button and confirm button. User can directly delete his/her profile by touching the remove button. Otherwise, user can pick his/her profile and enter to the main panel (advance to stage 3). The second one is “new dialog boxes”. In this box, the delete button would be deactivated. User can touch the right “confirm button” in order to start the creation of user profile (advance to stage 4);

Stage 3- Main Function Panel:

When user selected his/her profile, he/she would be brought to the stage 3, which come to the main panel. This stage is similar to the main screen in phase 2, but with aforementioned enhanced graphical control and after-mentioned additional functionalities. It should be highly mentioned that a “Home” function is added such that user is allowed to go back to user identification page (stage 2) in order to switch user
without restart the application. The time would be recorded when he/she leave as the last time login.

Stage 4 - Choose Gender:

If user chooses to create a new user, he/she would be brought to the gender-choosing page. It is a simple page which user can choose the gender.

Stage 5-7 - Body Scale Estimation and Photo Snapping:

After user chooses his/her gender, an instruction would be shown to instruct the user to stay still for body estimation (stage 5). Then the program would hide the instruction, show the position user should stand and count from 9 to 0. User can use these 10 seconds to move to the given position (stage 6). After counting 10 seconds, program would assume the user is standing in the correct position and estimate body scale using skeleton stream and snap the photo for profile picture (stage 7).
Stage 8- Estimation Result

![Stage 8 Result](image)

After collecting all information from user, the application would show the result of the estimation, such as height, shoulder size, chest size and waist size, with the snapped photo. If the user is not satisfied with the body scale estimation or the snapped photo, then he/she can touch the “reset” button and go back to stage 5 to re-estimate the body scale and snap another photo. Otherwise, user can confirm the result by touching the button on the right top corner of the message box.

The main flow is shown below:

![Main Flow Diagram](image)

By introducing the above structure, the application becomes more systematic. This helps the program to control the application flow. User information captured also helps to personalize the application.
3. Easy Control - User Interface design
For the design of positions of controls, we should make sure that each panel and control buttons would not block the screen after being used. The user interface design of the main function page is improved as following:

![Fig 1.14 Closed and opened wardrobe](image1)

**i)** As user reported in the mid-term evaluation phase that “wardrobe panel” block the user view for most of the time, the “wardrobe panel” can now be opened and hidden by touching the newly added “wardrobe handle”. This can allow user to hide the “wardrobe” after selecting clothes which would not block the view of user when trying on.

![Fig 1.15 Closed and opened control panel](image2)

**ii)** Those “control button” such as tutorial button, photo-taking button and clear button are now grouped in one control panel such that user can found where
there are on screen easily. Also, it would also be hidden, i.e. move back to the original place if user does not touch it for more than 5 seconds.

iii) Other controls like the “category button” on left top corner and color panel on the left can also move back to original position automatically if user does not touch it for 5 seconds.

Through these improvements, the screen can leave more space for user to visualize how the clothes are fitted onto their bodies, and help the user to use our application in a much easier way.
4. Functionality

In Phase 3, we have added several functionalities into our application.
1. The newly added user profile can be shown in the main screen.
2. User can save their pattern in the particular file and application would automatically read the pattern pictures and map on the clothes.
3. User can “open” the color panel and change the color of the clothes.

We have also attempted to add the functionality of removal of background. The reason is that user can take a photo with his/ her clothes in different background. We can utilize the depth stream and video stream provided by the Kinect device to remove of background. The program would read each byte on depth stream and extract the required bit which is within a depth range. Then it would find all pixels in the color image which matches the required coordinate from the depth image.

The code for removal of background is:

```csharp
PlanarImage depthImage = new PlanarImage();
WriteableBitmap target = new WriteableBitmap(depthWidth, depthHeight, 96, 96, PixelFormats.Bgra32, null);
var depthRect = new System.Windows.Int32Rect(0, 0, depthWidth, depthHeight);

runtime.DepthFrameReady += (s, e) =>
{
    depthImage = e.ImageFrame.Image;
};

runtime.VideoFrameReady += (s, e) =>
{
    if (depthImage.Bits == null) return;
    byte[] color = e.ImageFrame.Image.Bits;
    byte[] output = new byte[depthWidth * depthHeight * 4];

    // loop over each pixel in the depth image
    int outputIndex = 0;
    for (int depthY = 0, depthIndex = 0; depthY < depthHeight; depthY++)
    {
        for (int depthX = 0; depthX < depthWidth; depthX++, depthIndex += 2)
        {
            int player = depthImage.Bits[depthIndex] & 7;

            // find a pixel in the color image which matches this coordinate from the depth image
            int colorX, colorY;
        }
    }
```

Eventually this function is not applied to our application. It is because the depth stream retrieved from the Kinect device is still not good enough to cut out the shape of the user even we have upgraded the Kinect SDK to Beta 2. For instance, some part of our head could not be shown in the picture. The reason would be the user hair will reflect the IR light, CMOS sensor cannot capture the light and treat that part of the head as background.

Fig 1.17 Problem of background removal using depth stream
Conclusion

All in all, this application has been improved into a comparatively acceptable application to provide a virtual fitting room for user to utilize. With the assistance of our group mates, our application has been improved in several performances:

1. A human model generated according to user body size and gender;
2. An impressive, flexible and real-looking cloth model for user to “wear”.
   With the support of parallel programming of GPU, the cloth model is more synchronized with user motion;
3. A fashionable body-motion-based GUI with basic structure for user to utilize;
4. A lot of interesting and useful functionalities for user to utilize in our application.
References

GUI:

http://channel9.msdn.com/coding4fun/kinect/Kinecting-the-Dots-Adding-Buttons-to-your-Kinect-
Application Kinecting the Dots: Adding Buttons to your Kinect Application

http://blogs.msdn.com/b/tess/archive/2011/08/16/kinect-sdk-for-windows-hover-button-hover-
control.aspx Kinect SDK for Windows ¡V Hover Button / Hover Control

Display-Kinect-color-image-containing-only-players-aka-background-
removal&h=dAQEXsx7WAQF5Ron3MMCiKTGvM5aL_8aEY6tqNFFnAAkzoQ Display Kinect color
image containing only players (aka background removal) (Channel 9)

http://www.uniqlo.com/us/ UNIQUO Mainpage

Kinect SDK:


http://research.microsoft.com/en-
us/um/redmond/projects/kinectsdk/docs/ProgrammingGuide_KinectSDK.pdf Programming Guide of
Kinect SDK

Kinect SDK Beta 2

from Kinect for Windows SDK Beta 1 to Beta 2 | Smart++;

Windows Presentation Foundation (WPF):

http://www.wpftutorial.net/IntroductionTo3D.html WPF Tutorial.net