Chapter No. 1
"Understanding the Kinect Device"
In this package, you will find:
A Biography of the author of the book
A preview chapter from the book, Chapter NO.1 "Understanding the Kinect Device"
A synopsis of the book’s content
Information on where to buy this book

About the Author

Abhijit Jana works at Microsoft as a .NET Consultant, as part of Microsoft Services Global Delivery, India. As a consultant, his job is to help customers design, develop, and deploy enterprise-level secure solutions using Microsoft Technology. Apart from being a former Microsoft MVP, he is a speaker and author as well as an avid technology evangelist. He has delivered sessions at prestigious Microsoft events such as TechEd, Web Camps, Azure Camps, Community Tech Days, Virtual Tech Days, DevDays, and developer conferences. He loves to work with different .NET communities and help them with different opportunities. He is a well-known author and has published many articles on different .NET community sites.

He shares his thoughts on his personal blog at http://abhijitjana.net. You can follow him on Twitter at @abhijitjana. Abhijit lives in Hyderabad, India, with his wife, Ananya and a beautiful little angel Nilova.

For More Information:
Kinect for Windows SDK Programming Guide

Ever since its inception, Kinect has brought about a revolution in the field of NUI and hands-free gaming. There is no wonder that Kinect went on to shatter all records and become the fastest selling electronic device on earth. Although touted as a controller for Xbox console, Kinect applicability is beyond gaming domain and you can think of building applications for diverse domains such as health care, robotics, imaging, education, security, and so on. Thus we have the Kinect for Windows sensor, that enables applications to interacts with users via gestures and voice, and this opens up avenues that developers couldn't even have imagined before.

This book is mainly focussed on the Kinect for Windows SDK with which you can build applications that can leverage the power of the Kinect sensor. This book doesn't require any prior knowledge about the platform from the reader and its strength is the simplicity in which the concepts have been presented using code snippets, a step-by-step process, and detailed descriptions. This book covers:

- A practical step-by-step tutorial to make learning easy for a beginner
- A detailed discussion of all the APIs involved and the explanations of their usage in detail
- Procedures for developing motion-sensing applications and also methods used to enable speech recognition

What This Book Covers

Chapter 1, Understanding the Kinect Device, introduces Kinect as a hardware device. You will get an insight into the different components that make up Kinect and the technology behind this device, which makes the components work together. This chapter will also give an overview of the difference between Kinect for Xbox and Kinect for Windows sensor. You will also become familiar with different possibilities of domain specific applications that can be developed using the Kinect sensor.

For More Information:
Chapter 2, Getting Started, introduces the Kinect for Windows SDK, its features, and how to start working with the Kinect sensor. In this chapter, you will get to know about the requirements for preparing your development environment. This will also walk you through a step-by-step guide for downloading and installing the SDK. You will delve into the installed components to verify that everything is set up properly. This chapter will also provide you with a quick lap around the different features of the Kinect for Windows SDK as well as introduce the Kinect for Windows Developer Toolkit.

Chapter 3, Starting to Build Kinect Applications, explains the step-by-step process of building your first Kinect-based application. You will understand how applications interact with the Kinect sensor using the SDK libraries. This chapter will give you an in-depth guide on how to start building Kinect applications using the Kinect for Windows SDK and Visual Studio. You will also learn how to deal with applications when there is any change in the device status.

Chapter 4, Getting the Most Out of Kinect Camera, covers the in-depth discussion of the Kinect color camera and how to use it. In this chapter, you will learn about the different types of image streams and different approaches to retrieve them from the Kinect sensor. You will get an understanding of Color camera stream pipeline and its events. You will also explore the different features of the Kinect for Windows SDK that control the color camera and process the color data. This chapter will give you an understanding of processing color images and applying different effects to the captured images and how to save the image frames. You will also learn how you can use the Kinect camera to capture images in low light.

Chapter 5, The Depth Data – Making Things Happen, explores the fundamentals of the Kinect depth sensors and how they produce depth information. This chapter describes how to work with object distances and player indices from the captured depth data. You will also learn about the capturing of data using the near mode and also get a quick view of generating 3D depth data.

For More Information:
Chapter 6, Human Skeleton Tracking, describes how a Kinect sensor tracks the human skeleton and how you can leverage the features of the Kinect for Windows SDK to play around with tracked skeletons and joints. You will also learn how to change the sensor elevation angle based on the player position. This chapter also explores how skeletons can be tracked in a seated mode. You also learn about details of the skeleton joints and bone hierarchy. The sample application in this chapter will help you to understand the APIs for skeleton tracking in better ways such as using Kinect as an intrusion detector. At the end of this chapter, you will be familiar with a few debugging tips and tricks to boost your development speed.

Chapter 7, Using Kinect's Microphone Array, introduces the microphone array that captures and processes the audio signal. You will learn why Kinect uses an array of microphones rather than a single microphone. In this chapter you will get an insight into the Kinect audio processing pipeline that helps Kinect to capture good-quality audio signals and makes Kinect a highly directional audio device. This chapter provides you with information on how to capture and record audio signals using the Kinect microphone array and process the audio data for better quality. You will also learn about different concepts such as Noise Suppression, Automatic Gain Control, Echo Cancellation, and Beam forming.

Chapter 8, Speech Recognition, introduces the building of speech-enabled applications using Kinect. You will explore how speech recognition works and how Kinect's microphone array helps Kinect to recognize human speech. This chapter also shows how you can use Kinect as the default speech recognition device for your PC. You will also learn about the Microsoft Speech API and how it is integrated with Kinect for Windows SDK, which helps us to build speech-enabled applications.

For More Information:
Chapter 9, Building Gesture-controlled Applications, describes how to build applications that can be controlled by human gestures. You will learn different approaches for recognizing gestures and how to apply these approaches in the form of programs to build motion-sensing applications using the Kinect sensor. This chapter will also help you understand how to build some gestured-enabled controls.

Chapter 10, Developing Applications Using Multiple Kinects, explains how multiple Kinect sensors can be placed together and used to build applications. This chapter describes how to set up environments for developing applications using multiple Kinects and walks you through building applications by reading data from multiple devices. You also learn how multiple Kinects work together and different scenarios where multiple Kinects can be used, along with the challenges while developing applications using multiple devices.

Chapter 11, Putting Things Together, introduces us to more advanced developments using Kinect by integrating it with other devices such as Windows Phone, microcontrollers, and so on. This chapter addresses how we can take things up from Kinect to Windows Azure and control the Kinect sensor using Windows Phone via Windows Azure. You will also learn how Kinect can be integrated with the Netduino microcontroller and how you can use a Kinect device for face tracking.

For More Information:
Welcome to the world of motion computing with Kinect. Kinect was originally known by the code name "Project Natal". It is a motion-sensing device which was originally developed for the Xbox 360 gaming console. One of the distinguishing factors that makes this device stand out among others in this genre is that it is not a hand-controlled device, but rather detects your body position, motion, and voice. Kinect provides a **Natural User Interface (NUI)** for interaction using body motion and gesture as well as spoken commands. Although this concept seems straight out of a fairytale, it is very much a reality now. The controller that was once the heart of a gaming device finds itself redundant in this Kinect age. You must be wondering where its replacement is. The answer, my friend, is YOU. It's you who is the replacement for the controller, and from now on, you are the controller for your Xbox. Kinect has ushered a new revolution in the gaming world, and it has completely changed the perception of a gaming device. Since its inception it has gone on to shatter several records in the gaming hardware domain. No wonder Kinect holds the Guinness World Record for being the "fastest selling consumer electronics device". One of the key selling points of the Kinect was the idea of "hands-free control", which caught the attention of gamers and tech enthusiasts alike and catapulted the device into instant stardom. This tremendous success has caused the Kinect to shatter all boundaries and venture out as an independent and standalone, gesture-controlled device.

It has now outgrown its Xbox roots and the Kinect sensor is no longer limited to only gaming. Kinect for Windows is a specially designed PC-centric sensor that helps developers to write their own code and develop real-life applications with human gestures and body motions. With the launch of the PC-centric Kinect for Windows devices, interest in motion-sensing software development has scaled a new peak.

For More Information:
As Kinect blazed through the market in such a short span of time, it has also created a necessity of resources that help people learn the technology in an appropriate way. As Kinect is still a relatively new entry into the market, the resources for learning how to develop applications for this device are scant. So how does a developer understand the basics of Kinect right from scratch? Here comes the utility of this book.

This book assumes that you have basic knowledge of C# and a great enthusiasm to program for Kinect devices. This book can be enjoyed by anybody interested in knowing more about the device and learning how to interact with devices using Kinect for Windows **Software Development Kit (SDK)**. This book will also help you explore how to process video depth and audio stream, and build applications that interact with human body motion. The book has deliberately been kept simple and concise, which will aid in the quick grasping of the concepts.

Before delving into the development process, we need a good understanding of the device and, moreover, what the different types of applications are, which we can develop using these devices. In order to develop standard applications using the Kinect for Windows SDK, it is really important for us to understand the components it interacts with.

In this chapter we will cover the following topics:

- Identifying the critical components that make up Kinect
- Looking into the functionalities of each of the components
- Learning how they interact with each other
- Choosing between Kinect for Windows and Kinect for Xbox
- Exploring different application areas where we can use Kinect

**Components of Kinect for Windows**

Kinect is a horizontal device with depth sensors, color camera, and a set of microphones with everything secured inside a small, flat box. The flat box is attached to a small motor working as the base that enables the device to be tilted in a horizontal direction. The Kinect sensor includes the following key components:

- Color camera
- Infrared (IR) emitter
- IR depth sensor
- Tilt motor
- Microphone array
- LED

Apart from the previously mentioned components, the Kinect device also has a power adapter for external power supply and a USB adapter to connect with a computer. The following figure shows the different components of a Kinect sensor:

![Kinect components](image)

**Inside the Kinect sensor**

From the outside, the Kinect sensor appears to be a plastic case with three cameras visible, but it has very sophisticated components, circuits, and algorithms embedded. If you remove the black plastic cover from the Kinect device, what will you see? The hardware components that make the Kinect sensor work.

The following image shows a front view of a Kinect sensor that’s been unwrapped from its black case. Take a look (from left to right) at its IR emitter, color camera, and IR depth sensor:

![Kinect front view](image)

Let’s move further and discuss about component.

---

For More Information:
The color camera

This color camera is responsible for capturing and streaming the color video data. Its function is to detect the red, blue, and green colors from the source. The stream of data returned by the camera is a succession of still image frames. The Kinect color stream supports a speed of 30 frames per second (FPS) at a resolution of 640 x 480 pixels, and a maximum resolution of 1280 x 960 pixels at up to 12 FPS. The value of frames per second can vary depending on the resolution used for the image frame.

The viewable range for the Kinect cameras is 43 degrees vertical by 57 degrees horizontal. The following figure shows an illustration of the viewable range of the Kinect camera:

![Viewable range diagram](image)

The following image shows a color image that was captured using Kinect color sensors with a resolution of 640 x 480 pixels:

![Color image](image)
IR emitter and IR depth sensor

Kinect depth sensors consist of an IR emitter and an IR depth sensor. Both of them work together to make things happen. The IR emitter may look like a camera from the outside, but it’s an IR projector that constantly emits infrared light in a "pseudo-random dot" pattern over everything in front of it. These dots are normally invisible to us, but it is possible to capture their depth information using an IR depth sensor. The dotted light reflects off different objects, and the IR depth sensor reads them from the objects and converts them into depth information by measuring the distance between the sensor and the object from where the IR dot was read. The following figure shows how the overall depth sensing looks:

It is quite fun and entertaining to know that these infrared dots can be seen by you. All we need is a night vision camera or goggles.

The depth data stream supports a resolution of 640 x 480 pixels, 320 x 240 pixels, and 80 x 60 pixels, and the sensor viewable range remains the same as the color camera.
Understanding the Kinect Device

The following image shows depth images that are captured from the depth image stream:

How depth data processing works

The Kinect sensor has the ability to capture a raw, 3D view of the objects in front of it, regardless of the lighting conditions of the room. It uses an infrared (IR) emitter and an IR depth sensor that is a monochrome CMOS (Complimentary Metal-Oxide-Semiconductor) sensor. The backbone behind this technology is from PrimeSense, and the following diagram shows how this works:

For More Information:
The sequence explained in the diagram is as follows:

When there is a need to capture depth data, the PrimeSense chip sends a signal to the infrared emitter to turn on the infrared light (1), and sends another signal to the IR depth sensor to initiate depth data capture from the current viewable range of the sensor (2). The IR emitter meanwhile starts sending an infrared light invisible to human eyes (3) to the objects in front of the device. The IR depth sensor starts reading the inferred data from the object based on the distance of the individual light points of reflection (4) and passes it to the PrimeSense chip (5). The PrimeSense chip then analyzes the captured data, and creates a per-frame depth image and passes it to the output depth stream as a depth image (6).

The IR emitter emits an electromagnetic radiation. The wavelengths of the radiations are longer than the wavelength of the visible light, which makes the sensor's IR lights invisible. The wavelengths need to be consistent to minimize the noise within the captured data. Heat generated by the laser diode when the Kinect sensor is running can impact the wavelength. The Kinect sensor has a small, inbuilt fan to normalize the temperature and ensure that the wavelengths are consistent.

**Tilt motor**

The base and body part of the sensor are connected by a tiny motor. It is used to change the camera and sensor’s angles, to get the correct position of the human skeleton within the room. The following image shows the motor along with three gears that enable the sensor to tilt at a specified range of angles:

For More Information:

The motor can be tilted vertically up to 27 degrees, which means that the Kinect sensor's angles can be shifted upwards or downwards by 27 degrees. The following figure shows an illustration of the angle being changed when the motor is tilted:

Do not physically force the device into a specific angle. The Kinect for Windows SDK has a few specific APIs that can help us control the sensor's motor tilting. Do not tilt the Kinect motor frequently; use this as few times as possible and only when it's required.

**Microphone array**

The Kinect device exhibits great support for audio with the help of a microphone array. The microphone array consists of four different microphones that are placed in a linear order (three of them are spread on the right side and the other one is placed on the left side, as shown in the following image) at the bottom of the Kinect sensor:

For More Information:

The purpose of the microphone array is not just to let the Kinect device capture the sound but to also locate the direction of the audio wave. The main advantages of having an array of microphones over a single microphone are that capturing and recognizing the voice is done more effectively with enhanced noise suppression, echo cancellation, and beam-forming technology. This enables Kinect to be a highly bidirectional microphone that can identify the source of the sound and recognize the voice irrespective of the noise and echo present in the environment.

LED
An LED is placed in between the camera and the IR projector. It is used for indicating the status of the Kinect device. The green color of the LED indicates that the Kinect device drivers have loaded properly. If you are plugging Kinect into a computer, the LED will start with a green light once your system detects the device; however for full functionality of your device, you need to plug the device into an external power source.

Kinect for Windows versus Kinect for Xbox
Although "Kinect for Windows" and "Kinect for Xbox" are similar in many respects, there are several subtle differences from a developer's point of view. We have to keep in mind that the main purpose of Kinect for Xbox was to enhance the gaming experience of the players. Developing applications was not its primary purpose. In contrast, Kinect for Windows is primarily a developing device and not for gaming purposes.
You can develop applications that use either the Kinect for Windows sensor or the Kinect for Xbox sensor. The Kinect for Xbox sensor was built to track players that are up to 12 feet (4.0 meters) away from the sensor. But it fails to track objects that are very close (80 cm), and we might need to track objects at a very close range for different applications. The Kinect for Windows sensor has new firmware, which enables Near Mode tracking. Using Near Mode, Kinect for Windows supports the tracking of objects as close as 40 cm in front of the device without losing accuracy or precision. In terms of range, both the sensors behave the same.

Kinect for Windows SDK exposes APIs that can control the mode of the sensors (Near Mode or Default Mode) using our application, however the core changes for this feature are built within the firmware of the Kinect for Windows sensor.

Both the Kinect for Windows and Kinect for Xbox sensors need additional power for the sensors to work with your PC. This might not be required when connected to the Xbox device as the Xbox port has enough power to operate the device. There is no difference between Xbox Kinect and Kinect for Windows in this respect. However in Kinect for Windows, the USB cable is small and improved to enable more reliability and portability across a wide range of computers.

And finally, the Kinect for Windows sensor is for commercial applications, which means that if you are developing a commercial application, you must use the Kinect for Windows device for production, whereas you can use Kinect for Xbox for general development, learning, and research purposes.

**Where can you use Kinect**

By now it has already struck you that this is something more than just gaming. The Kinect sensor for Windows and the Kinect for Windows SDK unwrap a new opportunity for the developer to build a wide range of applications. These can include:

- Capturing real-time video using the color sensor
- Tracking a human body and then responding to its movements and gestures as a natural user interface
- Measuring the distances of objects and responding

For More Information:

Chapter 1

- Analyzing 3D data and making a 3D model and measurement
- Generating a depth map of the objects tracked
- Recognizing a human voice and developing hands-free applications that can be controlled by voice

With this you can build a number of real-world applications that fall under a different domain. The following are a few examples, which will help you understand the applicability of Kinect sensors:

- **Healthcare**: Using the Kinect sensor, you can build different applications for healthcare, such as exercise measurement, monitoring patients, their body movements, and so on
- **Robotics**: Kinect can be used as a navigation system for robots either by tracking human gestures, voice commands, or by human body movements
- **Education**: You can build various applications for students and kids to educate and help them to learn subjects either by their gesture and voice commands
- **Security system**: Kinect can be used for developing security systems where you can track human body movement or face and send the notifications
- **Virtual Reality**: With the help of Kinect 3D technology and human gesture tracking, several virtual reality applications can be build using the Kinect sensor
- **Trainer**: Kinect can potentially be used as a trainer by measuring the movements of human body joints, providing live feedback to users if the joints are moving in an appropriate manner by comparing the movements with previously stored data
- **Military**: Kinect can be used to build intelligent drones to spy on enemy lines

Well these were just a few specific examples of domains where you can use Kinect, but at the end of the day it's up to your imagination; where and how you want this device to work.

For More Information:
Summary

This chapter gave you an inside look at the different components of the Kinect sensor. You saw that the major components of a Kinect device are its color sensor, IR depth sensors, IR emitter, microphone arrays, and a stepper motor that can be tilted to change the Kinect camera angles. While the color sensor and depth sensors ensure video and depth data input, which is of prime importance for the functioning of the device, the microphone arrays on the other hand ensure that the audio quality is also at par. Also worthwhile is mentioning about how Kinect processes the depth data, and the array of microphones, which is a design novelty that helps in clear voice recognition with the use of the noise suppression and echo cancelation mechanisms. Kinect for Windows is also capable of tracking humans at a close range of approximately 40 centimeters using Near Mode. It wouldn’t be wrong to say that it is this combination of technological innovations that make Kinect the awe-inspiring device that it is. You have also gone through the different possibilities of applications that can be developed using Kinect. In the next chapter, we will walk you through the step-by-step installation and configuration of the development environment setup along with different troubleshooting tips and tricks that will help you to be sure about everything before beginning with development.
Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.