

DENEY NO: 8

KINECT İLE İNSAN HAREKETLERİNİN ALGILANMASI

1. GİRİŞ

Eğitim ve öğretimde teknolojik araçların kullanılması, zenginleştirilmiş öğrenme ortamlarının oluşturulması, sürecin kalite ve verimliliğinin artırılması açısından önemlidir. Özellikle bireyselleştirilmiş öğrenme ortamlarının yapılandırılmasında pek çok avantaj sunan teknolojik araçların özel eğitimde kullanılmasının önemi dikkat çekmektedir. Özel eğitim gereksiniminde olan iletişim, davranış ve öğrenme özellikleri teknoloji ile zenginleştirilerek eğitiminde kullanılması, öğrenme çıktılarının iyileştirilmesi ve yaşam kalitesinin artırılması açısından destekleyici olmaktadır.

1.1.Amaç

Kamera karşısında duran insana ait pozisyonların tespit edilerek bunlarla ilgili çıkarımlar üretmektir. Örneğin herhangi bir rahatsızlığından dolayı egzersiz yapan bir insanın yaptığı hareketlerin doğruluğunu tespit etmek onun tedavi süreci için önemlidir.

1.2.Deneyin Tanıtımı

Kinect'in (IR) Kızılötesi sensörü yardımıyla görüş alanındaki altı kullanıcıya kadar tanımlama yapılabilir. Fakat bunlardan en fazla iki tanesi ayrıntılı olarak izlenebilir. Bu kullanıcıların eklemlerini bularak onların hareketlerini gözlemlemek mümkündür.

2. Kinect Teknolojisi

Microsoft Xbox 360 ve Windows uyumlu Kinect sensörü son derece gelişmiş algılama sensör donanımları içermektedir; derinlik kamera sensörü, RGB kamera sensörü, dörtlü mikrofon grubu ile üç boyutlu şekilde hareketleri belirtilen aralıklarla yakalama, yüz tanıma ve ses algılama imkanı sunmaktadır. Kinect yazılım geliştirme kitinde bir iskelet, insan vücudundaki bulundukları yerlere göre numaralandırılmış 20 adet eklemden oluşur ve bu eklemlerin pozisyonları ve uzaklıkları takip edilerek vücut hareketlerini görüntüye işlemektedir.

Kinect kamerası kullanıcıyı 80 cm ile 400 cm (4 m) arasındayken algılayabilmektedir. En ideal uzaklık 120 cm ile 350 cm arasındadır. Kullanıcının yüzünün olduğu ön tarafının kameraya dönük halinde istenilen şekilde çalışmaktadır. Kamera kullanıcı 350 cm uzaklıktayken ortalama 2 metrelik bir insan boyunu algılayabilmektedir. Kinect For Windows kamerası, Skeleton kütüphanesi yardımıyla bu eklemlerin, eklem tipi ve pozisyon (X,Y,Z) bilgisi döndürülür.

Kinect'in üzerinde 3 adet göz, sıra mikrofonlar ve hareket sağlayıcı bir motor mekanizması bulunmaktadır. Soldaki göz lazer projeksiyonu yaparken, sağdaki kızılötesi sensör bu ışınların gidiş - geliş süresini hesaplayarak 320x240 çözünürlüğünde her bir noktanın mesafesini bildirmektedir. Kinect firmware'i ise bu veriler ışığında iskelet yapısını hesaplamakta ve bunu XBOX ya da PC'ye göndermektedir.

2.1.Uygulama için Ön Gereksinimler

Kinect SDK ile sensör ve kamera özelliklerine erişebilmek için:
- Çift çekirdekli 2.66 Ghz işlemci

- En az 2 GB RAM
- 32 yada 64 bit Windows 7-10 işletim sistemi ve .NET Framework 4.0' a sahip olmanız gerekmektedir.

2.2.Uygulama Geliştirmek için:

- Visual Studio 201x'un herhangi bir sürümüne sahip olmanız,
- Microsoft Kineck SDK'yı indirmeniz, (buradan indirebilirsiniz)
<http://www.microsoft.com/en-us/search/DownloadResults.aspx?q=kinect+sdk>

Not: Yazılımınızı indirdiğiniz SDK'ya uygun olarak kodlamanız gerekmektedir.

- İsteğe bağlı olarak bazı metodlar için:

OpenNi/PrimeSense kütüphaneleri için (buradan indirebilirsiniz)

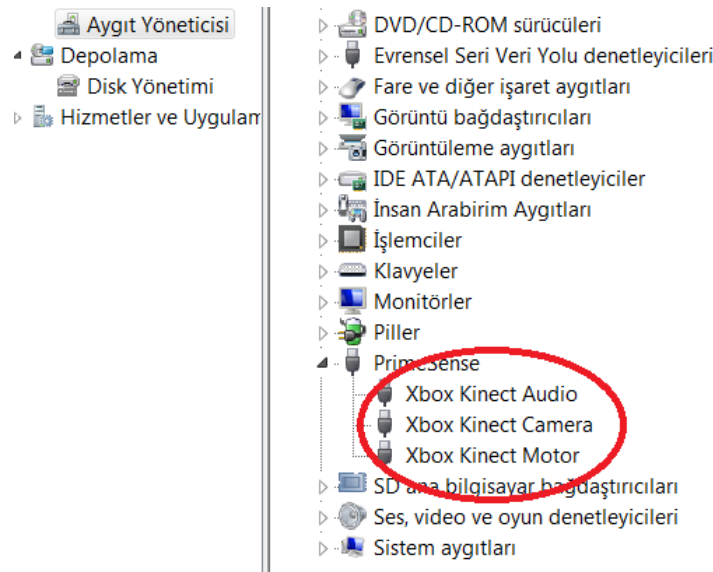
<http://fivedots.coe.psu.ac.th/~ad/kinect/installation.html>

-Coding4Fun Kinect Toolkit'i indirmeniz için. (buradan indirebilirsiniz)

<http://c4fkinect.codeplex.com/>

3. Donanım Kurulumu

Kinect sensörünün bağlantı portu yeni nesil XBOX 360'lar için tasarlanmıştır. PC ile kullanabilmek için özel adaptörünü kullanmamız gerekiyor. Adaptör kablosunun dişi ucunu Kinect'e, USB ucunu ise PC'ye bağladığımızda, Windows aygıt sürücülerini yüklemeye başlayacak. Sürücüler yüklendikten sonra, aygıt yöneticisinde Microsoft Kinect ve ses denetleyicileri altına veya ilgili kütüphanelerin olduğu dizine Xbox Kinect'in cihazları resim 1'de görüldüğü gibi eklenecektir.



Resim 1: Kurulu Kinect'in aygıt yöneticisinde görünümü.

Kamera donanımının aygıt yöneticisinin görüntü aygıtları arasında yer almamasının nedeni, Kinect kamerasının aslında fotoğraf makinesi gibi davranıp görüntüyü resimler halinde iletmesidir. Bu sayede uygulama içerisinde görüntüye çok daha kolay ulaşabiliyoruz. Kinect'in yüksek hassaslıktaki mikrofonunu ise Windows uygulamalarında kullanabiliyoruz.

4. Uygulama Geliştirmek

Kinect SDK ile C#, VB.NET ve C++ dilleriyle WPF ve Windows Forms uygulamaları geliştirebilmekteyiz. Örnek bir uygulamada C# ile bir WPF uygulaması yapılacağı varsayılırsa öncelikle, Visual Studio ile oluşturduğumuz WPF uygulamamızın referansları arasında

"Microsoft.Research.Kinect.dll" ve "Coding4Fun.Kinect.Wpf.dll" bileşenleri eklenmelidir. Kinect SDK'sına ve yardımcı metodlara ulaşmak için ise using statementlar arasına:

```
using Microsoft.Research.Kinect.Nui;  
using Coding4Fun.Kinect.Wpf;
```

satırlarını eklememiz gerekecektir. Bu aşamadan sonra Kinect'in özelliklerine ve yardımcı metodlarına ulaşabiliyor olacağız.

4.1.Kamera Görüntüsü Alma

MainWindow içerisine eklenen Image elementini kullanarak kamera görüntüsüne ulaşılmaktadır. Öncelikle Kinect runtime'ı oluşturacağız, runtime ayarlarında yalnızca görüntü kullanacağımızı bildireceğiz, her bir görüntü karesi aktarıldığında tetiklenecek eventi belirleyeceğiz, görüntü akışını açıp gelen görüntüleri Image elementine aktaracağız.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Windows;  
using System.Windows.Controls;  
using System.Windows.Data;  
using System.Windows.Documents;  
using System.Windows.Input;  
using System.Windows.Media;  
using System.Windows.Media.Imaging;  
using System.Windows.Navigation;  
using System.Windows.Shapes;  
using Microsoft.Research.Kinect.Nui;  
using Coding4Fun.Kinect.Wpf;  
  
namespace KinectSensorKameraErisimi  
{  
    public partial class MainWindow : Window  
    {  
        public MainWindow()  
        {  
            InitializeComponent();  
        }  
  
        Runtime kinectNui = new Runtime();  
  
        private void Window_Loaded(object sender, RoutedEventArgs e)  
        {  
            // Yalnızca renkli görüntü aktarımı yapacak şekilde Kinect runtime'ı başlat  
            kinectNui.Initialize(RuntimeOptions.UseColor);  
  
            // Renkli görüntü iletiminde tetiklenecek event  
            kinectNui.VideoFrameReady +=  
                new EventHandler<ImageFrameReadyEventArgs>(kinectNui_VideoFrameReady);  
  
            // Renkli görüntü aktarımını başlat  
            kinectNui.VideoStream.Open(ImageStreamType.Video, // Video yayını  
                2, // Önbellekteki resim sayısı  
                ImageResolution.Resolution640x480, // Video çözünürlüğü  
                ImageType.Color // Yayınlanan video türü
```

```

);

}

void kinectNui_VideoFrameReady(object sender, ImageFrameReadyEventArgs e)
{
    // Gelen görüntüyü Image elementinde göster
    imgVideo.Source = e.ImageFrame.ToBitmapSource();
}

private void Window_Closed(object sender, EventArgs e)
{
    // Kaynakları serbest bırak
    kinectNui.Uninitialize();
}
}
}

```

Uygulamayı çalıştırdığımızda renkli kamera görüntüsünü Image elementi içerisinde görebiliriz. Microsoft SDK CoordinateMapper adında kullanışlı bir yardımcı program ile kullanıma hazır geliyor, CoordinateMapper 3B uzayda renkli bir nokta veya derinliği 2D uzayda aynı noktasına karşılık olup olmadığını tespit etmektedir. CoordinateMapper KinectSensor ile bu bağlamda kullanılabilecek bir fonksiyondur.

İnsan eklemlerin koordinatlarına erişen C # kodu:

```

foreach (Joint joint in body.Joints)
{
    // 3D coordinates in meters
    CameraSpacePoint cameraPoint = joint.Position;
    float x = cameraPoint.X;
    float y = cameraPoint.Y;
    float z = cameraPoint.Z;
}

```

Koordinatlar CameraSpacePoint yapı içine paketlenmiş 3D noktaları temsil etmektedir. Her CameraSpacePoint X, Y ve Z değerlerine sahiptir. Bu değerler, metre cinsinden ölçülür. Görsel öğelerin boyutları piksel olarak ölçülür, bu yüzden bir şekilde 2D ekran piksel içine gerçek dünya 3D değerleri dönüştürmek gerekir. Kinect SDK bu dönüşüm için ColorSpacePoint ve DepthSpacePoint fonksiyon imkanı sunar.

İskelet bağlantı noktalarını birleştirmek için:

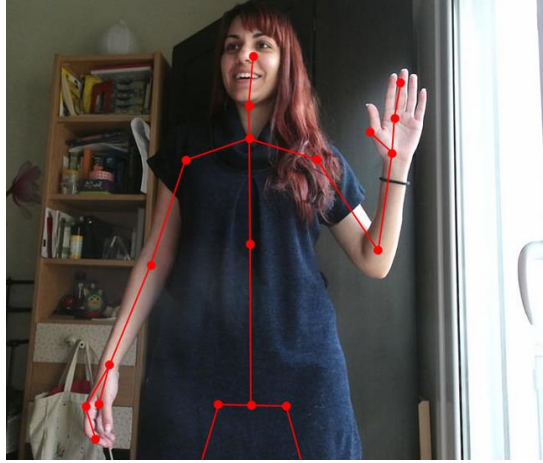
```

public void DrawPoint(ColorSpacePoint point)
{
    // Create an ellipse.
    Ellipse ellipse = new Ellipse
    {
        Width = 20,
        Height = 20,
        Fill = Brushes.Red
    };
    // Position the ellipse according to the point's coordinates.
    Canvas.SetLeft(ellipse, point.X - ellipse.Width / 2);
    Canvas.SetTop(ellipse, point.Y - ellipse.Height / 2);

    // Add the ellipse to the canvas.
    canvas.Children.Add(ellipse);
}

```

Derinlik çerçevesinin üzerinde bir DepthSpacePoint çizebilirsiniz. Ayrıca iki nokta arasındaki kemikleri (çizgiler) çizebilirsiniz. Renkli görüntünün üstüne bir koordinat haritalama da ekleyebilirsiniz.



İskelet tahmini için elde edeceğiniz görüntü yaklaşık resimde görüldüğü gibi olacaktır.

DENEYLER:

Duruş pozisyonuna ait bilgilerin elde edilmesinin ardından yapılması gereken çalışmalar:

- 1- Örnek bir trafik polisi hareketine ait tespitin gerçekleştirilmesi (1,-7).
- 2- Kamera karşısındaki kişiye ait boy uzunluğunun hesaplanması.
- 3- Örnek bir egzersiz hareketine ait tespitin gerçekleştirilmesi (1,-6).
- 4- Kamera karşısındaki kişiye ait kulaç uzunluğunun hesaplanması.
- 5- Örnek bir hakem hareketine ait tespitin gerçekleştirilmesi (1,-10).

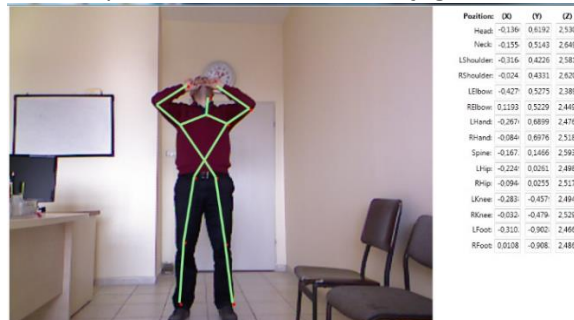
Çıkarılan iskelet bilgileri üzerinden hareket tespitin gerçekleştirme tekniği öğrencinin insiyatifindedir. Nasıl gerçekleştirdiği konusundaki çalışmalarını deney sırasında aktarması gerekmektedir.

Koordinatları çıkarılan örnek görüntüler aşağıdadır:



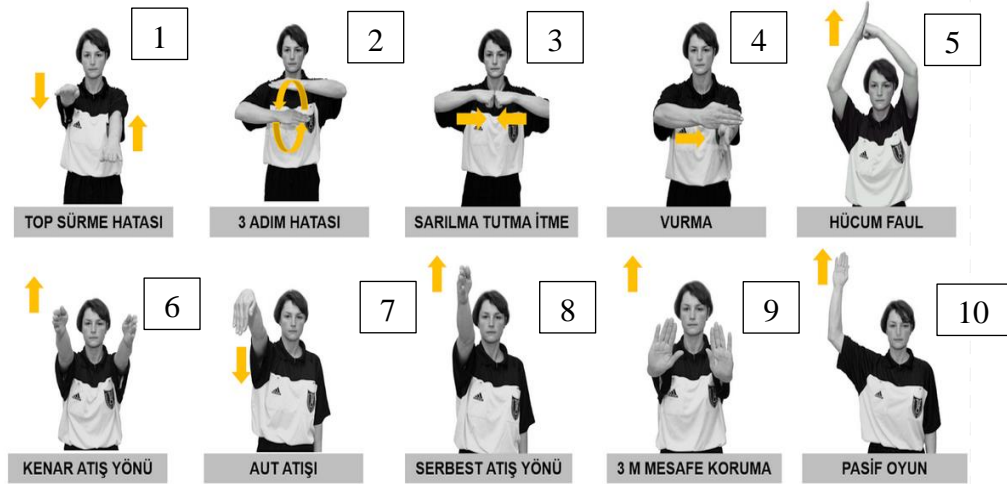
Trafığe açık alan gösteren trafik polisi.

Kenar atışı gösteren maç hakemi.



Boyun egzersizi yapan kişi.

TRAFİK POLİSİNİN HAREKETLERİ



5. Kaynaklar

- <http://filepi.com/i/HMUop4Y> (Beginning Kinect Programming with the Microsoft Kinect SDK)
- Ayrıca yukarıda uygulama geliştirmek için kaynak olarak verilen linkler.