İSTANBUL ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ DEPARTMANI

Robotik Uygulamalar Bitirme Projesi

Burhan ARAS

Istanbul University Computer Engineering

aras@burhanaras.net

27.05.2009

İSTANBUL

ÖNSÖZ

Bu çalışma İstanbul Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde yapılan **"Wireless and Mobile Robotics Application"** adlı lisans tez çalışmasını içermektedir.

Çalışmalarım süresince bana rehberlik eden danışman hocam Dr. Halim Zaim'e teşekkür ederim.

Ayrıca çalışmayı yürütürken teknolojilerinden faydalandığım Microsoft ve Lego Corp. şirketlerine ve hayatımın her evresinde beni destekleyen değerli aileme teşekkürü bir borç bilirim.

Haziran,2009

İçindekiler

Şekil ve Tablo listesi

Özet

Robotik teknolojisi , insan zekasını model alarak her geçen gün daha da geliştirilmektedir. Bu proje kapsamında ülkemizde var olan robot teknolojisine katkıda bulunmak ve bu teknolojilerin kullanılabilirliğini artırmak amacıyla bazı robotik çalışmalar yapılmıştır. Bu proje geliştirilirken Microsft Robotics Studio yazılımı kullanılmış ve donanım olarak ta Lego Mindstorms NXT donanım seti kullanılmıştır.

Abstract

Robotics tehcnology is being developed exponentially day to day due to modelling human intelligence. In this research paper, there has been done some robotics applications to help developing and increasing familiarity with such technologies in Turkey. This project has been developed in Microsoft Robotics Studio . Lego Mindstorms NXT hardware kit is used to build hardware of the robot.

Giriş

Bu proje kapsamında geliştirilen robot , kendisinden istenen renkte bir topu otonom olarak arayıp bulmakta ve yine kendisinden istenen renkte bir hedef noktasına taşıyabilmektedir. Bunu yaparken değişik görüntü işleme algoritmaları ve rota hesaplama algoritmaları kullanmaktadır. Geliştirimesi durumunda endüstriyel amaçla kullanılabilecek bu sistem aynı zamanda bir yapay zeka uygulamasıdır.

Introduction

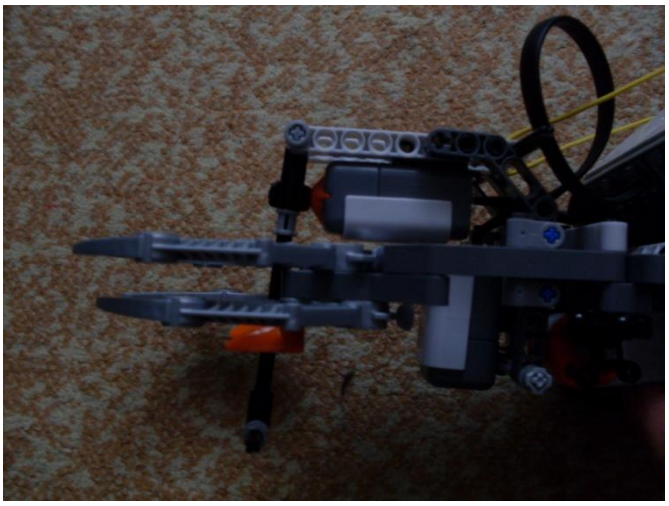
In this project, my robot has ability to search and capture a ball with desired color and transport it to an area with another desired color. It uses various image processing and routing algorithm to do its task accurately. This robot can be used for industrial functions by the time it is modified. It is also a good example of Artifical intelligence.

1. Robotun Donanımsal Yapısı

Robot Ordu'nun donanımsal yapısı Lego Mindstorms NXT Hardware Kit ile gerçeklendi. Üzerinde dört tane sensör (renk sensörü, ses sensörü, ultrasonik sensör, basınç sensörü) bulunmaktadır. Robotun ön kısmına monte edilmiş pençeleri sayesinde bir pinpon topunu rahatça yakalayabilmektedir.

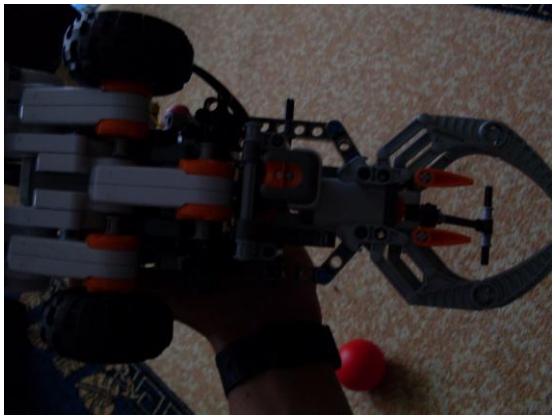
Şekil 1 : Robotun topu yakalama donanımı

Pençelerin hemen ardındaki basınç sensörü, topu tutma işleminin başarısını kontrol etmektedir. Basınç sensörünün geri dönüş değeri 1 ise top tutma işlemi başarılı demektir. Basınç sensörünün hemen altındaki renk sensörü ise zeminin rengini kullanarak istenilen bölgeye gelindiğini kontrol eder.



Şekil 2: Basınç ve ışık sensörleri

Robotun hareket mekanizması sağ ve sol taraftaki birer motora bağlanmış tekerlekler ve arka taraftaki her tarafa dönebilen bir tekerden oluşmaktadır.Sağa gitmek için sol teker, sola gitmek için sağ teker ve ileri gitmek içinse heriki tekerlek çalıştırılmalıdır.



Şekil 3: Motorların bağlanış şekli

Pençe bölümü, robotun alt kısmındaki üçüncü bir motora bağlıdır. Bu sayede robotun topu daha iyi kavraması sağlanmıştır. Robotun üst kısmına bir kamera yerleştirilmiştir. Bilgisayar bu kamerayla görüntüleri alı işlemekte ve robota komut vermektedir.

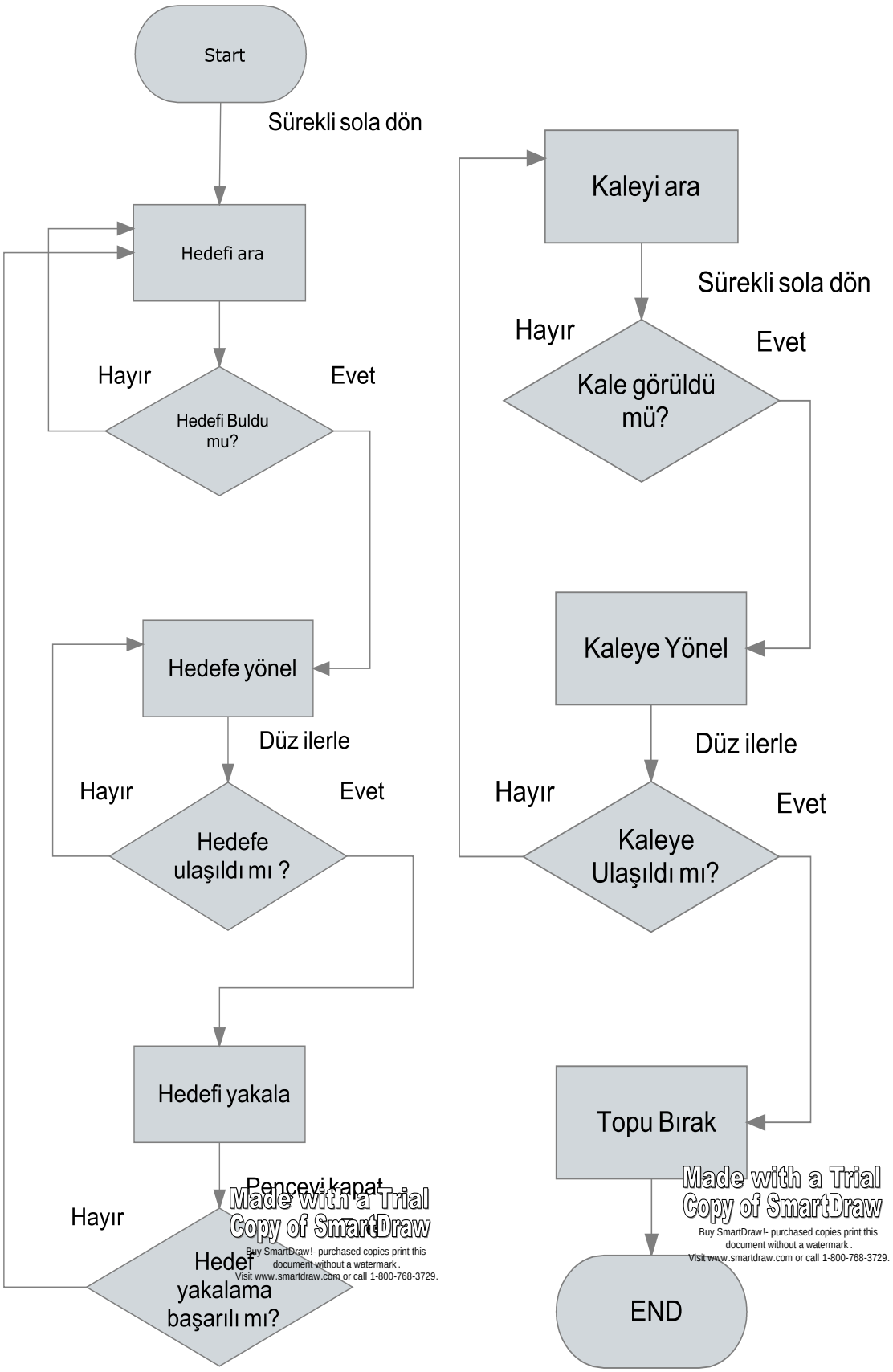
Şekil 4: Robotun genel görünümü

2. Yazılımsal Yapısı

2.1 Aloritma

Robotun yazılımı Visual Studio 2008 ortamında geliştirildi ve visual C# dili kullanıldı. Kameradan alınan görüntü işlenerek hedef noktalar tespit edilerek robotun hareket etmesi sağlandı. Bu uygulama geliştirilirken şu algoritma kullanıldı:

1. Kırmızı topu bulana kadar kendi eksenin etrafında dön
2. Topu yakalamayı yetecek bir mesafeye kadar topa yaklaş
3. Pençeleri kullanarak topu yakala
4. Dokunma sensörü kullanarak tutma işleminin başarısını kontrol et
5. Mavi hedefi görene kadar kendi eksenin etrafında dön
6. Koniye doğru ilerle
7. Yeterince yaklaşınca topu bırak



Şekil 5: Robotun yazılımında kullanılan algoritma

2.2 Kullanılan Sınıflar

Uygulama içinde birden fazla robot kullanılabilmesi için bir Robot sınıfı geliştirilmiştir. Bu sını içinden robota ait tüm donanım kontrol edilebilmektedir. Bu donanımlar : 3 motor, ışık sensörü, ses sensörü, basınç sensörü ve kameradır.

Bu sayede bu sınıfın örnekleri türetilerek uygulama içinde birden fazla robot kullanılarak bir takım oluşturulabilir.

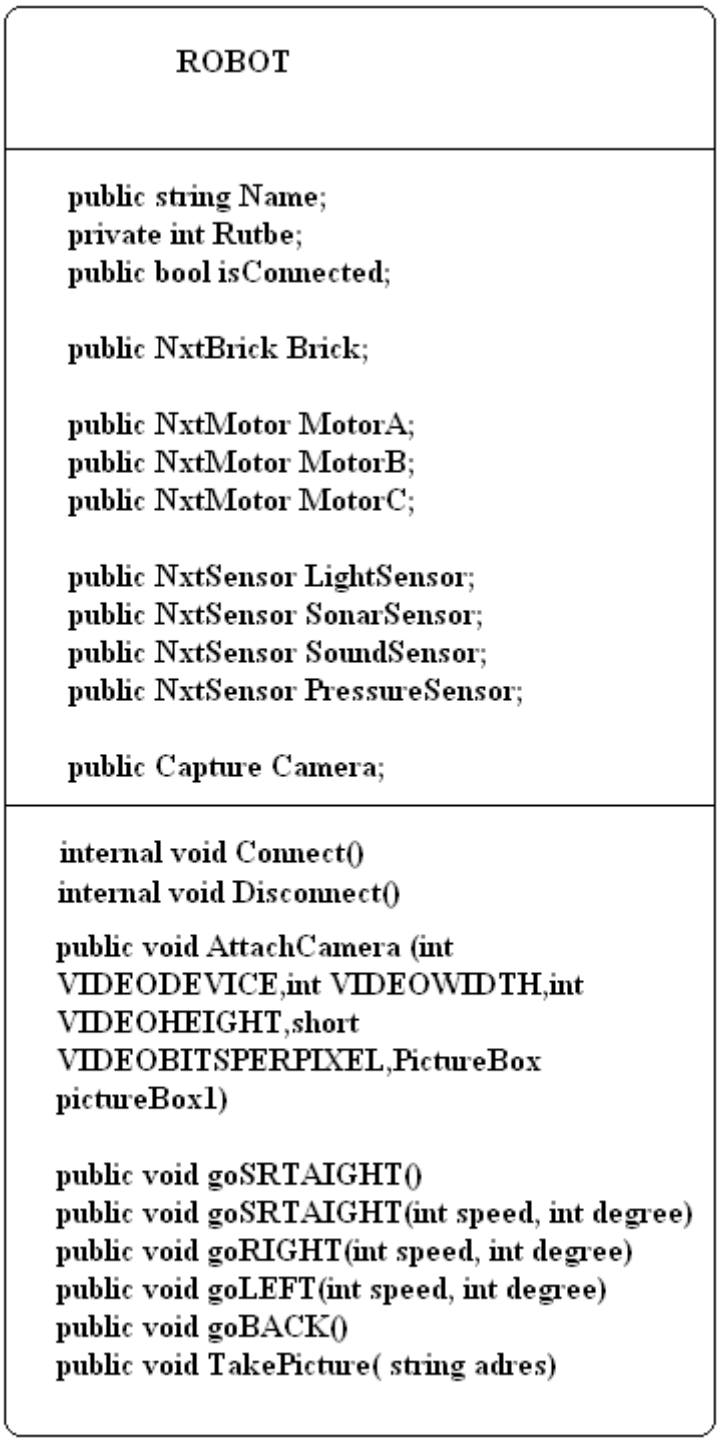
```
class Robot
    {
        public string Name;
        private int Rutbe; // 1, 2, 3
        public bool isConnected; // true = connected, false= unconnected

        public NxtBrick Brick;

        public NxtMotor MotorA;
        public NxtMotor MotorB;
        public NxtMotor MotorC;

        public NxtSensor LightSensor;
        public NxtSensor SonarSensor;
        public NxtSensor SoundSensor;
        public NxtSensor PressureSensor;

        public Capture Camera;
      }
```



Şekil 6 : Uygulama içinde kullanılan Robot sınıfı

2.3 Görüntü İşleme

Görüntü işlenirken kameradaki görüntünün her pikselinin işleme konması gerekir. Fakat normal bir bilgisayarla bu işlem düşük perdormanslı olmaktadır. Bu yüzden görüntü işleme algoritması 0.5 saniyede bir defa çalışacak şekilde düzenlendi. Bu sayede robotun performansı istenen seviyeye yaklaştırılmıştır.

Görüntü işleme sırasında en önemli nokta bulunmak istenen hedefin rengidir. Bu uygulamada aranan topun rengi kırmızı olduğu için kırmızı filtresi uygulanmıştır. Bunun için her pikselin R (red), G(green), B(blue) değerleri alınarak şu algoritma kullanılmalıdır:

Her piksel için;

R=pikselin Red değeri , G= pikselin Green değeri, B= pikselin Blue değeri

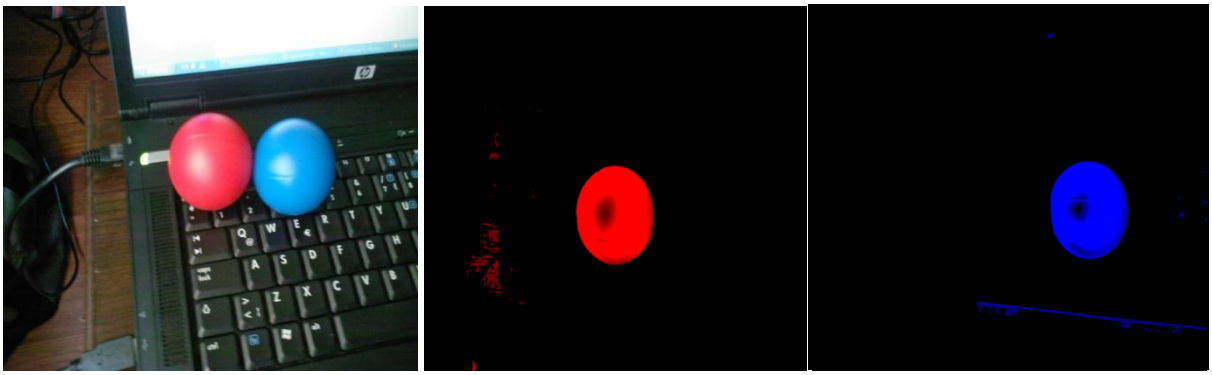
R=(R-G) + (R+G)

G=0, B=0

```
private Image FilterBLUE(Bitmap harita)
        {
            int R, G, B;
            for (int i = 0; i < harita.Height; i++)
            {
                for (int j = 0; j < harita.Width; j++)
                {
                    R = harita.GetPixel(j, i).R;
                    G = harita.GetPixel(j, i).G;
                    B = harita.GetPixel(j, i).B;

                    if ((2 * B - G - R) >= 255)
                    {
                        B = 255;
                    }
                    else if ((2 * B - G - R) >= 100 && (2 * B - G - R) <
255)
                    {
                        B = (2 * B - G - R);
                    }
```

```
                else
                {
                    B = 0;
                }
                harita.SetPixel(j, i, Color.FromArgb(0, 0, B));
            }
        }
        return harita;
    }
```



Şekil 7: Kameradan alınan görüntünün RGB filtresinden geçirilmiş hali

Görüntüyü daha da netleştirmek için bir sınır değeri belirlenerek, bu değerin altındaki pikseller silinebilir. Bu uygulamada sınır değeri olarak 128 kullanılmıştır.

Her piksel için;
Eğer B<sınır_degeri
O halde bu pikseli siyah yap.

Bu noktadan sonra yapılması gereken topun nerede olduğunun belirlenmesi ve topa doğru hareket edilmesidir. Bu nedenle öncelikle topun ağırlık merkezini şu algoritmayla buluyoruz:

İşlenmiş görüntüdeki her nokta için;

$I = (R+G+B)/3$

$COG\_X = COG\_X + (I*x)$

$COG\_Y = COG\_Y + (I*y)$

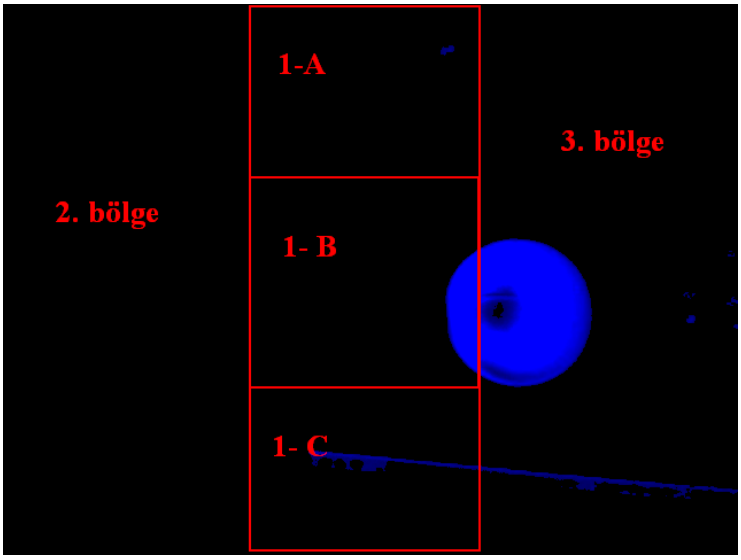
$$Total = Total + I$$

$$COG\_X = COG\_X/Total$$

$$COG\_Y = COG\_Y/Total$$

COG_X ağırlık merkezinin X koordinatını, COG_Y ise Y koordinatını verecektir.

Robotun topa ne kadar yakın olduğunu tespit edebilmek için ekranı parçalara bölerek ağırlık merkezinin hangi bölüme düştüğünü hesaplayıp buna göre hareket etmemiz gerek.



Şekil 8 : Ekranın bölgeler ayrılmış görüntüsü

Tablo 1: Ağırlık merkezinin koordinatlarına göre bölgeler

Bölge	X koordinatı	Y koordinatı	Hareket Yönü
1-A	120<X<200	0<Y<80	Hedef çok uzak
1-B	120<X<200	80<Y<160	Düz ilerle
1-C	120<X<200	160<Y<240	Hedef tutulacak mesafede
2	0<X<120	Y>0	Sola dön
3	200<x<320	Y<0	Sağa dön

Ağırlık merkezinin hangi bölgeye düştüğü belirlendikten sonra robotun hedefe yönelmek için ne yöne doğru hareket edeceği belirlenebilir.

3. Sonuçlar

Sonuç olarak sözkonusu robot yazılımsal ve donanımsal olarak geliştirilmiş ve istenilen görev başarıyla gerçekleştirilmiştir. Gelişim sürecinde bazı zorluklarla karşılaşılmıştır. Bunlar :

- Bluetooth donanımı her zaman iyi bir performans vermemektedir. Ayrıca kapsama alanı çok düşüktür. Robotların daha geniş bir alanda çalışabilmeleri için Wi-fi teknolojileri kullanılabilir.
- Mindstorms donanım setinin büyük bir enerji problemi var. Hızlı enerji tüketimi test aşamasında büyük bir sorun teşkil etmektedir.
- Kameradan alınan görüntünün netliği, robotun çalışmasında en önemli etkendir. Kameranın görüntü kalitesinin düşük olması nedeniyle problem yaşanmıştır.
- Görüntü işleyen bilgisayarın yeterince güçlü olmaması nedeniyle bazen hedefi kaçırabilmektedir. Bu nedenle daha güçlü bir bilgisayar kullanılmalı ve daha pratik algortimalar geliştirilmelidir.

4. Kaynaklar

[1] Oskar Janssen, Wifi interface for mobile robots,University of Twente, February 2008

[2] D. Johnson, T. Stack, R. Fish, D.M. Flickinger, L. Stoller, R. Ricci, J. Lepreau., Mobile Emulab: A Robotic Wireless and Sensor Network Testbed, School of Computing, University of Utah, IEEE Infocom, April 2006

[3] G.Wilfong I.Cox. Autonomous robot vehicles. Springer Verlag, NW, 1990.

[4] S.Thrun. Probabilistic algorithms in robotics. 21:93–109, 2000.

[5] NXT Programming:
http://www.roborealm.com/tutorial/ball_picker/slide010.php

[6]Image Processing in Robotics
http://www.societyofrobots.com/programming_computer_vision_tutorial.shtml

[7] Object Tracking
http://www.endurance-rc.com/tracking.html