

## A Robot's Image Recognition System

Serkan Oncu<sup>1</sup>, Orhan Er<sup>2</sup>

<sup>1</sup>Bozok University, Vocational School, 66200, Yozgat, Turkey

<sup>2</sup>Bozok University, Department of Electrical and Electronics Engineering, 66100, Yozgat, Turkey

---

**Abstract:** In this study, tracing of mobile substances with computerized scanning (image) procedure was carried out in an application correspondingly. In the developed practice, mobile substances will be traced by using scanning (image process) techniques with two cameras. The data from the cameras will be taken as a picture and then direction and speed of the movement will be determined by using scanning process techniques and this is conveyed to the test apparatus as a command by the aid of motors leading the mechanical set up to start. This way scanning process was realized. Thus, the control set up starts the prototype by the aid of commands sent by the software. Owing to this study, detection abilities of humanized robots was developed and a practice which will prepare a back ground for the formation of robots sensitive to movement was developed.

*Keywords:* Mobility of Robot, Image Recognition System.

---

## Bir Robotun Görüntü Tanıma Sistemi

**Özet:** Bu çalışmada bilgisayarlı görü yöntemi metodu ile hareketli cisimlerin takibi, gerçekleştirilen bir uygulama üzerinde çalıştırılmıştır. Geliştirilen uygulamada iki adet kamera sayesinde görüntü işleme teknikleri kullanılarak hareketli cisimler takip edilecektir. Kameralardan alınan bilgiler bir resim olarak alınıp üzerinde görüntü işleme teknikleri kullanılarak hareketin yönü ve hızı tespit edilecektir. Bu tespit akabinde motorlar yardımı ile gerçekleştirilen deney düzeneğine komut olarak gönderilmekte ve mekanik düzeneği harekete geçirmektedir. Bu sayede görüntü takibi işlemi gerçekleştirilmiştir. Bu çalışma sayesinde insanlaştırılan robotların algılama yetenekleri geliştirilerek harekete duyarlı robotların oluşturulmasına zemin hazırlayacak bir uygulama geliştirmiştir.

*Anahtar Kelimeler:* Robot Gezinimi, Görüntü tanıma sistemi.

---

### 1. Giriş

Günümüzde otomatik hareket yeteneğine sahip sistemler teknolojik gelişim sürecinin büyük bir payını içermektedirler. Robot sistemlerinin ilerleyişinde araştırmacılar, daha hızlı, daha dinamik ve daha doğru kararlar verebilen sistemler üretebilmek için, dış dünyaya açılan, insanların sahip oldukları algılayıcılara benzer duyargalar kullanmak ve benzer yollarla algı prensipleri geliştirmek zorundadırlar. Ayrıca bu çalışma şekli, insansı fonksiyonların çalışma hızına yakın olmalı, gerçek zaman içerisinde üretilebilmelidir [1].

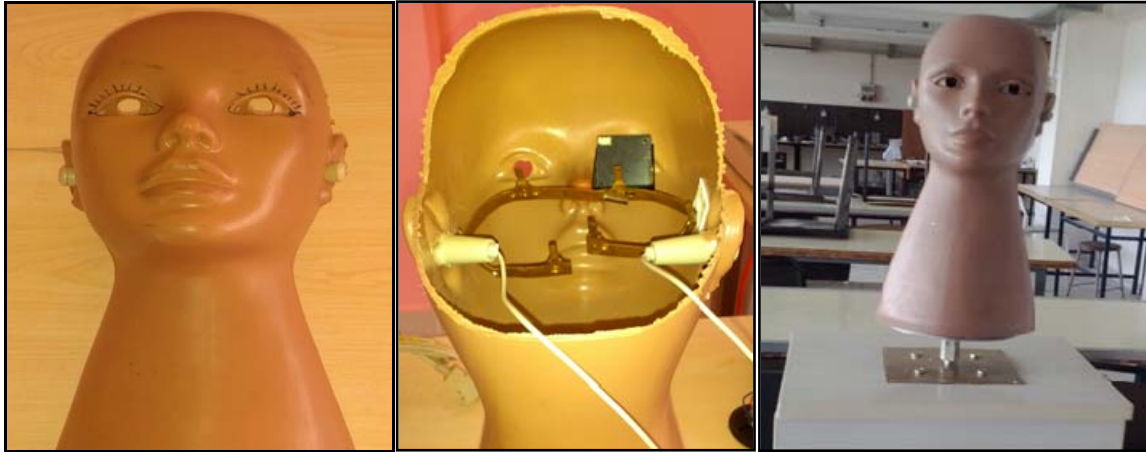
Görüntünün bilgisayar ortamına aktarılabilmesiyle birlikte görüntü işleme cihazlarının hız ve kapasite oranlarında önemli gelişmeler yaşanmıştır. Her ilerleyen yılla birlikte daha yüksek çözünürlüklü ve pikseli görüntüler elde etmeye olanak veren sayısal resim işlemcileri geliştirilmeye başlanmıştır. Yüksek çözünürlük ve piksel oranı beraberinde yüksek veri kapasitesini açığa çıkarmış ve kayıt ortamlarında önemli gelişmeler yaşanmıştır. Pek çok üretici kendi kayıt standardını kabul ettirmeye çalışmış ve bununla birlikte piyasaya çok farklı kayıt ortamları kullanan görüntü işleme cihazları sürülmüştür [2].

---

\* Corresponding author; Tel.: +(90) 354 242 1001, e-mail: orhaner2009@gmail.com

Otomatik hareket yeteneğine sahip sistemler günümüz teknolojik gelişim sürecinin büyük bir payını içermektedirler. Robot sistemlerinin ilerleyişinde araştırmacılar, daha hızlı, daha dinamik ve daha doğru kararlar verebilen sistemler üretebilmek için, dış dünyaya açılan, insanların sahip oldukları algılayıcılara benzer duyargalar kullanmak ve benzer yollarla algı prensipleri geliştirmek zorundadırlar. Ayrıca bu çalışma şekli, insansı fonksiyonların çalışma hızına yakın olmalı, gerçek zaman içerisinde üretilebilmelidir [3].

Bilgisayarla Görü tabanlı Robot sistemleri, araştırmacılar tarafından yoğun olarak çalışılan alanlardan biridir. Özellikle yüksek teknolojiye güvenlik çözümleri, karmaşık algılar gerektiren endüstriyel uygulamalar ve savunma teknolojilerinin gelişimiyle paralellik arz eden bu konu günümüz uygulayıcıları için temel çalışma hedefi haline gelmiştir [4].



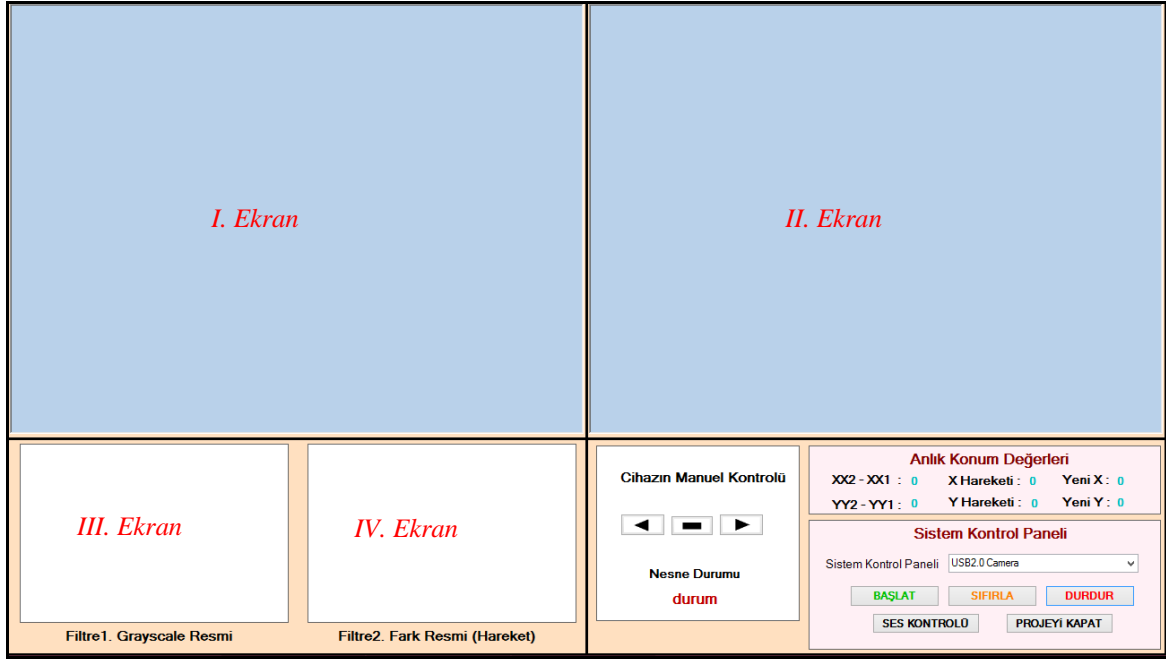
Şekil 1. Hazırlanan robotun baş bölgesi.

Bu çalışmada öncelikle hazırlanmış olan bir prototip (Şekil 1) içerisine 2 adet kamera yerleştirilerek görüntü işlemleri ile hareketli nesnenin takibi amaçlanmıştır. Uygun step açısına sahip olan step motor ile prototipin mekaniksel bağlantısı sağlanmıştır. Step motorun kontrolü için uygun olan sürücü kartı belirlenerek step motor bağlantıları gerçekleştirilmiştir. Bilgisayar ortamında hazırlanan gerekli yazılım ile step motor kontrol sisteminin iletişimi sağlanarak motor kontrolü gerçekleştirilmiştir [5].

## 2. Görüntü Tanıma Sistemi Yazılımı

Geliştirilen bu çalışma ile bir robot prototipinin nesne takibi için iki bölümden oluşan bir yazılım geliştirildi. Bu yazılımın ilk bölümü görüntü işleme teknikleri ile nesnenin hareketini analiz etme ve belirlenen yön doğrultusunda step motor kontrol kartına komut yollayarak robot düzeneğinin hareketini sağlamaktır. Bu amaç doğrultusunda hazırlanan yazılım C# .NET platformunda gerçekleştirilmiş olup genel görüntüsü Şekil 2’te verilmiştir.

Aşağıdaki şekilde görüldüğü üzere görüntü işleme algoritmalarının işletildiği formun birinci ekranında video gerçek zamanlı olarak yüklenmektedir. Bu ekrana yüklenen video bir zamanlayıcı 1 (timer1) yardımıyla belirlenmiş zaman aralıklarıyla “screenshot” komutu kullanılarak 640\*480 boyutunda resimler “ilkframe” adlı değişkene aktarılır. Yüklenen resmin merkez noktası da 320\*240 pikseli olarak belirlenir. Daha sonra zamanlayıcı 1 (timer1) durdurulur ve zamanlayıcı 2 başlatılır. Bu işlemlerin gerçekleştirildiği C# kodları Şekil 3’de verilmiştir.



Şekil 2. C# platformunda geliştirilmiş yazılımın görüntü tanıma arayüzü.

```
public void timer1_Tick(object sender, EventArgs e)
{
    pictureBox4.Visible = true;
    FinalVideoSource = new VideoCaptureDevice(VideoCaptuerDevices
        [comboBox1.SelectedIndex].MonikerString);
    FinalVideoSource.NewFrame += new
        NewFrameEventHandler(FinalVideoSource_NewFrame);
    // Görüntü kalitesi
    FinalVideoSource.DesiredFrameRate = 10;
    // Görüntü büyüklüğü
    FinalVideoSource.DesiredFrameSize = new Size(640, 480);

    FinalVideoSource.Start();

    Size Boyut = new Size(640, 480);
    Bitmap bmpScreenshot = new Bitmap(640, 480);
    System.Drawing.Graphics grafik =
        System.Drawing.Graphics.FromImage(bmpScreenshot);
    grafik.CopyFromScreen(new Point(0, 0), new Point(0, 0), Boyut);
    Bitmap img = (Bitmap)bmpScreenshot;
    if (img != null)
    {
        kameraFrame = img;
        Filtre_Uygula_1(Grayscale.CommonAlgorithms.BT709);
        pictureBox1.Image = ilkFrame;
        pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
        XX1 = 320;
        YY1 = 240;
        timer1.Stop();
        timer2.Start();
    }
    else return;
}
```

Şekil 3. Zamanlayıcı 1 (timer1) ile yapılan işlemler.

Zamanlayıcı 2'nin çalışması ile üzerinden biraz zaman geçmiş ikinci bir görüntü 640\*480 boyutunda ve screenshot komutu ile tekrar resim halinde "kameraframe" adlı değişkene kopyalanır. Daha sonra "kameraframe" ile "ilkframe" adlı resimler "difference" filtresinden geçirilir ve iki resim arasındaki fark resmi elde edilmiş olunur. Bu sayede arka plan çıkartılarak sadece hareketli kısmın piksel değerleri elde edilmiştir. Elde edilen fark resim arka arkaya "Threshold", "Erosion", "Median"

filtrelerinden geçirilerek III. Ekranda gri tonlama resmi ve IV. Ekranda bu üç filtrenin uygulandıktan sonraki fark resminin görüntüsü gösterilir. Bu görüntüler Şekil 4'te verilmiştir.



Şekil 4. Filtreler ile elde edilen gri-ton ve fark resimleri.

Filtre uygulanmış resimler sırasında piksel sayıcı ile hareketin boyutu ölçülere kırmızı kutucuk içerisinde gösterilme işlemi gerçekleştirilerek II. Ekranda gösterilmiştir. Burada kırmızı kutunun merkezi hareketin yoğunluğunun ağırlık merkezi olarak hesaplanmıştır. Bu işlemlerin gösterildiği II. Ekran Şekil 5'te verilmiştir.



Şekil 5. Hareketin boyutunun ve ağırlık merkezinin gösterilmesi.

Oluşan fark resminden elde edilen hareket işleminin ağırlık merkezi bulunduğundan sonra resmin orijinal halinin ağırlık merkezine X ve Y eksenindeki farkı bulunarak hareketin koordinatları hesaplanarak form üzerindeki "Anlık Konum Değerleri" bölümüne yazılmıştır. Bu sayede robot prototipine seri port yardımı ile X ve Y ekseninde nasıl bir hareket verileceği bilgisi gönderilebilmektedir. Bu bilgilerin gösterildiği ekran Şekil 6'de verilmiştir.

| Anlık Konum Değerleri |     |                |
|-----------------------|-----|----------------|
| XX2 - XX1 :           | -25 | X Hareketi : 0 |
| YY2 - YY1 :           | -99 | Y Hareketi : 2 |
|                       |     | Yeni X : 295   |
|                       |     | Yeni Y : 141   |

Şekil 4.28 Hesaplanan hareketin koordinatlarının form üzerinde gösterilmesi.

Bu işlemlerin sonunda zamanlayıcı 2 (timer2) işlemleri sonlandırmış ve hareketin konum bilgileri prototipe gönderilmiş olacaktır. Daha sonra zamanlayıcı 2 (timer2) durdurulup işlemlerin tekrarlanabilmesi için başa dönmek adına zamanlayıcı 1 (timer1) başlatılır. Bu sayede devam eden hareket döngüsel bir şekilde tekrarlanarak sürekli hareketi takip eden bir algoritma haline getirilerek nesne takibi sağlanmıştır.

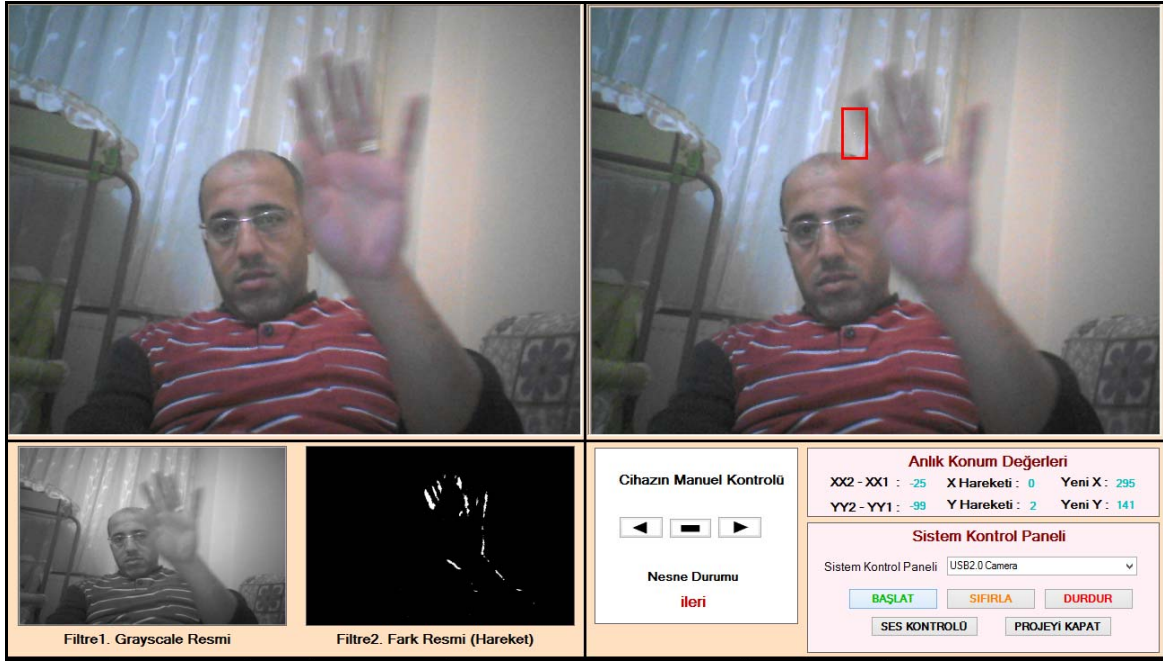
Bu form üzerindeki “Sistem Kontrol Paneli” yardımı ile nesne takibi durdurulup sıfırlanabildiği gibi yeniden başlatılma işlemine de olanak sağlamaktadır. Sistem kontrol paneli Şekil 7’de verilmiştir.

Şekil 7. Sistem kontrol paneli.

Ayrıca görüntü işleme formu üzerine yer alan “Cihazın manuel kontrolü” ile sağa-sola ve durma komutları da kullanıcı tarafından fare yardımı ile gerçekleştirilebilmektedir. Burada amaç görüntünün dışına hızlı hareket etmiş nesnenin takibinin tekrar başlatılabilmesidir. Bu işlemlerin gerçekleştirilebilmesi için tasarlanan menünün görüntüsü Şekil 8’de verilmiştir.

Şekil 8. Cihazın manuel kontrol menüsü.

Yukarıda belirtilen işlemlerin çalışması ile hareketli nesnenin takibinin sağlandığı formun çalışma anındaki görüntüsü Şekil 9’da verilmiştir.



Şekil 9. Nesne takip için görüntü işleme arayüzünün çalıştırılması.

Görüntü işleme sırasında çalıştırılan zamanlayıcı 2 ile gerçekleşen işlemlerin gösterildiği C# kodu Şekil 10'da verilmiştir.

```
private void timer2_Tick(object sender, EventArgs e)
{
    pictureBox4.Visible = true;
    FinalVideoSource = new VideoCaptureDevice(VideoCaptuerDevices
        [comboBox1.SelectedIndex].MonikerString);
    FinalVideoSource.NewFrame += new
        NewFrameEventHandler(FinalVideoSource_NewFrame);
    FinalVideoSource.DesiredFrameRate = 10;
    FinalVideoSource.DesiredFrameSize = new Size(640, 480);

    FinalVideoSource.Start();

    Size Boyut = new Size(640, 480);
    Bitmap bmpScreenshot = new Bitmap(640, 480);
    System.Drawing.Graphics grafik =
        System.Drawing.Graphics.FromImage(bmpScreenshot);
    grafik.CopyFromScreen(new Point(0, 0), new Point(0, 0), Boyut);

    kameraFrame = (Bitmap)bmpScreenshot;
    Filtre_Uygula_2(Grayscale.CommonAlgorithms.BT709);

    FiltersSequence processingFilter = new FiltersSequence();

    processingFilter.Add(new Difference(ilkFrame));
    processingFilter.Add(new Threshold(7 * 7));
    processingFilter.Add(new Erosion());
    processingFilter.Add(new Median());

    Bitmap FiltreUygulanmisFrame = processingFilter.Apply(suankiFrame);
    pictureBox2.Image = FiltreUygulanmisFrame;
    pictureBox2.SizeMode = PictureBoxSizeMode.StretchImage;

    // piksel sayıcı algoritma.
    BlobCounter bc = new BlobCounter();
    // blob counter ayarı
    bc.MinWidth = 5;
    bc.MinHeight = 5;
    bc.FilterBlobs = true;
    bc.ObjectsOrder = ObjectsOrder.Size;
    bc.ProcessImage(FiltreUygulanmisFrame);

    Rectangle[] kareler = bc.GetObjectsRectangles();

    if (kareler.Length > 0)
    {
        Rectangle kare = kareler[0];
        Graphics g = Graphics.FromImage(kameraFrame);
    }
}
```



```

using (Pen pen = new Pen(Color.FromArgb(255, 0, 0), 3))
{
    g.DrawRectangle(pen, kare);
    int X1 = kare.X + (kare.Width / 2);
    int Y1 = kare.Y + (kare.Height / 2);

    XX2 = X1;
    YY2 = Y1;

    kameraFrame.SetPixel(X1, Y1, Color.White);
    pictureBox3.Image = kameraFrame;

    label5.Text = Convert.ToString(X1);
    label6.Text = Convert.ToString(Y1);
    label11.Text = Convert.ToString(XX2 - XX1);
    label2.Text = Convert.ToString(YY2 - YY1);

    int farkX, farkY;
    farkX = Math.Abs((XX2 - XX1) / 128);
    farkY = Math.Abs((YY2 - YY1) / 48);

    label3.Text = Convert.ToString(farkX);
    label4.Text = Convert.ToString(farkY);

    if (XX2 - XX1 > 100)
    {
        if (farkX > 0)
        {
            for (int i = 1; i <= farkX; i++)
            {
                SendToSerialPort("02");
                label7.Text = "ileri";
                Thread.Sleep((XX2 - XX1)*3);
                SendToSerialPort("00");
            }
        }
        else if (XX2 - XX1 < -100)
        {
            if (farkX > 0)
            {
                for (int i = 1; i <= farkX; i++)
                {
                    SendToSerialPort("20");
                    label7.Text = "geri";
                    Thread.Sleep(-1 * (XX2 - XX1)*3);
                    SendToSerialPort("00");
                }
            }
            else SendToSerialPort("00");
        }
    }
    timer2.Enabled = false;
    timer2.Stop();
    timer1.Enabled = true;
    timer1.Start();
}

```

Şekil 10. Zamanlayıcı 2 (timer2) ile yapılan işlemler.

### 3. Sonuç ve Öneriler

Bu çalışmada, oluşturulan bir prototip ile C# programında oluşturulan yazılımda görüntü işleme teknikleri kullanılarak hareketli nesne takibi gerçekleştirilmiştir. Tasarlanan sistemde yapılan işlemler 2 ana kısımda gerçekleştirilmiştir. Bunlar; prototipin mekanik hareketini sağlayan aksamı oluşturma ve prototipin hareketli nesneyi takip etmesi için gerekli olan görüntü işleme tekniklerini kullanılarak elde edilen yazılımın oluşturulmasıdır. Bir cansız manken kafa kısmının gözlerine mikro kamera yerleştirilerek hareketli nesne takibini gerçekleştirecek olan prototip oluşturulmuştur. Bu prototipin göz bölümlerinden sayısal görüntü elde edilmiştir. Oluşturulan prototipin mekanik hareketini sağlamak için bir uygun özellikte bir step motor ve step motor sürücüsünün kontrolü için kontrol kartı tasarlanarak bunların bir kabin içerisine yerleştirilip bağlantıları gerçekleştirilmiştir.

Yapılan deneysel çalışma ile görüntü işleme ile hareketli nesnenin takibinin yapılabileceği ve bunların bir motor sistemi sayesinde çalıştırılabileceği görülmüştür. Bu doğrultuda oluşturulan yazılımlar ve prototipin birbirileri ile uyumlu bir şekilde çalıştığı belirlenmiştir. Prototip üzerindeki görüntü bilgilerinin alındığı kameraların daha hassas ve kaliteli hale getirilerek daha hassas ve daha başarılı nesne takibi gerçekleştireceği öngörülmektedir.

Bu çalışmanın sonucunda elde edilen veriler ışığında çalışmanın daha da geliştirilerek özellikle savunma sanayi ve endüstriyel sanayide birçok çalışma için örnek teşkil edeceği düşünülmektedir.

## **References**

- [1] ER, O. "İnsan Yüzü Bulma ve Tanıma", Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı Yüksek Lisans Tezi, Sakarya, 2004.
- [2] İKMEB Elektrik-elektronik programı dokümanları, [www.ikmeh.gov.tr](http://www.ikmeh.gov.tr). (son erişim 01.12.2015).
- [3] Koyuncu, E, Ceylan, O, Yeniçeri, R, "Bilgisayarla Görü Tabanlı, Hareketli Cisim Yörüngesi İzleyen Robot Kol Tasarımı", Otomatik Kontrol Türk Milli Komitesi, Otomatik Kontrol Ulusal Toplantısı (TOK'05) Bildiri Kitabı, sayfa 291-296, İstanbul, 2-3 Haziran 2005.
- [4] Serkan ÖNCÜ, "Moving subject pursuit by dint of computer controlled flying camera", Master Thesis, Science Institute, Bozok University, 2014.