

1. GİRİŞ

Görüntü işleme, bilgisayarlı görü ve gerçek zamanlı hareket algılama tanıma sistemleri geçtiğimiz yıllarda bilimsel araştırma grupları ve ticari yazılım sektöründe oldukça geniş ilgi görmüştür. Halen bu sistemlerin kurularak gerçek zamanlı uygulamaların geliştirilmesi süreci, yüksek donanım ve yazılım maliyetlerinden ayrıca yetişmiş insan gücünün yetersizliğinden dolayı istenen konuma gelememiştir.

Görmeye dayalı algılama yapılarında, bir takım fiziksel sınırların aşılabilmesiyle bilgisayarlarla görme, üstün bir algılama sistemi olarak uygulamalarda boy göstermeye başlamıştır. Bu türden gelişimler bir sistem olarak düşünüldüğünde, hata yapma olasılığı yüksek ve yavaş kalan insan denetimli yapılardan yapay zeka yapılarına doğru geçişi hızlandırmıştır. İnsan denetiminin üstün özellikleri olan tecrübe, öğrenme, karar verebilme gibi yetenekler , yapay olarak en azından çalışılacak alan için tasarlanabildiği takdirde ortaya hedef iş için yüksek başarımlı gösteren akıllı yapılar çıkacaktır. Bu durum bazı beklentilerin aksine, insan etkeninin dışlanması olarak değil, akıllı sistemlerin kurulmasında uzman görüş ve uzman kural tabanları kurulması gibi üstün görevler için, daha üst düzeyde kullanılması olarak görülmelidir. Böylelikle çalışan sistemlerde; karmaşık algılamalar ve duygusal çıkarımlar içermeyen, tekrar eden, yorulma etkeninin yok edilmek istendiği görevlerde kurulabilecektir.



Bu çalışmada önce gerçek zamanlı görüntü laptop kamerası kullanılarak

alınıp bilgisayar ortamına aktarılmış ve Matlab programında görüntü işleme çalışmalarından sonra belirli bir renkteki cisimler algılanmıştır. Algılanan cisimlerin koordinatları bulunmuştur.

Projenin Matlab Kodu:

- **a = imaqhwinfo;**

Mevcut bulunan kameradan hakkındaki teknik bilgilerin alınma işleminin yapıldığı kısımdır.

- **[camera_name, camera_id, format] = getCameraInfo(a);**

Kamera bilgilerinin alındığı kısımdır.

- **set(vid, 'FramesPerTrigger', Inf);**

Bu fonksiyonu ile video giriş nesnesinin her bir tetikleyiciyi çalıştırdığı zaman elde ettiği frame sayısını belirtilmiştir. Varsayılan olarak Matlab'da 10 olarak ayarlanmıştır. Bu komut ile video nesnesi bir hata oluşana ya da bir durdurma komutu verilene kadar frameler hafızada tutulmuştur.

- **set(vid, 'ReturnedColorspace', 'rgb') ;**

Video RGB uzayına çevirilmiştir.. Bu proje kapsamında kırmızı renk nesne takibi yapılacağından RGB renk uzayı ayarlanmıştır. RGB uzayında belirlediğimizde: kırmızı yeşil ve mavi renge göre işlem yapabiliyoruz. Bunun dışında; YCbCr Y ile luminance (parlaklık) sinyalini, Cb ve Cr ile ise chrominance (renk) bilgilerini saklayan bir renk uzayıdır.

- **vid.FrameGrabInterval = 3;**

Bu fonksiyon ile bir video giriş nesnesinin üretilen bir videodan nasıl frame yakalayacağını belirtmektedir. Varsayılan olarak bu değer birdir. Programımızda kullandığımız komut ile her üç frame elde edilmiş ve depolanmıştır.

- **start(vid) ;**

Kameradan görüntü alma işlemi başlatılmıştır.

- **while(vid.FramesAcquired<=200)**

Belli bir frame aralığında çalışma yapabilmemiz için döngü kurulmuştur.

- **data = getsnapshot(vid);**

Frame yakalayıcıdan gerekli frame'i elde etmek için kullanılmıştır. Programımızda data = getsnapshot(video) komutuyla kameradan sürekli olarak frame yakalanmıştır.

- **diff_im = imsubtract(data(:,:,1), rgb2gray(data));**
diff_im = medfilt2(diff_im, [3 3]);
diff_im = im2bw(diff_im,0.18);

RGB uzayına çevirdiğimiz video'muzdan referans rengimiz olan kırmızı renkleri ayıklayabilmemiz için belli adımlar gereklidir. 3 renk katmanından (RGB) oluşan resmin kırmızı katmanı ayıklanmıştır ve gri katmana göre farkına bakılmıştır. Resmin kırmızı katmanı data değişkeninin 1. boyutunda tutulmaktadır ve erişmek için resim (:,:,1) kod satırı, resmin gri renkli hali için ise rgb2gray(data) fonksiyonu kullanılmıştır. Daha sonra bulunan kırmızı ve gri katmanlar birbirinden imsubtract() fonksiyonu ile çıkartılıp, fark elde edilmiştir. Medyan filtresi ile gürültüler yok edildikten sonra belirlenen eşik değerine göre kırmızı cisimlerin beyaz, arkaplanın ise siyah olması sağlanarak resim içindeki renkler ayırt edilip, resim daha kolay çalışabilir hale getirilmiştir.

- **diff_im = bwareaopen(diff_im,300);**

Bwareaopen fonksiyonunda 300 değeri belirlenerek 300'den küçük olan nesneler görüntüden çıkarılmıştır. Bu işlem ders kapsamında da görmüş olduğumuz aşındırma (erosion) işleminin benzeridir. **Aşındırma:**Yapı elemanı altındaki tüm pikseller ön plan değilse, test edilen piksel arka plan yapılıp. Bağlantılı bileşenler aşınır. Öznelikler küçülür. Köprü, dal yapıları ve küçük boyutlu gürültüler kaldırılır .

- **bw = bwlabel(diff_im, 8);**

Bölgenin belirlenmesi için kırmızı renk özelliği taşıyan toplu piksel bloklarının seçilmesi gerekir. Bu amaçla Matlab'ın bwlabel fonksiyonun faydalanılmıştır. Bwlabel özelliği sayesinde birbirine yakın duran piksel grupları bütün bir öbek şeklinde alınmıştır. Programımızda bwlabel(diff_im,8); komutu ile 8 komşuluğuna bağlı bileşenler bulunmuştur.

- **stats = regionprops(bw, 'BoundingBox', 'Centroid');**

regionprops() komutuyla bulunan nesnelerin merkezleri ve alanları, çevrelemek için kullanılan boundingbox() fonksiyonuyla da alan,merkez değerleri gibi birçok özelliği alınarak stats değişkenine atanmıştır. Regionprops ile alınan etiket matrisi geri döndürülmesi gereken etiket tanımlarını içeren bir dizi string içerebilir. Boundingbox yöntemi ile tüm bölge kaplanıp işaretlenir. İşaretli bölgede varsayımsal ölçekte

kırmızı nesne niteliği taşıyacak büyüklükte piksel grubu aranır eğer bulunursa İşaretlenir. Bu durum tüm aday bölgeler için tek tek çalıştırılır. Eğer hiçbir aday bölge için kırmızı renk şartı sağlanmıyorsa o çerçevede ilgili nesne yok demektir.

- **for object = 1:length(stats)**
 bb = stats(object).BoundingBox;
 bc = stats(object).Centroid;
 rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
 plot(bc(1),bc(2), '-c+')
 a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y:',
num2str(round(bc(2)))));
 set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');
 end

Burada for döngüsü kullanılarak bulunan kırmızı nesneler dikdörtgen kutu içerisinde gösterilmiştir. Gösterilecek dikdörtgen çerçevenin rengi, cisimlerin X ve Y koordinatları, hangi yazı tipi'nde, hangi renkte gösterileceği gibi detay ayarlamaları yapılmıştır. Görsellik isteğe bağlı olarak buradan değiştirilebilir.

stop(vid);
flushdata(vid);

Kamera durdurulur. Kalan bilgiler temizlenir.