21.04.2024

Görüntü İşleme

Bilardo oyunu top tespiti ve takibi

top, yeşil, havuz, bilardo, dinlenme odası içeren bir resim

Açıklama otomatik olarak oluşturuldu

M Fatih Sarıbıyık

B210109013

Özet

Bilardo topu tespiti ve takibi projesi Python ve OpenCV kütüphanelerini kullanarak topun konumunu ve hareketini tespit etmektedir.

Topun Tespiti :video dosyasını frame frame fotoğraflara ayırır ve bunu bir maskeleme işleminde geçirir. Maskeleme işlemi videodaki kırmızı rengi ve tonlarını 1 kabul eder. Kalan bütün renkleri 0 olarak kabul eder.çıktı olarak elimize siyah beyaz bir görüntü oluşmaktadır. Kırmızının olduğu yer beyaz , kalan alanlar siyah olarak gözükür.

Hareketin Tespiti: Hareket tespiti için bir önceki konum ile şuanki konum arasındaki farka bakılır.Aradaki piksel farkı zamana bölünür ve piksel/saniye cinsinden bir hız değeri verilir.

Video çıktısı: Çıktıdaki Hız pixel /saniye cinsinden verilmiştir ve her framede hesaplanır. Çıktı ekranında Zamanın verilme sebebi ise hızın tespiti içindir. Her framede bulunan fark geçen saniyeye bölünerek hız bulunur.

Fotoğraf çıktısı: Çıktıda zaman , hız ,topun hareketi , başlangıç ve bitiş noktaları , bu aradaki mesafe gösterilmektedir.

Bölüm 1

Topun Yerinin Tespiti:

İlk olarak kırmızı renkli bir maske oluşturulur.Bu işlem numpy kütüphanesinin array fonksiyonunu kullanarak yapılır. HSV(ton,doygunluk,değer) şeklinde bir aralık seçilerek yapılır. Bu aralık Lower\_red ve Upper\_red arasında seçilir. Video görüntüsünde karıncalanma olduğu için ton aralığı büyük seçilmiştir.

Sonrasında Kontur tespiti yapılır.Bu işlem openCV findConturs fonksiyonu ile yapılır. Bu işlem kırmızı rengin dış hatlarını tespit eder.

Sonrasında topu belirtmek için (kırmızı büyük halka) bir sınırlayıcı çizilir. bu işlem için OpenCV boundingRect fonksiyonu kullanılmıştır. Bu Sınırlayıcı Daire biçimindedir ve merkez noktası topun merkezi kabul edilir.

Sonrasında Hız hesabı hesaplamak için bir önceki ve şuanki topun merkesi arasındaki mesafe hesaplanır . Onceki konum (prevCX,prevCY) şuanki konum (CX,CY) arasındaki vesafe iki nokta arasındaki dik uzaklık yani Öklid mesafesi hesabı yapılarak bulunur. Bu işlemi detect\_red\_ball\_motion fonksiyonunun içindenumpy kutuphanesini kullanrak yapmış olduğum x’ler farkı ve y’ler methodunu kullanarak yaptim. Onceki ve şuanki x in kordinatlarını çıkarıp karelerini alıp aynı işlemi y içinde yaptıktan sonra çıkan değerleri toplarsak ardından kareköke alırsak çıkan sonuç bize aradaki dik uzaklığı verecektir. Sonucumuzu bir değişkende tutup (motion\_distance) zamana bölerek anlık hızı bulmuş oluyorum. Videodaki karıncalanmadan dolayı çok küçük titreşimlerde bir eşik değer koydum ve topun hareketinin çok küçük olduğu (titreme anları) hız eşik değerden küçük olduğu için 0 olarak gösteriliyor ve daha iyi bir hız tespiti yapmış oluyoruz.

Sonuçlar:Yapitğim işlerim sonuçlarını görmek için önce ekrana bastıyorum . Çok hızlı olduğu için terminalede yazdırıyorum.Son frame’yi bulamk için bir değişkende çıktıların kopyalarını tutuyorum ve sonucusunu final\_output\_frame.jpg dosyası olarak kaydettim.

Bölüm 2

Çıktı Dosyaları

top, yeşil, havuz, bilardo, dinlenme odası içeren bir resim

Açıklama otomatik olarak oluşturulduFinal\_Output\_Frame

Projeye Ek olarak Başlangıç ve Bitiş noktası tespiti yaptim.Topun başlangıcından Bitişe kadara en kısa yolu buluyor. Bunun için başlangıç noktası olan start\_point yani ilk büyük kırmızı nokta , end\_point yani son büyük kırmızı nokta arasındaki uzaklığı tespit eder.Bu işlemi her frame için değil ikaydediğim ilk nokta ve son framedeki son büyük noktayı iki nokta arasındaki uzaklığı hesaplayan fonksiyona atarak yapıyorum. Çıktı:

renklilik, çizgi, yeşil, sanat içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu işlemi her framede uygularsak topun hareketi esnasında gittiği yerleri görebiliriz.Yeşil noktalar her frame için uyglanarak hesaplanmıştır.

yeşil, çocukların yaptığı resimler, çizgi, sanat içeren bir resim

Açıklama otomatik olarak oluşturuldu

Alttaki İki büyük kırmızı çerçevede topun ilk harekete başlangıcını ve hareket esnasında gittiği yerleri gösterir.

Hareket başlangıç noktası:

renklilik, lamba içeren bir resim

Açıklama otomatik olarak oluşturuldu

Hareket bitiş noktası:

renklilik, ekran görüntüsü, sanat içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ek olarak konsol ekranına daha iyi anlaşılabilmesi için (ekranda hız çok hızlı değiştiği için) terminal ekranına da hız ve iki büyük kırmızı nokta (başlangıç ve bitiş arasındaki mesafe) değerlerini yazdırmayı düşündük. Bu kodların doğru çalışıp çalışmadığını takip etme ve hız değerlerinin ne olduklarını görme açısından bize daha iyi bir olanak sağlamaktadır.