ABSTRACT

Bu projede, insan-bilgisayar etkileşimini mümkün kılan gerçek zamanlı el jesti tanıma yaklaşımı sunulmaktadır. Sadece bir web kamerası ve Bilgisayarlı Görü (BG) teknolojisi gibi görüntü işleme gibi "Görüntü Tabanlı" bir yaklaşımı kullanmaktadır. Bu teknoloji sayesinde birçok el jestini tanıyabilen sistem, gerçek zamanlı el jesti tanımanın birçok uygulamasına sahiptir. Bu uygulamalar, temel uygulamalardan başlayarak alan özelindeki özelleşmiş uygulamalara kadar, bilgisayarlarla etkileşimde bulunduğumuz hemen hemen her yerde kullanılabilir. Şu anki seviyesi toplum için yararlı olsa da, endüstriyel düzeyde kullanılmak üzere daha da genişletilebilir. Jest tanıma, bilgisayarlı görüntüdeki aktif araştırma alanlarından biridir. Mevcut sistemler genellikle belirli bir işaretleyici türü ile el tespiti kullanırken, bizim sistemimiz gerçek zamanlı el görüntüsü tanıma sistemini kullanmaktadır.

INTRODUCTION

Bu çalışmada, bir kameranın renk tespit tekniğine dayanarak fare imleci hareketini ve tıklama olaylarını kontrol etmeye çalıştık. Gerçek zamanlı video, bir Web Kamerası kullanılarak yakalandı. Kullanıcı, sisteme bilgi sağlamak için renkli bantlar kullanmaktadır. Videonun bireysel kareleri ayrı ayrı işlenir. İşleme teknikleri, renkleri algılamak için bir görüntü çıkarma algoritması içerir. Renkler algılandıktan sonra, sistem imleci takip etmek ve çeşitli işlemler yapmak için çeşitli işlemler gerçekleştirir.

Günümüz bilgisayar çağında, her birey günlük görevlerinin çoğunu bilgisayarlarla gerçekleştirmektedir. Bir bilgisayarı kullanırken kullandığımız başlıca giriş cihazları klavye ve faredir. Ancak birçok insanı etkileyen geniş bir sağlık sorunu vardır: sürekli ve sürekli bilgisayarla çalışmanın neden olduğu sağlık sorunları. Ellerin doğrudan bir giriş cihazı olarak kullanılması, doğal İnsan Bilgisayar Etkileşimi sunmak için çekici bir yöntemdir. Bu tür, metin tabanlı arayüzlerden 2D grafik tabanlı arayüzlere, multimedya destekli arayüzlere ve tam teşekküllü çoklu katılımcı Sanal Ortam (SO) sistemlerine kadar evrim geçirmiştir. El jestleri tamamen doğal bir iletişim formu olduğundan, klavye ve fareyi aşırı kullanımının olumsuz etkisi olmaz. Gelecekteki insan-bilgisayar etkileşimini düşünün: Bir nesneyi hareket ettirmek ve döndürmek için elinizi hareket ettirerek ve döndürerek yapabileceğiniz 3D bir uygulama - hiçbir giriş cihazına dokunmadan. Bu makalede, görüntü tabanlı el jesti tanıma üzerine bir inceleme sunulmaktadır.

HAND GESTURE RECOGNITION'IN GEREKLİLİĞİ

Genellikle, El Jest Tanıma teknolojisi, "Veri Eldivenleri" veya "Renkli İğneler" kullanılarak uygulanır, bu da ek maliyet ve çoğunluk tarafından kullanılamama sorununa yol açar. Ayrıca, ek cihazlar kullanmak, daha fazla bakım gerektirir. Web kamerası kolayca temin edilebilen bir cihazdır ve günümüzde her dizüstü bilgisayarın birleşik bir web kamerası bulunmaktadır. Projemizde, bir web kamerası karesinde hareket eden bir eli ve jestini algılayabilir

**MATLAB IMPLEMENTATION**

function jestTanima(kirmiziEsik, yesilEsik, maviEsik, numFrame)

warning('off','vision:transition:usesOldCoordinates');

if nargin < 1

kirmiziEsik = 0.22;

yesilEsik = 0.14;

maviEsik = 0.18;

numFrame = 1000;

end

ekranBoyutu = get(0,'ScreenSize'); % Ekran boyutunu tanımla

% Video ayarları

[vidCihaz, vidBilgi] = setupVideoDevice();

% Diğer ayarlar ve nesneler

robot = java.awt.Robot;

blobAnaliz = setupBlobAnalysis();

sekilEkleme = setupShapeInserter();

videoOynatici = setupVideoPlayer(vidBilgi);

% İşlemler

frameSayisi = 0;

lSayisi = 0; rSayisi = 0; dSayisi = 0;

iPozisyon = vidBilgi.MaxWidth/2;

while (frameSayisi < numFrame)

rgbKare = step(vidCihaz);

rgbKare = flipdim(rgbKare,2);

[merkezKirmizi, kutuKirmizi] = processColor(rgbKare(:,:,1), kirmiziEsik, blobAnaliz);

[merkezYesil, kutuYesil] = processColor(rgbKare(:,:,2), yesilEsik, blobAnaliz);

[~, kutuMavi] = processColor(rgbKare(:,:,3), maviEsik, blobAnaliz);

performActions(robot, merkezKirmizi, kutuKirmizi, merkezYesil, kutuYesil, kutuMavi, iPozisyon, vidBilgi, ekranBoyutu);

vidGiris = step(sekilEkleme, rgbKare, kutuKirmizi,single([1 0 0]));

vidGiris = step(sekilEkleme, vidGiris, kutuYesil,single([0 1 0]));

vidGiris = step(sekilEkleme, vidGiris, kutuMavi,single([0 0 1]));

step(videoOynatici, vidGiris);

frameSayisi = frameSayisi + 1;

end

release(videoOynatici);

release(vidCihaz);

clc;

end

function [vidCihaz, vidBilgi] = setupVideoDevice()

kamera = imaqhwinfo;

kameraAdi = char(kamera.InstalledAdaptors(end));

kameraBilgi = imaqhwinfo(kameraAdi);

kameraId = kameraBilgi.DeviceInfo.DeviceID(end);

kameraFormat = char(kameraBilgi.DeviceInfo.SupportedFormats(end));

vidCihaz = imaq.VideoDevice(kameraAdi, kameraId, kameraFormat, 'ReturnedColorSpace', 'RGB');

vidBilgi = imaqhwinfo(vidCihaz);

end

function blobAnaliz = setupBlobAnalysis()

blobAnaliz = vision.BlobAnalysis('AreaOutputPort', false, 'CentroidOutputPort', true, ...

'BoundingBoxOutputPort', true, 'MaximumBlobArea', 3000, ...

'MinimumBlobArea', 100, 'MaximumCount', 3);

end

function sekilEkleme = setupShapeInserter()

sekilEkleme = vision.ShapeInserter('BorderColorSource', 'Input port', 'Fill', true, ...

'FillColorSource', 'Input port', 'Opacity', 0.4);

end

function videoOynatici = setupVideoPlayer(vidBilgi)

videoOynatici = vision.VideoPlayer('Name', 'Son Video', ...

'Position', [100 100 vidBilgi.MaxWidth+20 vidBilgi.MaxHeight+30]);

end

function [merkez, kutu] = processColor(kanal, esik, blobAnaliz)

if size(kanal, 3) == 3 % Eğer gelen görüntü RGB ise

kanal = im2gray(kanal); % RGB'yi grileştir

end

diffKanal = imsubtract(kanal, esik); % Gerekli işlemleri yap

binKanal = im2bw(diffKanal, esik);

[merkez, kutu] = step(blobAnaliz, binKanal);

end

function performActions(robot, merkezKirmizi, kutuKirmizi, merkezYesil, kutuYesil, kutuMavi, iPozisyon)

lSayisi = 0; rSayisi = 0; dSayisi = 0;

eminOlay = 5;

if length(kutuKirmizi(:,1)) == 1

robot.mouseMove(1.5\*merkezKirmizi(:,1)\*ekranBoyutu(3)/vidBilgi.MaxWidth, 1.5\*merkezKirmizi(:,2)\*ekranBoyutu(4)/vidBilgi.MaxHeight);

end

if ~isempty(kutuMavi(:,1))

if length(kutuMavi(:,1)) == 1

lSayisi = lSayisi + 1;

if lSayisi == eminOlay

robot.mousePress(16);

pause(0.1);

robot.mouseRelease(16);

end

elseif length(kutuMavi(:,1)) == 2

rSayisi = rSayisi + 1;

if rSayisi == eminOlay

robot.mousePress(4);

pause(0.1);

robot.mouseRelease(4);

end

elseif length(kutuMavi(:,1)) == 3

dSayisi = dSayisi + 1;

if dSayisi == eminOlay

robot.mousePress(16);

pause(0.1);

robot.mouseRelease(16);

pause(0.2);

robot.mousePress(16);

pause(0.1);

robot.mouseRelease(16);

end

end

else

lSayisi = 0; rSayisi = 0; dSayisi = 0;

end

if ~isempty(kutuYesil(:,1))

if (mean(merkezYesil(:,2)) - iPozisyon) < -2

robot.mouseWheel(-1);

elseif (mean(merkezYesil(:,2)) - iPozisyon) > 2

robot.mouseWheel(1);

end

iPozisyon = mean(merkezYesil(:,2));

end

end

**PROBLEMS AND DRAWBACKS**

Sistem, bir web kamerası aracılığıyla görüntü yakalamaya dayandığından, belirli bir ölçüde aydınlatmaya bağımlıdır. Ayrıca, arka planda diğer renkli nesnelerin bulunması, sistemi yanlış bir yanıt vermeye zorlayabilir. Sistem, bu sorunu eşik değerlerini ve diğer parametreleri yapılandırarak azaltabilir, ancak yine de işletme arka planının aydınlık olması ve parlak renkli nesnelerin bulunmaması önerilir. Düşük hesaplama yeteneklerine sahip belirli bilgisayarlarda sistem daha yavaş çalışabilir, çünkü çok sayıda karmaşık hesaplama çok kısa bir sürede gerçekleştirilmelidir. Ancak standart bir bilgisayar veya dizüstü bilgisayar, sistemin optimum performansı için gereken hesaplama gücüne sahiptir. Kameranın çözünürlüğü çok yüksekse sistem yavaş çalışabilir. Ancak bu sorun, görüntünün çözünürlüğünü düşürerek sistemde değişiklikler yaparak çözülebilir.

**CONCLUSION**

Bu çalışmada, bir web kamerası kullanılarak uygulanan bir nesne takibi tabanlı sanal fare uygulaması geliştirilmiştir. Sistem, MATLAB Görüntü İşleme Aracı Kutusu kullanılarak MATLAB ortamında uygulanmıştır. Bu teknolojinin artırılmış gerçeklik, bilgisayar grafikleri, bilgisayar oyunları, protezler ve biyomedikal enstrümantasyon gibi geniş uygulama alanları bulunmaktadır. Ayrıca, sanatçılar arasında popülerlik kazanan dijital bir tuval gibi uygulamalar oluşturmak için benzer bir teknoloji uygulanabilir. Bu teknoloji, uzuvlarını kontrol edemeyen hastalara yardımcı olmak için kullanılabilir. Bilgisayar grafikleri ve oyun alanında, bu teknoloji bir kişinin hareketlerini izleyerek ve komutlar olarak yorumlayarak etkileşimli oyunlar oluşturmak için modern oyun konsollarında kullanılmıştır. Çoğu uygulama genellikle çok pahalı olan ek donanım gerektirir. Amacımız, bu teknolojiyi mümkün olan en ucuz şekilde oluşturmak ve aynı zamanda standart bir işletim sistemi altında oluşturmaktı. Bu teknoloji için özel olarak çeşitli uygulama programları yazılabilir ve kaynakların minimum gereksinimleri ile geniş bir uygulama yelpazesi oluşturulabilir