

# Görüntü işleme ile insan el parmaklarının sayısının gerçek zamanlı olarak tahmin edilmesi

Ahmet YAYLALIOĞLU<sup>1</sup>

*Balıkesir Üniversitesi Mühendislik-Mimarlık Fakültesi, Çağış Kampüsü, Balıkesir*

## Özet

*Günümüzde yapay zekalı sistemler yaygın bir şekilde kullanıldığından, bu sistemlere basit sensör verileri yeterli gelmeyebiliyor. Bu yüzden birçok sistem tasarlanırken kameralardan görüntü alması, bu görüntüleri işleyerek bir karar mekanizması oluşturması hedefleniyor. İnsan ve robot etkileşimi de gün geçtikçe arttığından, robotların/sistemlerin insan hareketlerini anlamlandırması bekleniyor. Bu sebeple geliştirilen projede insan elinin hareketleri anlamlandırılmaya çalışılarak, gerçek zamanlı el parmağı sayısı tahmin ediliyor. Bunu yaparken arkaplanda sadece düz bir renk olmak zorunda değildir. Arkaplan gürültüsünü mümkün olabildiğince gidermeye çalışan bu yazılım, sadece sinyal işleme yöntemleri ile oluşturulmuştur ve yüksek verimlilik ile çalışmaktadır.*

**Anahtar kelimeler:** *Görüntü işleme, el algılama, el hareketi tanımlama, parmak sayısı*

---

<sup>1</sup> Ahmet YAYLALIOĞLU, [ahmetyaylalioglu@gmail.com](mailto:ahmetyaylalioglu@gmail.com), Tel: (534) 618 86 51

# Estimation of the number of real-time hand fingers with image processing

## Abstract

*Since artificial intelligence systems are widely used today, simple sensor data may not be enough for these systems. Therefore, when many systems are designed, it is aimed to take images from the cameras and to construct a decision mechanism by processing these images. Since human and robot interaction is increasing day by day, it is expected that robots / systems will make sense of human movements. For this reason, the number of real-time fingerprints is estimated by trying to make sense of human hand movements in the developed project. In doing so, the background does not just have to be a solid color. This software, which tries to minimize background noise as much as possible, is created by only signal processing methods and works with high efficiency.*

**Keywords:** *Image processing, hand detection, hand motion identification, number of fingers*

## 1. Giriş

Robot ve insan etkileşimi, yapay zeka uygulamalarının gelişmesi ile birlikte hızla artmaktadır. Bununla birlikte sensör teknolojisi de hızla gelişmektedir. Ancak basit olmayan verilerin dışarıdan alınmasında sensörler ya yetersiz kalmakta ya da maliyetleri çok artmaktadır. Bu sebeple görüntü işleme teknolojisi hem maliyeti düşürmek için hem de bilgisayarların kompleks verileri daha akıllı şekilde yorumlaması için tercih edilmektedir. Basit bir web kamerası olsa bile geliştirilen algoritmalar ile akıllı sistemler kurulabilir. Görüntü işleme ile el algılama ve parmak sayılarını anlamlandırma engelli insanların teknolojik sistem ile etkileşime geçmesine yardımcı olabilir. Örnek olarak, yüksek derecede görme bozukluğu olan bir öğrenci sözel test sorularını sesli dinledikten sonra kameraya cevabını parmakları ile gösterebilir. Ayrıca güvenlik sistemlerinde yüz tanıma ek olarak el hareketleri ile şifre girilmesi düşünülebilir. Daha farklı olarak parmak hareketleri ile işaret dili yorumlayan bir sistem geliştirilebilir.

Projede, insan elinin algılanması için bir çok filtre kullanılmıştır. İlk olarak MOG2 algoritması ile arkaplan eksiltme yapılmaktadır. Sonrasında insan derisi rengini algılamak için özel tasarlanmış bir filtre kullanılarak, insan uzvu olmayan objeler siyah

olarak maskelenmektedir. Bu aşamadan sonra medyan filtresi ile görüntü sinyalleri üzerindeki tek tip olmayan gürültü mümkün olduğunca azaltılmaktadır. Daha sonra HSV renk uzayı dönüşümü ve gri tonlamalı dönüşüm yapılmaktadır. Filtrelemenin son aşaması olarak da Otsu algoritması ikili eşikleme yöntemi ile beraber kullanılarak eşik değeri altında kalanların siyah renk ile maskelenmesi sağlanmıştır.

## 2. Arka plan eksiltme ve deri rengi filtresi

Görüntüdeki her pixel ayrı bir olasılık yoğunluk fonksiyonuna sahiptir. Yeni görüntüdeki pikseller, eski görüntüye göre (arkaplan) farklı bir olasılık yoğunluk fonksiyonuna sahip olacaklardır. Kamera sabit durduğu için arkaplan olarak nitelendirdiğimiz kısmı bir fotoğraf gibi analiz edebiliriz. İstatistiksel modelleme analizinden sonra, piksel yoğunluklarının seviyeleri için varyasyonlar belirlenir. Çünkü varyans pikselden piksele değişim gösterebilir. Bunun için 1 numaralı denklemde belirtilen tekli Gaussian modeli kullanılabilir [3]. RGB renk uzayında ya da başka bir renk uzayındaki bir pikselin zamana göre konumu  $\vec{x}^{(t)}$  notasyonu ile gösterilir. Piksel temelli arkaplan silme işleminde bir pikselin arkaplana (AP) ait olma olasılığı ya da önplana (ÖP) ait olma olasılığını Bayes kuralı (R) ile belirleyebiliriz. Bu işlem aynı zamanda tekli Gauss modelinde de kullanılır.

$$R = \frac{p(AP | \vec{x}^{(t)})}{p(\ddot{O}P | \vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)} | AP)p(AP)}{p(\vec{x}^{(t)} | \ddot{O}P)p(\ddot{O}P)} \quad (1)$$



Şekil 1: Arkaplan eksiltme

İnsan derisi rengini algılamak için geliştirilen bir yöntem RGB, HSV, YCbCr renk uzaylarının bir kombinasyonu şeklinde çalışıp karar veren hibrid bir yöntemdir. RGB renk uzayında insan derisi rengi 3 kanal (kırmızı,yeşil,mavi) tarafından belirgin şekilde ayırt edilememektedir. Ancak gün ışığı gibi etmenler olduğundan RGB renk uzayını,

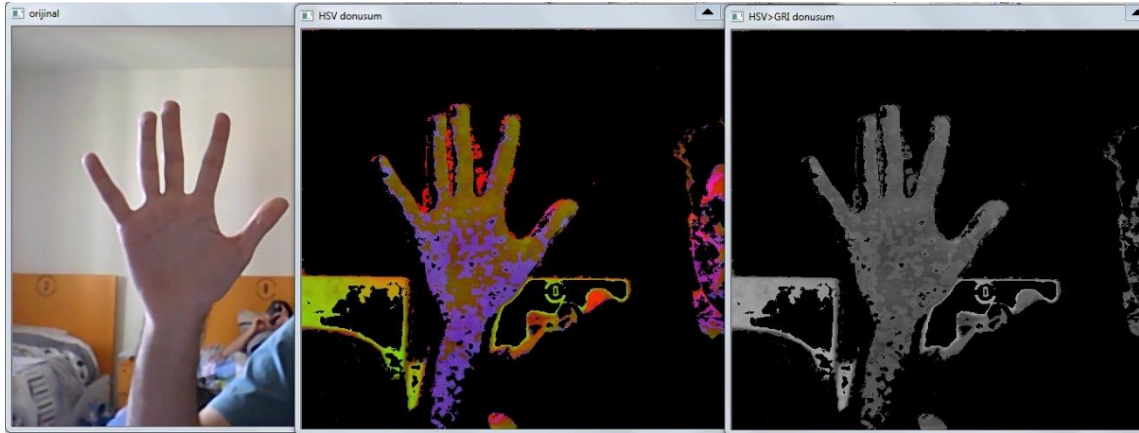
bu etmenlerin etkisini azaltmak için kullanacağız [1]. HSV (hue,saturation,value) renk özü, doygunluk ve parlaklık bileşenlerinden oluşmaktadır.



Şekil 2: Deri rengi filtresinin arkaplan eksiltme ile birlikte kullanılması

### 3. Hsv ve gri tonlamalı dönüşüm

El nesnesine eşik değeri uygulayıp maskeleme yapmadan önce (geometrik işlemlere başlamadan hemen bir önceki aşama), görüntüye Hsv dönüşümü ve gri dönüşüm uygulamamız gerekir.



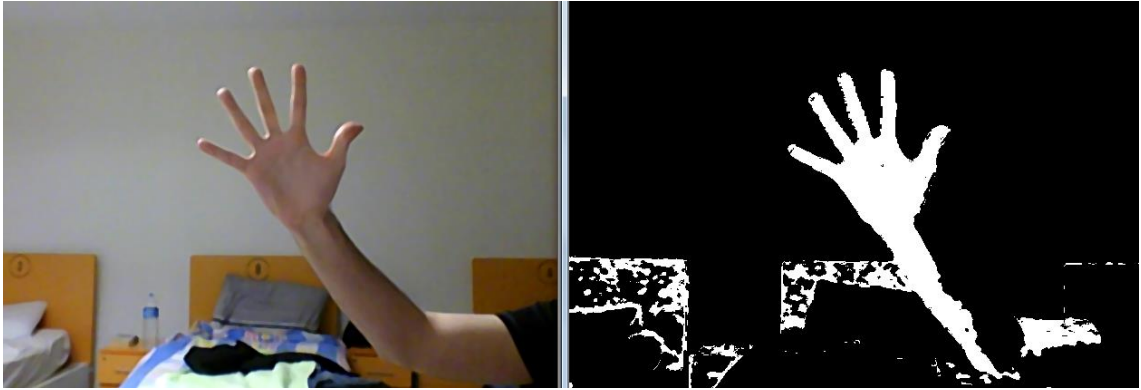
Şekil 3: Hsv renk uzayı dönüşümü ve gri tonlamalı dönüşüm

### 4. Maskeleme

Otsu algoritması, görüntünün bimodal histogramından faydalananarak, görüntüde arkaplan ve önplan şeklinde iki piksel sınıfı olduğunu varsayar. Bimodal histogramı normal bir histogram olarak düşünebiliriz. Ancak normalden farklı olarak iki noktada belirgin bir tepe değerine ulaşan bir histogramdır. Bu sayede otsu algoritması arkaplan

piksellerinin ortalamasının hangi değere daha yakın olduğunu ve önplandaki piksellerin ortalamalarının yaklaşık olarak hangi değere yakın olduğunu bilebilir. Otsu yöntemi 2 numaralı denklemde gösterilen eşitlik ile bu iki piksel sınıfını ayıran optimum eşik değerini hesaplar. Böylece sınıf içi varyans minimum, sınıflar arası varyans maksimum olur [2].

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (2)$$



Şekil 4: Otsu algoritması ve binary eşikleme ile maskeleme

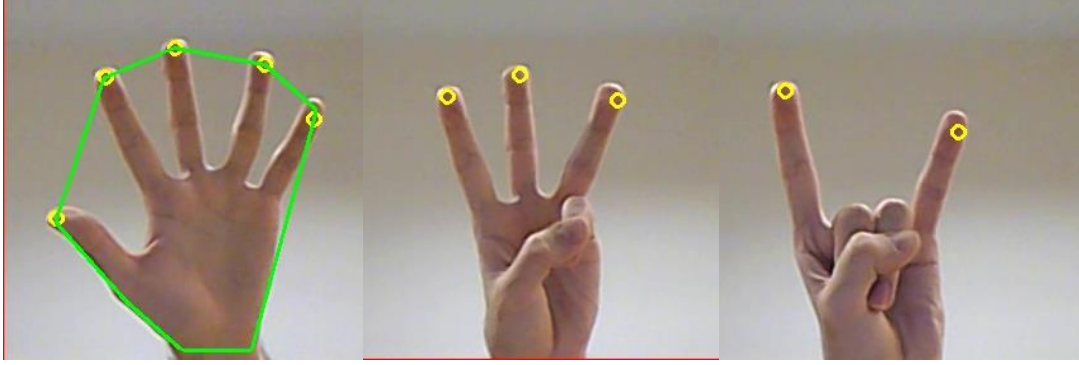
## 5. En büyük konturu bulma ve konveks gövdesi algoritmasının uygulanması

Elimizin görüntüsünün üzerinde geometrik işlemler yapabilmemiz için, filtrelenmiş siyah-beyaz görüntü üzerinde elimizin sınırlarını tespit etmemiz gerekir. Bunun için elimizin sınırlarını (kontur) anlık olarak algılayabilmeliyiz. Bu sayede elimizin sınırlarını belirlerken aynı zamanda elimizi yazılımsal olarak takip (tracking) de etmiş olacağız. Yani elimizi görüntünün belirli bir alanına koymak zorunda değiliz, elimiz nerede olursa olsun kamera algılayacaktır. OpenCV içerisindeki *findContours* komutu ile nesnelerin kenarlarını bulmak mümkün ancak bu komut görüntüdeki bütün kenarları bulduğundan, arkaplanda filtrelerin engelleyemediği bir obje gözükürse bu objenin de kenarları tespit edilecektir. Bunu önlemek için kenarları tespit edilen objelerden, alanı en büyük olan objeyi dikkate alacağız.



Şekil 5: Büyük konturun bulunması

Konveks gövdesi (convex hull) algoritması, konturları algılanan noktalar kümesinin en dış noktalarını birleştiren doğru parçalarını bulan bir algoritmadır. Aynı zamanda bu doğru parçalarının birleştiği noktaları da işlemlerimiz için kullanacağız. Bu noktalara ekstrem noktaları adı verilir. Ekstrem noktalarını saydırarak, görüntüde o an var olan parmak sayısını hesaplayacağız.



Şekil 6: Yeşil çizgiler konveks gövde doğruları ve sarı noktalar ekstrem noktaları

Eşik değerini verirken, maksimum değer 255 olmak üzere arkaplanda çok fazla farklı obje olduğunda 250 değeri, arkaplanda normal düzeyde farklılık olduğunda 215 değeri yazılımın düzgün çalışması için gereken optimum değerlerdir.



Şekil 7: Test sonuçları

## **Kaynaklar**

- [1] Abdul Rahman, N., Wei, K. C., & See, J. (2006). RGB-H-CbCr Skin Colour Model for Human Face Detection.
- [2] Kaehler, A., & Bradski, G. (2016). Otsu's Algorithm. In A. Kaehler, & G. Bradski, **Learning OpenCV 3** (pp. 258-259). O'Reilly Media, Inc.
- [3] Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In Pattern Recognition, **International Conference on Pattern Recognition**, 28-31.