# OpenCV ile Kamera Kalibrasyonu

**Eser Sert[1], Deniz Taşkın[2], Cem Taşkın[3], Nurşen Topçubaşı[4], İrfan Köprücü[2]**

1 Trakya Üniversitesi, Teknik Bilimler Meslek Yüksek Okulu, Edirne

2 Trakya Üniversitesi, Bilgisayar Mühendisliği Bölümü, Edirne

3 Trakya Üniversitesi, Tunca Meslek Yüksek Okulu, Edirne

4 Okan Üniversitesi, Uygulamalı Bil. Yüksekokulu, Bilişim Sist. ve Tek. Bölümü, İstanbul

esersert@trakya.edu.tr, deniztaskin @trakya.edu.tr, cemtaskin@trakya.edu.tr,

nursen.sucsuz@okan.edu.tr, irfankoprucu@gmail.com

**Özet:** Sayısal görüntü işleme ve analizinde kullanılan temel elemanlardan birisi kameradır. Kamera kullanılarak elde edilen resim ya da videolar üzerinde sayısal görüntü işleme ve analiz teknikleri kullanılarak çıkarımlar yapılabilmektedir. Birçok araştırmacı kameradan alınan görüntülerin kesin doğrulukta olduğunu varsaymaktadır. Fakat hassas ölçümler gerektiren uygulamalar, kameradan elde edilen görüntülerin hataları olduğunu göstermektedir. Fotoğraf ya da videoların üzerinde doğru analizler yapabilmek için kamera mutlaka kalibre edilmelidir.

Bu çalışmada, kamera kalibrasyonunu gerçekleştirmek için OpenCV kütüphanesinin kullanımı anlatılmaktadır. OpenCV, Intel tarafından geliştirilmiş, açık kaynak kodlu "Bilgisayarla Görü" kütüphanesidir. Bu kütüphane yardımı ile görüntünün üzerinde gerekli işlemler yapılarak kalibrasyon süreci tamamlanmakta ve düzeltilmiş görüntü elde edilmektedir.

**Keywords:** Görüntü İşleme, Kalibrasyon, OpenCV

**Camera Calibration with OpenCV**

**Abstract:** One of the basic elements used in the analysis and for digital image processing is camera. The inference can be done by using analyzing technics and digital image processing on videos or pictures taken by camera. Many researchers assume that the images taken by cameras are in definite accuracy. However, applications which require accurate measurements show that the images obtained from the camera are mistaken. The camera must be calibrated to make the right analysis on photographs or videos.

In this study, the usage of OpenCV library is described for performing the camera calibration. By the help of this library, the calibration process is completed by doing necessary processes over the image and the corrected image is obtained.

**Keywords:** Image Processing, Calibration, OpenCV

## 1. Giriş

Görüntü analizi ise, yapılan işlemler sonucunda yeni bir görüntü elde edilmeden, görüntüye ait sınıflandırmalar veya ölçümler yapılıyor olması, görüntüyle ilgili istatistikler üretilmesidir. Görüntü analizinde nesnelere ait parametrelerin (şekil, uzunluk, alan, açı, gri-ton ve renk değerleri vb.) ölçülmesi söz konusudur [4].

İşlenmemiş görüntülerde, görüntü alma aşmasında genellikle değişik nedenlerden dolayı bazı bozulmalar, özelliklede geometrik bozulmalar oluşabilmektedir. Görüntü üzerinde ana-

lizler yapılmadan önce kamera kalibrasyonu gerekmektedir [4].

## 2. OPENCV

OpenCV, bir resim ya da video içindeki anlamlı bilgileri çıkarıp işleyebilmek için INTEL tarafından C ve C++ dilleri kullanılarak geliştirilmiş, açık kaynak kodlu bir "Bilgisayarla Görü" kütüphanesidir [2].


**Şekil 1.** OpenCV Bileşenleri

Bu çalışmada OpenCV Kütüphanesi kullanarak kameranın kalibrasyonu OpenCV kütüphanesinin temel bileşenleri Şekil 1' de görülmektedir [1]. Bu bileşenler:

**CV (Computer Vision):** Resim işleme ve analizi için güçlü fonksiyonlar içeren bileşen olup, görsel işlemler için gelişmiş algoritmalar içermektedir.

**MLL (Machine Learning Library)** Makine öğrenmesi için istatistiksel sınıflayıcı ve kümeleme araçlarım içeren bir bileşendir.

**HighGui:** Form gibi bileşenleri oluşturmak için grafik arabirimidir. Aynı zamanda resim ve videolan görüntülemek için giriş/çıkış fonksiyonlarım içeren kütüphanedir.

**CxCore:** OpenCV'de bulunan *cvPoint, cvSize, cvMat, cvHistogram* gibi veri yapılarını içerisinde bulunduran kütüphanedir. Aym zamanda XML desteği de sağlanmaktadır. Bunlara ek olarak OpenCV'de yüz tanıma, kalibrasyon, nesne tanıma gibi işlemler için birçok algoritma geliştirilmiştir [2][5][7].

Intel tarafından geliştirilmiş olan OpenCV Kütüphanesi gerçek zamanlı uygulamalarda hızlı ve başarılı sonuçlar vermektedir. Farklı bilgisayarlarda ve gömülü sistemlerde çalıştırılabilmesinden dolayı endüstride de kullanımı yaygınlaşmaktadır.

## 3. Kalibrasyonun Temelleri

Kamera kalibrasyonu, dış faktörler dolayısı ile bozulmuş olan görüntünün düzeltilerek görüntü işlemeye hazır hale getirilmesi işlemidir. Kalibre edilmemiş bir kamera ile gerçekleştirilen görüntü işleme ve görüntü analiz aşamalan hata içermektedir. Şekil 2'de kalibre edilmemiş ve kalibre edilmiş kameradan alınan görüntüler verilmiştir.

Işınları yansıtmanın basit geometrisini incelemek için iğne delikli kamera modelini ele alınacaktır. Kamerada tek bir noktadan ışınları toplamak için lens kullanılmaktadır. Lensin ışık toplamaktaki problemleri ve bozulmalarından dolayı kameralarda elde edilen görüntüler hatalara açıktır.

Kameralarda kullanılan lensin dezavantajları ve iğne delikli kamera modelleri için sapma gibi faktörler de göz önünde bulundurularak kamera kalibrasyonu yapılmalıdır. Böylece kameradan elde edilen görüntü gerçek dünyadakine en yakın seviyede olacaktır. Kameranın davranışlarını anlayabilmek ve lens bozukluklarının analiz etmek için matematiksel araçlarla inceleme gerçekleştirilmektedir. Dolayısıyla, kameranın öğeleri (pikselleri) ve fiziksel dünyanın öğeleri (metre) modellemek için görüntü matris ve vektörlerinden yararlanılmaktadır. Böylece kameradan çekilen görüntüyle gerçek görüntü arasında daha tutarlı karşılaştırma gerçekleştirilir[1][5].

## 4. Kamera Modeli

Görme olayı ışıkla gerçekleşmektedir. Işık, güneş lamba, fener gibi bazı kaynaklardan elde edilir, sonra bir nesneye çarpana kadar yol alır.

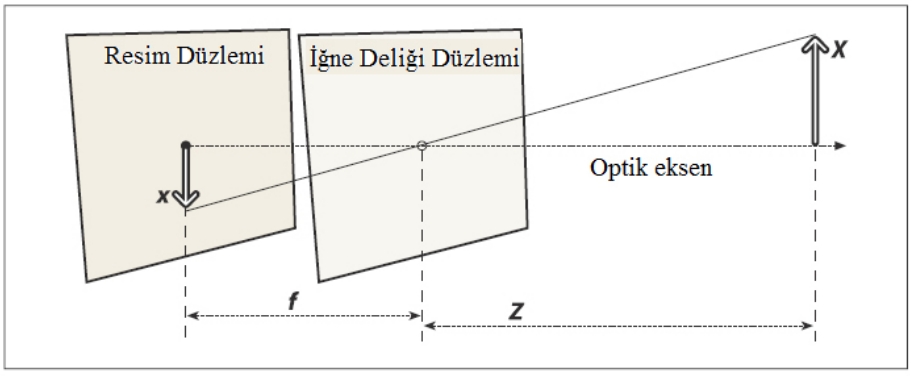
Bir nesneye çarptığında emilir ve yansır. Işık dalga boylarına bağlı olarak renk olarak algılanmaktadır. Objelerden ışınların geçişi sayesinde oluşturduğu hayali görüntü, kamera, göz retinası ve pratik bilgisayar görüntüleyicileri tarafından algılanmaktadır.

Basit bir kamera modeli incelendiğinde kameranın, uzak obje veya sahneleri görme özelliğine sahip olduğu görülmektedir. Odaklama ve uzak nesnelerin mesafesi, kameranın parametreleri ile ilgilidir.

Şekil 3'de görüldüğü gibi f kameranın odak uzaklığı, Z nesneye kameradan uzaklık, X objenin uzunluğu ve x görüntü düzlemi üzerinde objenin görüntüsüdür. Resimde, üçgen —x/f = X/Z, olarak ifade edilmektedir. Optik eksen mesafelerine göre X, x değişmektedir[1].

a) Lens bozukluğuna sahip görüntü b) Lens bozukluğu giderilmiş görüntü

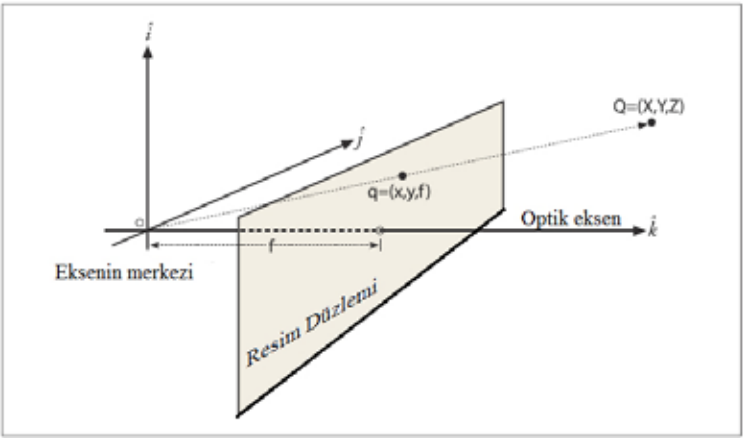
**Şekil 2.** Lens bozukluğunun giderilmesi



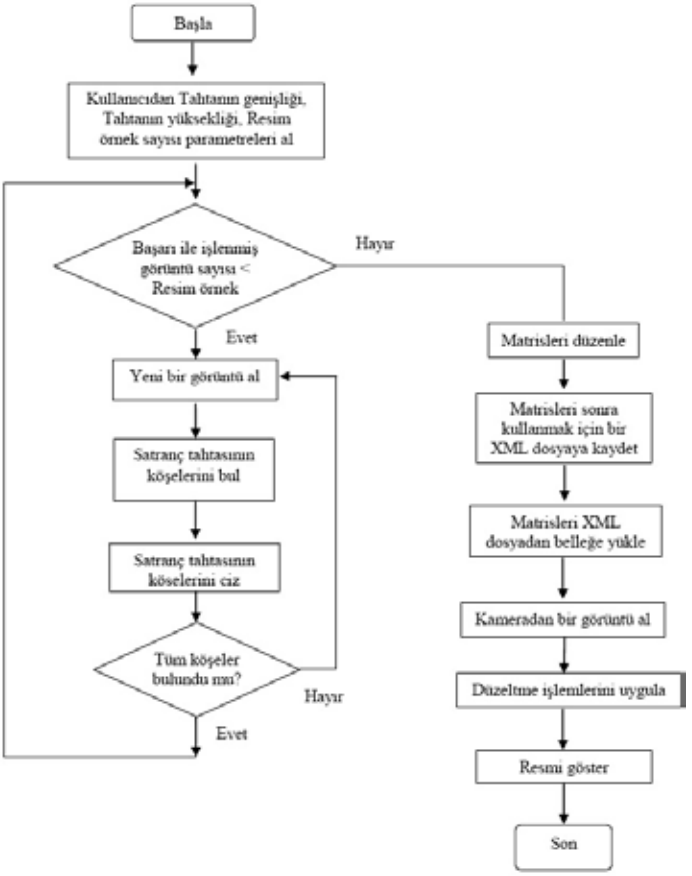

**Şekil 3.** İğne delikli kamera modeli

Resim düzlemi ve optik eksenin kesişimi şekil 4'de görülmektedir. Uzak objelerin resim boyutları gerekli hesaplamalar yapıldıktan sonra bulunabilmektedir. Bu hesaplamalarda x/f=X/Z ilişkisi kullanılmaktadır.

Gerçek dünyada Q=(X,Y,Z) noktası olarak ifade edilebilecek bir nokta, iz düşüm merkezinden geçen ışın aracılığıyla resim düzlemine yansıtılmaktadır. Bu nokta ise q=(x,y,f) olarak resim düzleminde ifade edilebilmektedir [1].




**Şekil 4.** Resim Düzlemi, eksen merkezi ilişkisi




**Şekil 5.** Kalibrasyon parametrelerinin elde edilmesi ve düzeltilmiş resmin görüntülenmesi için akış şeması

## 5. Kalibrasyon Süreci

Kalibrasyon işlemi için nitelendirilebilir bir nesne seçilmelidir. OpenCV bu işlem işçin düzlemsel zeminli nesneler kullanmaktadır. Satranç tahtası bu işlem için idealdir bir nesnedir. Literatürde bazı kalibrasyon metotları üç boyutlu nesneleri kullanmaktadır fakat iki boyutlu bir satranç tahtası bu işlemi yapmak için çok daha pratiktir.

Kalibrasyon işleminin akış diyagramı şekil 5'de görülmektedir [1].

Kalibrasyon süreci aynntılı olarak incelendiğinde aşağıdaki 6 temel işlemin olduğu görülmektedir.

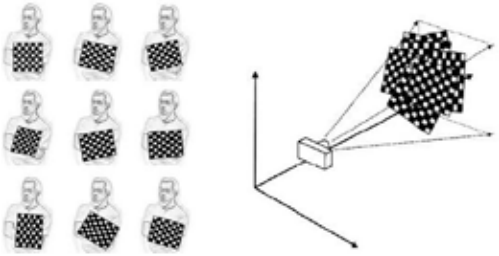
1. Satranç tahtası resmini al
2. Satranç tahtası kesişim noktalarını bul
3. Kesişim noktalarını say
4. Kesişim noktalarını göster
5. Kamerayı kalibre et
6. Düzeltilmiş görüntüyü göster

Bu işlemlerin aşamaları ayrıntılı olarak incelenmiştir.

### A. Satranç Tahtası Resmini Al

Aşağıdaki komut kullanılarak kameradan satranç tahtası resmi alınır.

```
CvCapture* capture = cvCreateCameraCapture(O);
```



**Şekil 6.** Satranç tahtasının fotoğrafının çekilmesi

Kalibrasyon süreci devam ettiği şekil 6'da görüldüğü [1] gibi farklı açılardan yeni resimler alınır. Alınacak resimlerin sayısı lens bozukluğunu giderecek sayıda olmalıdır.

### B. Satranç Tahtası Kesişim Noktalarını Bul

Verilen bir satranç tahtasının görüntüsünde, satranç tahtasının köşelerinin konumları için

```
cvFindChessboardCorners()
```

OpenCV fonksiyonu kullanılmaktadır.

### C. Kesişim Noktalarını Say

İkinci argüman, pattern_size, tahtanın her bir satır ve her bir sütunda kaç köşe olduğunu göstermektedir. Fonksiyon dahili köşelerin sayısını göstermektedir; bu yüzden standart bir oyun tahtası için değerleri

```
cvSize(7,7)
```

belirlemektedir.

Sonraki argüman, corners, köşe konumlarını kaydedebilen bir diziye bir işaretçidir. Bu dizi ön tahsisli olmalıdır ve elbette tahtanın üzerinde tüm köşeler için yeterince büyük olmak zorundadır. Bireysel değerler piksel koordinatları içinde köşelerin konumlarıdır.

*Corner_count* argümanı seçime bağlıdır; *non_NULL* ise, köşelerin sayısı kaydedilebildiyse bir integer bir işaretçidir. Fonksiyon hatalıysa, 0 geri dönecektir. Bitiş *flags* argümanı satranç tahtası üzerinde köşe bulmaya yardım için ek filtre işlemi yapmaktadır. Argümanların herhangi birini veya tümünü birleştirmek için *OR* işlemi kullanılmaktadır.

### D. Satranç Tahtası Köşelerinin Çizimi

Satranç tahtası üzerindeki köşelerin çizdirilmesi çoğu zaman istenmektedir; bu yolla gözlenmiş olan köşelerle bilgisayar tarafından belirlenmiş olan köşelerin eşleşip eşlenmediği belirlenebilmektedir. OpenCV yazılımı ile uygun rutinlerle bu işlem gerçekleştirilebilmektedir.

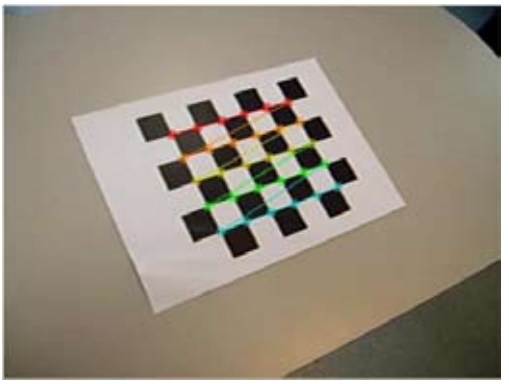
*cvDrawChessboardCorners()* fonksiyonu resim üzerinde *cvFindChessboardCorners()* aracılığıyla bulunan köşeleri çizer. Köşelerin tümü bulunamazsa mevcut köşeler küçük kırmızı hat-

larla temsil edilecektir. Tüm desen bulunduğunda köşeler farklı renklerle gösterilmektedir [1].

```
void cvDrawChessboardCorners(
        CvArr* image,
        CvSize pattern size,
        CvPoint2D32f* corners,
        int count,
        int pattern_wasfound
);
```

*cvDrawChessboardCorners()* fonksiyonuna gönderilen ilk argüman çizimi yapılacak olan resimdir. Köşeler renkli halkalarla temsil edilecektir, bu 8- bit renkli resim olmalıdır; çoğu durumda *cvFindChessboardCorners()* fonksiyonuna resimlerin bir kopyası gönderilmektedir[3].

Sonraki iki argüman *patternsize* ve *corners, cvFindChessboardCorners()* fonksiyonu için tanımlanan argümanlardır. Argüman *count* köşelerin sayısına eşit bir tamsayıdır. *pattern_was_found* argüman'ı tüm satranç tahtası deseninin başarıyla bulunup bulunmadığını göstermektedir. Şekil 7'de *cvDrawChessboardCorners()* fonksiyonunun bir satranç tahtasına uygulandıktan sonra elde edilen sonuç görülmektedir.



**Şekil 7.** Satranç tahtası resmine *cvDrawChessboardCorners()* fonksiyonunun uygulanışı

**E. Kamerayı Kalibre Et**

Kalibrasyon işlemi yapılacak nesnenin birden çok köşeye sahip olduğu durumlarda *cvCalibrateCamera2()* fonksiyonu kullanılmaktadır. Kalibrasyon fonksiyonlarının yürütülmesi aşamasında satranç tahtasının görüntüsü üzerinde gerekli parametrik hesaplamalar yürütülmektedir. OpenCV *cvCalibrateCamera2* fonksiyonu gerekli işlemleri gerçekleştirerek kalibrasyon için gerekli sayısal değerleri vermektedir. Kameranın gerçek matris değerlerini, bozulma katsayısını, dönüş ve çevrim vektörlerini üretmektedir. Bozulma katsayıları (kİ, k2, p1, p2 ve k3) radyal ve teğetsel bozulma eşitliklerindeki katsayılardır. Kameranın gerçek parametreleri en sonunda elde edilmektedir[1].

```
void cvCalibrateCamera2(
        CvMat* object_points,
        CvMat* image_points,
        int* pointcounts,
        CvSize imagesize,
        CvMat* intrinsicmatrix,
        CvMat* distortion coeffs,
        CvMat* rotationvectors = NULL,
        CvMat* translation_vectors = NULL,
        int flags = 0
);
```

**F. Düzeltilmiş Görüntüyü Göster**

Programın ürettiği düzeltilmiş görüntünün gösterilmesi işlemi bu aşamada yürütmektedir.

```
cvShowImage( " Ham resim ", image )
```

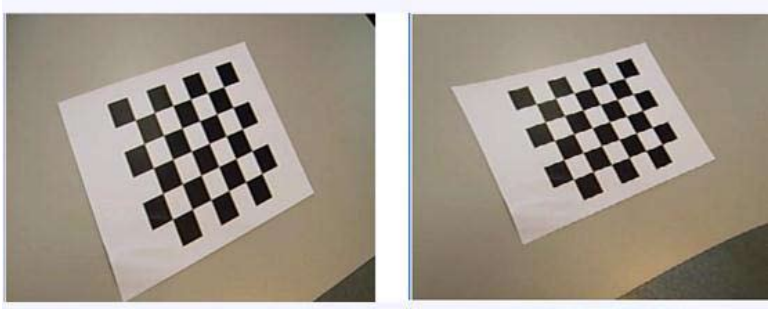
komut satırı ile işlenmemiş görüntü,

```
cvShowImage( " Düzeltilmiş ", image )
```

satırı ile de düzeltilmiş görüntü şekil 8'de görüldüğü gibi ekrana getirilmektedir.

```
while( image ){
        IplImage *t = cvCloneImage( image );
        cvShowImage( "Ham resim", image );
                // Ham resmi göster
        cvRemap( t, image, mapx, mapy);
                //Resmi düzelt
        cvReleaseImage( &t );
```

```
cvShowImage( "Düzeltilmiş", image);
            //Düzeltilmiş resmi göster
int c = cvWaitKey( 15 );
if( c == 'p'){
c = 0;
while(c != 'p' && c != 27){
            c = cvWaitKey( 250 );
            }
```



a) Ham görüntü b) Düzeltilmiş görüntü

**Şekil 8.** Program sonuç görüntüleri

## 6. Sonuçlar

Bu çalışmada temel düzeyde OpenCV kütüphanesi, kamera modeli, kamera kalibrasyonu konuları açıklanmıştır. Kalibrasyon için 2D koordinatlara sahip ve özellikleri bilinen satranç tahtası tercih edilmiştir. OpenCV kütüphanesinin sağladığı yardımcı fonksiyonlar kullanılarak, kalibrasyon süreci konusunda bahsedilen işlemler gerçekleştirilmiştir. Bu işlemler sonucunda kalibrasyon parametreleri elde edilmiştir. Bu parametreler ile kamera kullanılarak kaydedilen görüntüler düzeltilmiştir. Düzeltilmiş görüntülerin gerçek dünyadaki nesnelere en yakın düzeye olduklan görülmektedir.

## 7. Kaynaklar

[1] Bradski, G. and Kaehler, A., "Learning OpenCV: Computer Vision with the OpenCV Library", **O'Reilly Media**, USA, (2008).

[2] Erişti, E, 2010, "Görüntü İşlemede Yeni Bir Soluk, OPENCV", Akademik Bilişim, (2010).

[3] OpenCV Reference Manuals - HighGUI Reference Manual

[4] Taşdemira, Ş., Ürkmez, A., Yakar, M., İnal Ş., "Sayısal Görüntü Analiz İşleminde Kamera Kalibrasyon Parametrelerinin Belirlenmesi", 5.Uluslararası İleri Teknolojiler Sempozyumu (IATS'09)

[5] Wang, Y.M., Li Y., Zheng, J.B., "A Camera Calibration Technique Based On OpenCV", Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference, China, 403-406

[6] Yıldırım, K.S., İnce, C., Kalaycı, T. E., "Görüntü İşleme", Ege Üniversitesi (2003).

[7] Yu, Q., Cheng, H. H., Cheng, W. W., Zhou X., "Ch OpenCV for interactive open architecture computer vision", Advances in Engineering Software, 35: 527-536 (2004).