
GAZİ ÜNİVERSİTESİ
Gazi
1926

TEK KAMERALI STEREO GÖRÜŞ İLE DERİNLİK HESABININ YAPILMASI

Ali MUMCU

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EYLÜL 2016

Ali MUMCU tarafından hazırlanan "TEK KAMERALI STEREO GÖRÜŞ İLE DERİNLİK HESABININ YAPILMASI" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Öğrt. Gör. Dr. Murat HACIÖMEROĞLU

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ...……

Başkan : Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Üye : Yrd. Doç. Dr. Mehmet Serdar GÜZEL

Bilgisayar Mühendisliği, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ……...

Tez Savunma Tarihi: 02/09/2016

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..……

Prof. Dr. Hadi GÖKÇEN

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ali MUMCU
02/09/2016

TEK KAMERALI STEREO GÖRÜŞ İLE DERİNLİK HESABININ YAPILMASI

(Yüksek Lisans Tezi)

Ali MUMCU

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Eylül 2016

ÖZET

Teknolojinin gelişmesi ile birlikte el bilgisayarı olarak kabul edilebilen yüksek performansa sahip mobil cihazlar üretilmektedirler. Bu cihazlar hızlı işlem yapma kabiliyetine, zengin algılayıcılara, büyük ekranlara ve yüksek çözünürlüklü kameralara sahip olmaları sayesinde bilgisayarla görü sistemlerinde de yaygın olarak kullanılmaktadırlar. Bu çalışmada, nesnelerin kameraya olan uzaklığının hesaplanması için kullanılan iki kameralı yaklaşım, stereo kamera donanımına sahip olmayan tek kameralı mobil bir cihaz ile gerçekleştirilmektedir. Aynı sahnenin, cihazın yatayda el ile yer değiştirilmesi sonucu iki resmi çekilerek, nesnelerin kameraya olan uzaklıkları hesaplanmaktadır. Yatayda kameranın hareket yönü ve yer değiştirmesi cihazın içerisindeki algılayıcılar sayesinde tespit edilmektedir. İki resimdeki özellik noktaların elde edilmesi için SURF (Speed Up Robust Features) algoritması kullanılmaktadır. İki resimde de ortak bulunan noktalardan en iyi eşleşen noktalar seçilmektedir. Çıkarılan bu noktalar üzerine ortak nesnenin tespiti için K-ortalamalar kümeleme algoritması uygulanmıştır. Nesnenin, iki resimdeki yer değiştirmesi ve kameranın yataydaki hareket etme mesafesi kullanılarak kameraya olan uzaklığı ölçülmektedir. Bu hesabın yapılması çok hassas olup, gerçekleştirilirken kullanılacak bu iki değerin yüksek doğruluk ile tespit edilmesi işlemin doğru sonuç verebilmesi için çok önemlidir. Tüm bu adımlar Android işletim sistemi ile çalışan mobil cihaz üzerinde geliştirilmektedir. Mobil uygulamalarda doğruluk yanında hız da önemli olduğundan literatürde kabul görmüş ORB (Oriented FAST and Rotated BRIEF) and BRISK (Binary Robust Invariant Scalable Keypoints) özellik belirleme algoritmaları ile bu işlemler tekrarlanarak hız ve doğruluk karşılaştırılması yapılmaktadır.

Bilim Kodu : 92418

Anahtar Kelimeler : Stereo Görüş, Uzaklık Ölçümü, Mobil Hesaplama

Sayfa Adedi : 71

Danışman : Öğrt. Gör. Dr. Murat HACIÖMEROĞLU

DEPTH ESTIMATION USING SINGLE CAMERA STEREO VISION

(M. Sc. Thesis)

Ali MUMCU

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

September 2016

ABSTRACT

In conjunction with the development of technology, mobile devices which can be considered as a handheld computer with high performance are manufactured. Thanks to their fast processing capabilities, rich sensors, big screens and high resolution cameras, these devices are being utilized widely in computer vision systems. In this study, binocular stereo vision system, which can be used to calculate the distance between object and the camera, is performed with a mobile device that is not equipped with stereo camera. Distance of object to the camera is measured by taking two pictures of a scene by manually moving the camera. Camera's lateral displacement and direction of movement are detected using the device's inertial sensors. SURF (Speeded Up Robust Features) algorithm is used in order to detect and descript the feature points in the two images. K-means clustering algorithms are applied on the extracted key points for detection of the common object. Camera displacement and location difference of an object in the images are used to compute the distance of object to the camera. Performing this operation is sensitive, therefore determining these two values with high precision is very critical to be able to retrieve the correct result. All these steps are developed on mobile device with Android operating system. In mobile applications in addition to the accuracy speed is also important, thus this operation is repeated by applying ORB (Oriented FAST and Rotated BRIEF) and BRISK (Binary Robust Invariant Scalable Keypoints) algorithms which are generally accepted in the literature for targeting speed and accuracy comparison.

Science Code : 92418
Key Words : Stereo Vision, Distance Measurement, Mobile Computing

Page Number : 71
Supervisor : Lecturer Dr. Murat HACIÖMEROĞLU

TEŞEKKÜR

Bu çalışmanın planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren saygıdeğer danışman hocam Öğrt. Gör. Dr. Murat HACIÖMEROĞLU'na, yine kıymetli tecrübelerinden faydalandığım Gazi Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü Başkanı Prof. Dr. Şeref SAĞIROĞLU ve öğretim üyelerine sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

1. GİRİŞ

İnsan duyarlılığında, iki ayrı gözün gördüğü iki ayrı resim benzerliklerin eşleştirilip, farklılıkların tamamlanması ile beyin tarafından birleştirilerek üç boyutlu model çıkarılmakta ve derinlik bilgisi elde edilmektedir. Stereo görüşte de insan görüşüne benzer şekilde sahne iki kamera kullanılarak farklı açılardan çekilmektedir. Çekilen resimlerde çok ortak özellik olmakla beraber küçük farklılıklar bulunmaktadır. Çekilen nesne için derinlik bilgisi, stereo resimlerdeki benzerliklerin bulunup eşlenmesi ve bu eşleşmelerin iz düşüm geometrisi kullanılarak işlenmesi ile elde edilmektedir [1-3].

Stereo görüş temassız üç boyutlu hesaplama için etkili bir yaklaşım olarak ortaya çıkmıştır. Uzaklık hesabı, üç boyutlu sahnelerde hedef izlenmesi ve pozisyon tahmininin yapılmasında anahtar bir meseledir [4-7]. Genellikle uzaklık ölçümünde kullanılan lazer mesafe ölçüm cihazlarına kıyasla stereo kameralar ile aynı anda renk ve pozisyon bilgisine de ulaşılabilmektedir. Lazer mesafe ölçme cihazları stereo kameralardan çok daha pahalı olmakla beraber renk bilgisi için fazladan kameraya ihtiyaç duymaktadır. Bundan dolayı stereo kamera kullanımı tek başına bütün sistem için masrafların azalmasına olanak sağlamaktadır[8].

Nesnelerin kameraya ya da bir gözlemciye olan uzaklığının belirlenmesi çevresel algılama için temel işlemdir [9, 10]. Son zamanlarda cihazların hesaplama yapma kabiliyetlerinin artması, stereo kameraların birçok gerçek zamanlı uygulamalar ve bilgisayarla görüş problemleri üzerine uygulanmasına izin vermiştir [11-14]. Derinlik belirlemede de stereo görüş sistemleri bilgisayarla görüş tekniklerini kullanarak, gürbüz ve güvenilir bir şekilde nesnelerinin uzaklığının hesaplanması için kullanılmaktadır [15]. Bu sistemlerde kameraların farklı konumlarda bulunması nesnelerinde resimlerde farklı konumlarda çıkmasına neden olmaktadır. Nesnelerin resimlerdeki farkı ve kameralar arası uzaklık kullanılarak derinlik hesabı yapılmaktadır [16].

Stereo görüş iki ayrı kameranın kullanılması ile sınırlandırılmamalıdır. İki resmin tek bir kamera ile farklı konumlardan çekilmesi de aynı sonuçları verecektir. Bu şekilde tek kameralı stereo görüş oluşturulabilmektedir. Bu sistemlerde geleneksel yöntemlerden farklı olarak kameralar arası uzaklık sabit değil değişken olabilmektedir. Bu değişkenlik mesafe

hesaplamalarında yüksek doğrulukla sonuçlar alınmasına olanak sağlayabilmektedir [15]. Ayrıca iki kameralı stereo görüş sistemlerinde kameraların donanımlarından kaynaklı, kameralardan çekilen iki resim arasında yoğunluk farkına neden olabilecek değişiklikler, odak uzaklığı ve yakınlaştırma seviyesi farklılıkları olabilmesi nedeniyle, nesne uzaklığı hesaplamalarında hatalar olabilmektedir. Bu tür kısıtlar göz önünde bulundurulunca tek kameralı stereo görüş sistemini kullanmak daha avantajlı olabilmektedir [17].

Her geçen yıl mobil marketin çok hızlı bir şekilde büyüdüğü gözlemlenmekledir. Portio Research tarafından yapılan araştırmaya göre bütün dünya geneli mobil abone sayısının her yıl yüzde 7,3 arttığı ve 2016 yılı sonunda 8,5 milyar olacağı tahmin edilmektedir. Ayrıca dünya geneli 2017 yılında uygulama indirilme sayısının 200 milyardan fazla, mobil uygulama pazarının da 63,5 milyar dolar olacağı ön görülmektedir [18]. Mobil cihazların hızlı bir şekilde büyümesi sonucu, "istediğim zaman ve yerde bilgi parmaklarımın ucunda" anlayışıyla mobil uygulamalara olan talepte artmaktadır ve insanlar mobil cihazları ile her şeyi yapmayı ister hale gelmektedirler [19].

Akıllı cihazların kapasitelerinin gelişmesine rağmen mobil donanım bilgisayarlara oranla hala zayıf kalmaktadır. Batarya ömrü, hesaplama kapasitesi, depolama alanı gibi kısıtlar zengin içerikli ya da kaynağa dayalı görüntü işleme ve artırılmış gerçeklik gibi uygulamaların desteklenmesinde zorluklar yaşatabilmektedir [18]. Kapasitesi sınırlı mobil cihazlarda, karmaşık işlemler uzak bulut tarafından desteklenen sunucular yardımı ile gerçekleştirilebilmektedir fakat böyle durumlar internet bağlantısı ile sunuculara bağlanılıp, gerçekleştirilen işlemin cevabının beklenilmesini gerektirdiğinden her zaman verimli olmayabilmektedir. Bundan dolayı, hesaplamaların cihaz içerisinde hızlı bir şekilde az kaynak kullanılarak yapılması tercih edilmektedir.

Bu tez çalışmasında tek kameralı stereo görüş sistemi mobil cihaz kullanılarak cihaz içerisindeki algılayıcılar yardımı ile oluşturulmaktadır. Aynı cihazla farklı açılardan çekilen iki resimde istenilen nesnenin kameraya olan uzaklığı hesaplanmaktadır. Resimlerdeki özellik noktalarının çıkarılıp eşlenmesi sonucu eşlenen noktalar arası fark değeri kullanılarak hesaplama işlemi gerçekleştirilmektedir. Çalışma mobil cihazlarda bu işlemin kabul edilebilir bir hızla ve doğru bir şekilde gerçekleştirilebilmesini amaçlamaktadır. Daha hızlı ve doğru sonuçların elde edilebilmesi amacıyla özellik belirleme için kullanılan algoritmalar değiştirilip, çekilen resim sayısı artırılarak uygulama tekrarlanmıştır.

Bu tez çalışması 5 bölümden oluşmaktadır. İkinci bölümde, stereo görüş, bilgisayarda stereo görmenin temel prensipleri ve resimlerden özellik noktaları çıkarımında kullanılan algoritmalar anlatılmıştır. Üçüncü bölümde stereo kameralar ile uzaklık hesabı yapılmasına yönelik olarak yapılan bilimsel çalışmalar incelenmiştir. Dördüncü bölümde, mobil cihaz kullanılarak tek kamaralı stereo görüş ile derinlik hesabının yapılmasında kullanılan yöntem anlatılmış ve gerçekleştirilen deney sonuçlarına yer verilmiştir. Son olarak beşinci bölümde sonuç ve öneriler ile çalışma tamamlanmıştır.

2. STEREO GÖRÜŞ

Stereo kelimesinin kaynağı Yunan “stereos” kelimesinden gelmektedir ve sabit, hareket etmeyen anlamlarına gelmektedir. Stereo görüş ile insan görüşü modellenip nesnelerin üç boyutlu derinlik kestirimi yapılabilmektedir [20]. Günümüzde dijital kameralar ve bilgisayarlar stereo görüntülerin elde edilmesinde kullanılmaktadırlar.

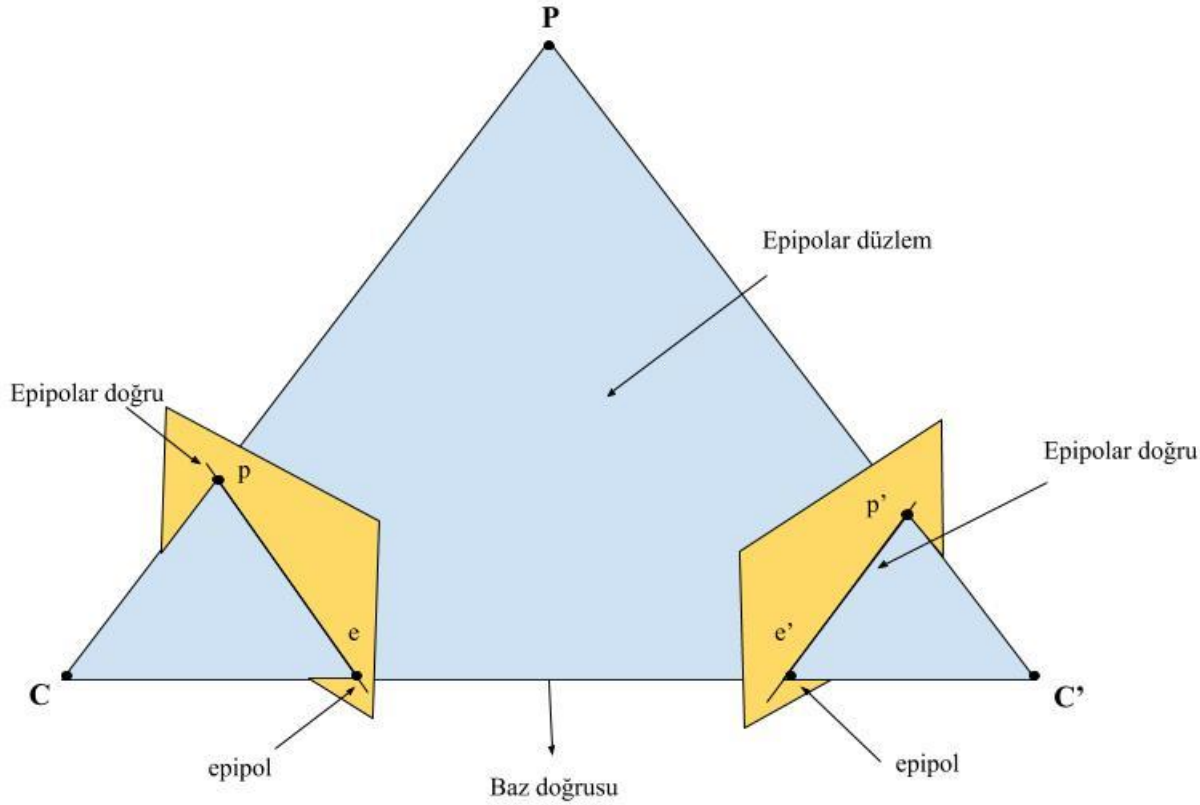
Epipolar geometri

Epipolar geometri genel olarak farklı açılardan çekilmiş resim çiftlerinde bulunan aynı sahnede, resimler ve nesne uzayı arasındaki geometrik ilişki olarak tanımlanmaktadır. Bu geometri resimlerin birbirileri ile olan ilişkisini belirttiği gibi her bir resmin nesne uzayı ile alakalı ilişkisini de ifade etmektedir [21].

Epipolar geometri stereo görüş sisteminin temelini oluşturmaktadır. Şekil 2.1’de gösterildiği gibi, cismin bulunduğu nokta *P*, iki kameranın iz düşüm noktaları *(C, C’)* ve cismin görüntüler üzerinde bulunduğu yerler *(p, p’)* aynı düzlemdedir. Bu düzleme epipolar düzlem denilmektedir [21].

Kamera merkezleri arasındaki doğru ile bu doğrunun görüntüleri kestiği varsayılan nokta *(e, e’)* epipol olarak adlandırılmaktadır. Diğer bir deyişle bir kamera merkezinin diğer görüntüde bulunan iz düşümüne denilmektedir. Bu noktalar ile resim noktalarını birleştiren doğru epipolar doğru olarak adlandırılmaktadır [21].

Kameranın görüş alanında bulunan üç boyutlu noktalar, resimleri epipolar doğru ile kesen epipolar düzlem üzerinde yer almaktadır. Bir resimdeki noktanın diğer resimdeki karşılığı epipolar doğru üzerinde bulunmaktadır. Bu durum epipolar kısıt olarak tanımlanmaktadır. Bu kısıt iki resimdeki karşılıklı noktaların bulunması işlemini kolaylaştırmaktadır. Böylelikle hesaplama yükü ve yanlış eşleştirmelerin önüne geçilebilmektedir [22].

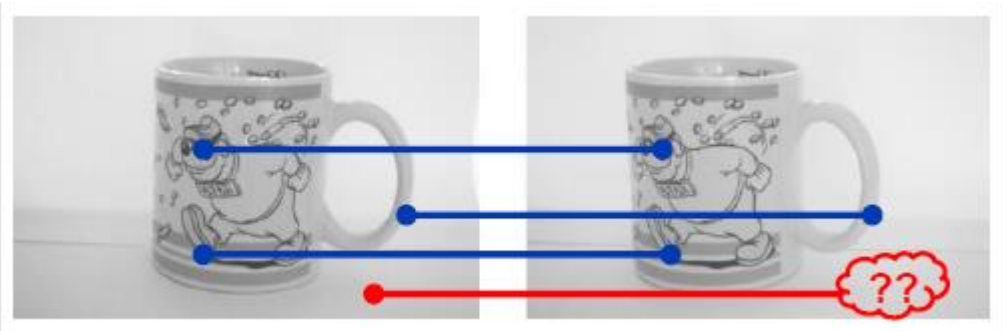


Şekil 2.1. Epipolar Geometri

Stereo görüş aktif ve pasif olarak ikiye ayrılmaktadır [23].

Pasif stereo görüş

Pasif stereo görüş tekniğinde, 2 adet kamera bulunmaktadır. Nesnelerin üzerindeki şekillerden ve desenlerden faydalanılarak, her bir pikselin ne kadar kaydığı hesaplanmaktadır.

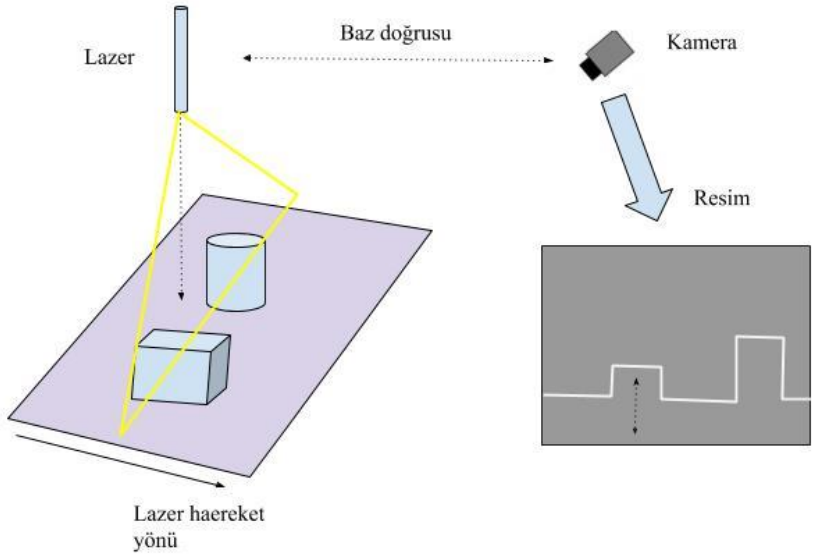


Şekil 2.2. Pasif Stereo Görüş Piksel Eşleme [24]

Örnek olarak bir kupanın sol ve sağ kameralar ile alınmış görüntüsü Şekil 2.2'de gösterilmektedir. Bu resimler kullanılarak kupa üzerindeki hayvanın gözünün sol ve sağ kamerada ne kadar kaydığı, kupanın kulpunun ne kadar kaydığı gibi bilgilere ulaşılabilmektedir.

Aktif stereo görüş

Aktif stereo görüş bir ışıldak yardımı ile nesneler üzerine bilinen örüntüye sahip kızılötesi görüntü göndermek ve stereo görüş kameraları ile bu kızıl ötesi desenleri yakalamaktır. Alanda yeterli desen olmaması durumlarında derinlik bilgisine ulaşılmak için kullanılmaktadır.



Şekil 2.3. Aktif Stereo Görüş

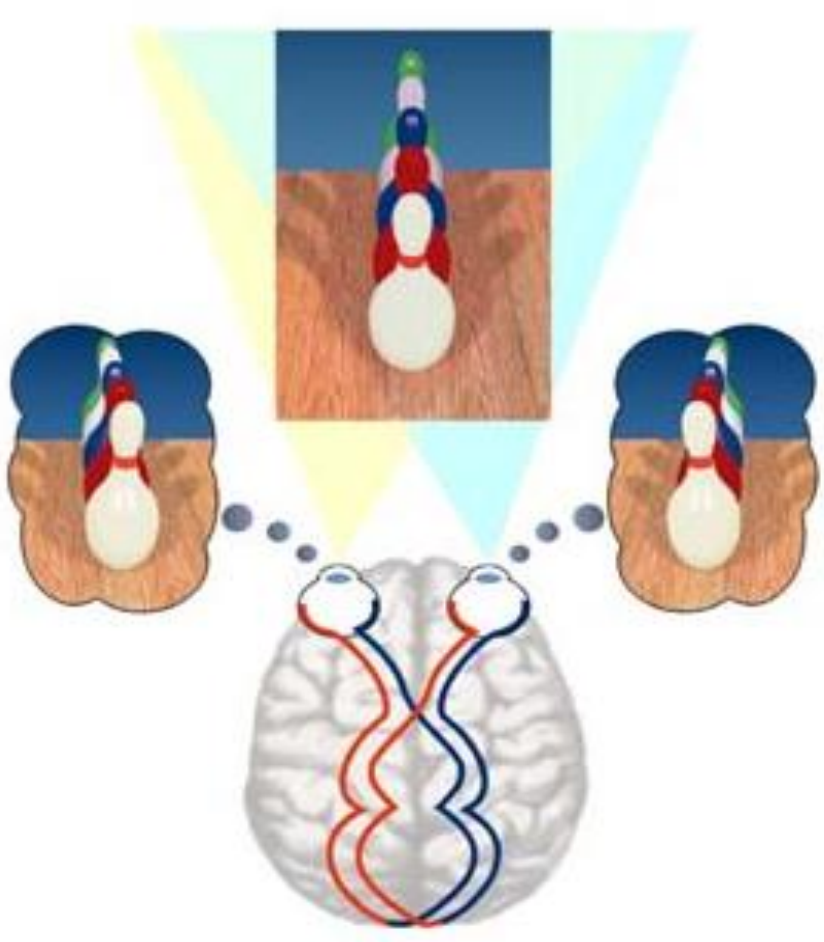
2.1. İnsan Görüşü

İnsanlar bir kafa ile iki göze sahiptirler ve gözler başın ön kısmında yan yana bulunmaktadır. Onların yakın ve aynı hizada olması sayesinde, her bir göz aynı alanı kısmen farklı açılarla görmektedir. İki gözün görüşü birçok benzerliklere sahip olmakla birlikte her bir göz çok kısa zamanda bir resmi yakalayarak farklı görsel bilgi elde etmektedir. Sağ ve sol göz tarafından algılanan iki resim beyine gönderilmektedirler. Beyinde yapılan hesaplamanın ardından, beynimiz bize ne gördüğümüzü göstermektedir. Gözler tarafından yakalanan bu resimlerin onaylanması için bir yol vardır. Sol gözü kapatıp resmi sağ gözle almak ve sağ

gözü kapatıp resmi sol gözle almak. Bu zıtlıklar göz önüne alındığında bu iki resim arasında küçük farklılıkların olduğu görülecektir. Bu farklılıklara dayanılarak, insan beyinin bu iki resmi özel bir teknikle birleştirdiği anlaşılmaktadır [25].

İnsanlar sağ ve sol göz tarafından görülen sahnenin farklılıklarından derinlik bilgisi sonucu çıkarabilmektedirler. Örnek olarak, eğer nesne1 nesne2 üzerinde hareket ediyorsa ve nesne2 daha büyükse bu nesnenin bir kısmı gözükmeyecektir ve böylelikle insan nesne1'in daha yakında olduğunu anlayacaktır [25].

Şekil 2.4'de sağ ve sol göz kendi görüşlerinden resimleri yakaladığından beyne iki farklı resim gönderilmektedir. Beyne gelen bu resimler birleştirilerek tek bir görüntü elde edilmektedir. İnsanda bulunan binokular sistemin incelenmesi ile stereo görüş sistemleri modellenmiştir.



Şekil 2.4. İnsanda Binokular Görüş [25]

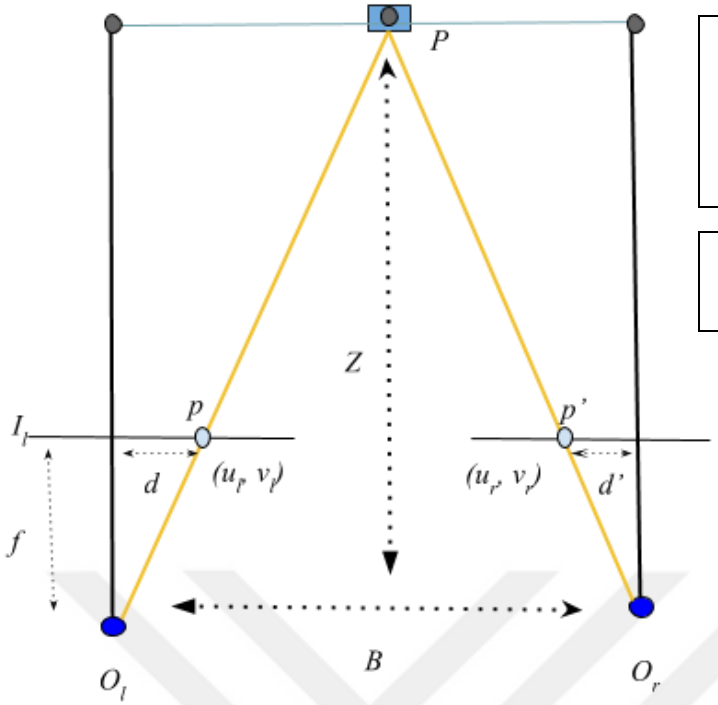
2.2. Bilgisayar da Stereo Görüş

Bilgisayar ile stereo görüş üç boyutlu bilginin dijital resimlerden elde edilmesi işlemidir. İki farklı bakış açısından bir sahne hakkında bilgiyi karşılaştırıp, iki tablodaki nesnelerin birbiri ile ilişkili pozisyonları incelenerek üç boyutlu bilgi elde edilmektedir. Bu işlem insanda görüntünün algılanmasındaki biyolojik sürece benzemektedir [25].

Şekil 2.5'de stereo görmenin genel prensibi açıklanmaktadır. P noktası sahnedeki herhangi bir nokta, O_l ve O_r sol ve sağ kameranın iz düşüm merkezleridir. I_l ve I_r ise her bir kameraya karşılık gelen görüntü düzlemidir. Sadece sol kameranın görüntü düzlemindeki konum bilgileri bilinen bir P iz düşüm noktası, O_l' den P' ye uzanan ışın üzerindeki herhangi bir yerde olabilir. Ancak P noktasının, sağ kamera görüntü düzlemindeki bilgileri de bilindiğinde, üçgenleme tekniğinden yararlanılarak iki ışının kesişiminden P noktasının ortamdaki asıl yeri bulunabilmektedir. B, O_l ve O_r arasındaki yani kameralar arasındaki uzaklık (baseline); (u_l, v_l) ve (u_r, v_l), P'nin sağ ve sol kameranın görüntü düzlemindeki yerleri; f, odak uzaklığı ve Z de, B' den P noktasına olan uzaklık olsun. Üçgenlerin benzerliği kullanılarak Z yani derinlik bilgisi Eş. 2.1'deki gibi hesaplanabilmektedir [26].

$$Z = \frac{(f * B)}{d} \tag{2.1}$$

Eş. 2.1' de, d dışındaki bütün terimler, sistemin bilinen sabit değerleridir. Bir noktanın gerçek uzaklığını bulmak için bu ifadenin hesaplanması gerekmektedir. Kameraların yerleştirilme konumlarına göre nesneler görüntü düzlemlerinde farklı yerlere düşmektedirler. Bu iki noktanın görüntü düzlemlerinde birbirlerine olan mesafeleri, farkı (d, farklılık değeri) vermektedir.



$$\frac{b}{Z} = \frac{b-(d+d')}{b} = \frac{d+d'}{f}$$

$$D = d + d'$$

$$Z = \frac{f * b}{D}$$

Şekil 2.5. Bilgisayarla Stereo Görmenin Temel Prensibi

2.2.1. Stereo kamera

Stereo görüş sistemindeki en önemli elemanlardan birisi stereo kameralardır. Görüntüler işlenip derinlik hesabı yapılırken kamera sisteminin iç ve dış parametrelerine ihtiyaç duyulmaktadır [27].

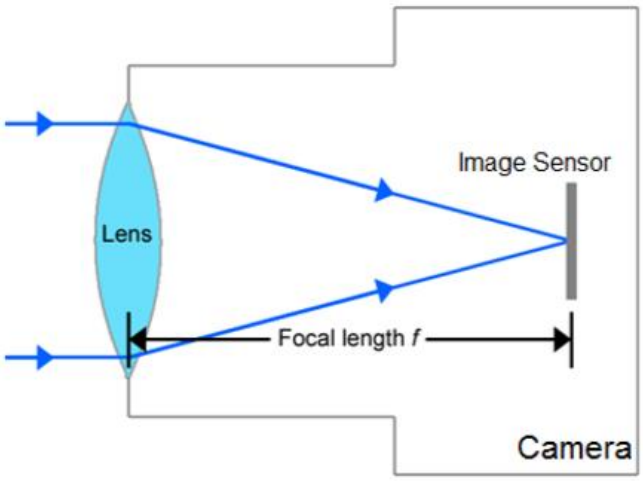
Şekil 2.6. Stereo Kamera [27]

2.2.2. Kameralar arası uzaklık (baseline)

Stereo kamera sisteminde bulunan iki kamera arası uzaklık değerine karşılık gelmektedir. İnsan gözleri arasındaki farka benzer şekilde stereo kamera sisteminde de kameralar arsında uygun bir mesafe bulunmak zorundadır. Sistemin insan gözüne benzetilmesi durumunda bu uzaklık iki göz arası ortalama mesafe olan 6,35 cm olarak alınmaktadır [27].

2.2.3. Odak uzaklığı

Kameranın odak uzaklığı, objektife gelen ışınların toplandığı nokta ile görüntü sensörü arasındaki uzaklığa denilmektedir. Genel olarak mm cinsinden ifade edilmektedir ve derinlik hesabı yapılırken kullanılan en önemli parametrelerden birisidir.



Şekil 2.7. . Kamera Odak Uzaklığı (f) [27]

2.2.4. Farklılık değeri (disparity)

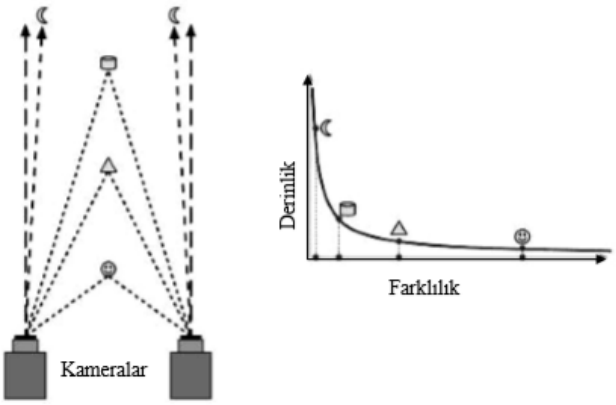
Alanda bulunan bir P noktasının sağ ve sol resimdeki karşılıkları u_l ve u_r olmak üzere Şekil 2.5'den de anlaşılacağı gibi farklılık değeri;

$$d = u_l - u_r \tag{2.2}$$

şeklinde hesaplanabilmektedir. Bu işlem bütün noktalar için yapıldığında ise farklılık haritası elde edilmektedir.

Farklılık değeri derinlik ile ters orantılıdır. Bu ikisi arasında lineer olmayan bir ilişki

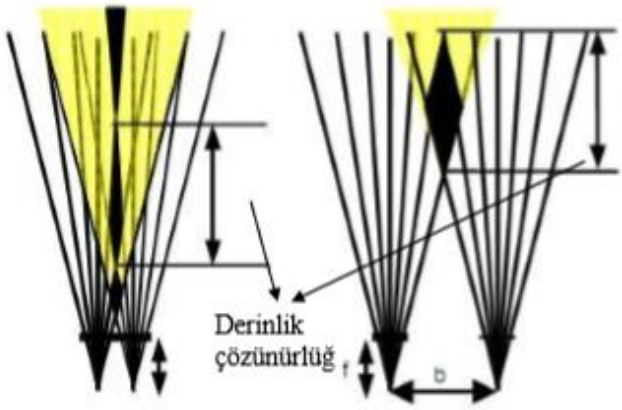
bulunmaktadır. Büyük farklılık değerleri nesneler kameraya yakın olduğunda, küçük farklılık değerleri de nesneler kameraya uzak olduğunda ortaya çıkmaktadır. Şekil 2.8'de bu durum gösterilmektedir.



Şekil 2.8. Farklılık ve Derinlik İlişkisi [22]

Derinlik çözünürlüğü

Kameralar arası uzaklık (baseline) stereo görüş sisteminde elde edilecek derinlik çözünürlüğüne bağlı olarak değişmektedir.



Şekil 2.9. Derinlik Çözünürlüğü [28]

Şekil 2.9'de kameralar arası uzaklığın farklı olduğu durumlarda elde edilebilecek derinlik çözünürlükleri belirtilmiştir. Çözünürlük derinliği kameralar arası uzaklığın artması ile artmaktadır [28].

2.3. Özellik Çıkarımı

Bir resmin yerel alanlara ya da özniteliklere ayrılması bilgisayarla görüş sistemlerinde yerel görünüm özelliklerin elde edilmesinde karmaşıklığı azaltmak için geniş bir şekilde kullanılmaktadır. Nesne tanıma ve eşleştirme, üç boyutlu nesne modellerinin geri çatılması (3D reconstruction) ve hareket takibi gibi uygulamaların başarısı resimlerdeki kararlı, temsil edebilen özelliklerin varlığına bağlıdır. Ayrıca bu özelliklerin belirlenip tanımlanması işleminin kabul edilebilir bir zamanda ve yüksek doğruluk ile gerçekleştirilmesi çok önemlidir [29].

Bilgisayarla görüş uygulamalarında doğal resimden kararlı özellik noktalarının çıkarılması işlemi karşılaşılan en önemli problemler arasındadır. Bu özellik noktaları daha sonra karşılıklı eşleşme gerektiren, stereo ve nesne tanıma gibi uygulamalarda yaygın bir şekilde kullanılmaktadırlar. Aynı sahnenin farklı bakış açılarından ve yönlerden çekilen resimleri arasında kayda değer farklılıklar bulunabilmesinden dolayı çıkartılacak olan bu özellik noktaları ölçek ve rotasyondan bağımsız olması gerekmektedir. Ayrıca kararlı noktaların çıkarılabilmesi için bu noktaların sahnenin ışık durumuna da bağımlı olmaması gerekmektedir [29].

Bu alanda bu amaç için geliştirilen en popüler algoritma, Lowe tarafından gerçekleştirilen SIFT (Scale Invariant Feature Detection) algoritmasıdır [30] ve başarılı bir şekilde birçok uygulamada kullanılmıştır. SIFT algoritmasının yüksek hesaplama yükü sebebiyle özellikle gerçek zamanlı uygulamalarda ve düşük güçlü mobil cihazlarda kullanılması çok verimli olamamaktadır. Bu durum araştırmacıları aynı performansı daha düşük hesaplama maliyeti ile elde edebilecek algoritmaların bulunabilmesine yönlendirmiştir ve böylece SURF algoritması ortaya çıkmıştır [31].

SURF (Speeded Up Robust Features) SIFT algoritmasının tanıtılmasından kısa bir süre sonra farklı bir özellik çıkarımı ve tanımlayıcısı önerilmiştir. SURF algoritması özellik tanımlamadaki hızlı ayarlanmış Hessian kullanması ve yüksek tanımlama oranı ile SIFT algoritmasına, kabul edilen bir alternatif olarak çıkmıştır [29]. SIFT ve SURF algoritmaları bilgisayarla görü üzerinde çalışan kişilerce doğruluğunun ispatlanmış olmasından dolayı sıklıkla kullanılmaktadırlar [32].

Bir diğer taraftan, FAST (Feature from Accelerated Segment test) özellik çıkarım algoritması [33] ile BRIEF (Binary Robust Independent Elementary Features) özellik tanımlayıcı algoritması [34] üzerinde geliştirilen ORB (Oriented FAST and Rotated BRIEF) algoritması Rublee ve diğerleri tarafından önerilmiştir [35]. ORB algoritması ismini bu iki algoritmadan almıştır. SIFT ve SURF algoritmalarına alternatif olarak tasarlanmıştır. ORB özellik çıkarımı FAST algoritmasının uzantısı, ORB özellik tanımlayıcısı da BRIEF algoritmasının geliştirilmiş halidir [36].

Resimlerden uygun özelliklerin seçilebilmesi iki rekabet eden zorluğun dengelenmesinde yatmaktadır. Bu iki zorluk; yüksek kaliteli çıkarım ve düşük hesaplama gerekliliklerdir. Bu amaca hizmet edebilmek için BRISK (Binary Robust Invariant Scalable Keypoints) [31] adındaki algoritma 2011 yılında önerilmiştir. Anahtar noktaların elde edilip tanımlanabilmesi için önerilmiş bir yöntemdir ve SIFT ile SURF yaklaşımından daha hızlı çalışmaktadır [36].

Özellik çıkarımı ve tanımlayıcı algoritmaları ayrı ayrı iki grupta taoplanabilmektedirler. Özellik çıkarım algoritmaları Blob Detectors ve Corner Detectors olarak, özellik tanımlayıcılarda binary ve non binary olarak ikiye ayrılabilmektedir [37]. Çizelge 2.1' de bahsedilen bu algoritmalar gruplandırılmış olarak gösterilmektedir.

Çizelge 2.1. Algoritmaların gruplandırılması

Algoritma	Özellik Çıkarım		Özellik Tanımlama	
	Blob Detectors	Corner Detectors	Binary Descriptors	Non-binary Descriptors
SURF	√			√
SIFT	√			√
FAST		√		
BRISK		√	√	
ORB		√	√	

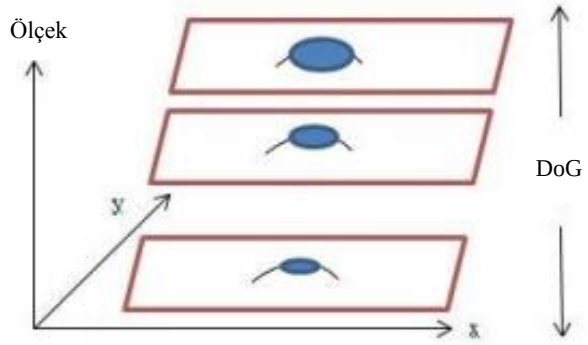
Bu bölümde özellik noktalarının seçilip tanımlanabilmesi için kullanılan SIFT, SURF, ORB ve BRISK algoritmaları ayrıntılı bir şekilde anlatılacaktır.

2.3.1. SIFT

SIFT (Scale Invariant Feature Transform) bir resimde ölçeklendirme, döndürme ve aydınlatmaya karşı değişmeyen bölgesel özellikleri belirleyip tanımlayan bir algoritmadır. SIFT algoritması, takip eden dört adımla tanıma işlemini gerçekleştirmektedir.

Ölçek uzayında uç değer tespiti

Hesaplamanın ilk aşaması anahtar noktaların bulunması ile başlamaktadır. Bunun için, verilen resme farklı ölçeklerde Gaussian süzgeci uygulanmaktadır. Gaussian süzgeci ile bulanık hale gelen bu resimler arasındaki farklar alınmaktadır. Farklı ölçeklerde alınan Gaussian farkının (DoG) ekstremum noktaları bize anahtar noktalarını vermektedir.



Şekil 2.10. Ölçek Uzayında DoG [38]

Anahtar noktanın yerini belirleme

Ölçek uzayındaki ekstremumların buluma yöntemi, aralarında bulunduğu yer konusunda kararsız olunan çok sayıda anahtar nokta sunmaktadır. Algoritmadaki takip eden adım, iyi bir şekilde konumlanamamış ya da düşük zıtlığa sahip noktaları elemine edilmesini sağlamaktadır.

Doğru pozisyonun bulunabilmesi için komşu veriye interpolasyon yapma

İlk olarak, yerini doğru bir şekilde tespiti edebilmek için her bir anahtar nokta adayına interpolasyon uygulanmaktadır. Bu işlem yapıldıktan sonra belirlenen yeni bölge, anahtar noktanın bulunması gereken yer olarak atanmaktadır. Daha sonra, bu adım herhangi bir değişim olmayana kadar, anahtar nokta adayının ölçeği değiştirilerek tekrarlanmaktadır.

Kenar tepkilerini elimine etme

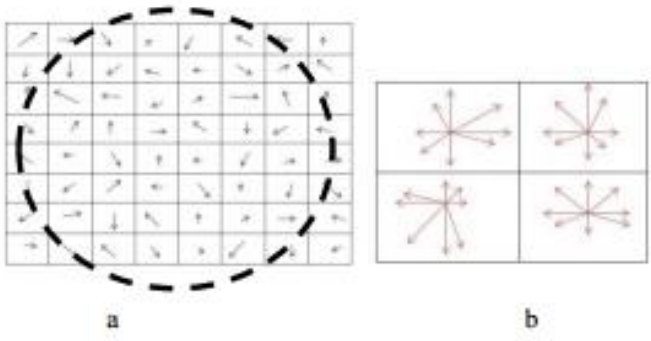
Aday anahtar noktalar küçük miktardaki parazitlere karşı zayıf olsalar bile DoG fonksiyonu, bu noktaları tespit edebilmektedir. Bu durumun düzeltilmesi için kenara güçlü tepki veren fakat karasız olan anahtar noktalarının elimine edilmesi gerekmektedir. İkinci dereceden Hessian matrisinin Eigen değerleri kullanılarak elimine işlemi yapılmaktadır [38].

Yönelim tespiti

Bu adımda, her bir anahtar noktaya, yerel resimdeki eğim yönüne bakılarak bir ya da daha fazla yönelim atanmaktadır. Gaussian uygulanarak bulanık hale gelmiş bu resimde, anahtar noktaların etrafındaki komşu bölgede yer alan her piksel için, eğimin büyüklüğüne ve yönüne bakılmaktadır. Her bir kutusu 10 derece içeren 36 kutuluk bir histograma bulunan değerler yerleştirilmektedir. Histogramın zirve noktası, aranan yönelimi belirtmektedir [38].

Anahtar nokta tanımlayıcıları

İlk olarak, her biri 8 kutu içeren 4x4 piksel komşuluğunda dönüşüm histogramları oluşturulmaktadır. Anahtar nokta etrafındaki 16x16'lık bir bölgede yer alan, dönüşüm ve büyüklük ile ilgili bilgi veren histogramlar hesaplanmaktadır. Her bir histogram gerçek komşuluk bölgesinin 4x4'lük bir alt bölgesini içermektedir. Her biri 8 kutu içeren, 16(4x4) histogramlar, toplamda 128 adet vektör belirtmiş almaktadırlar. Bu vektörler, anahtar nokta tanımlayıcılarını temsil etmektedirler [38].



Şekil 2.11. a) Büyüklük ve oryantasyonları b) Anahtar nokta [30]

2.3.2. SURF

Bu algoritma ilişkili özellik noktalarını elde etmek için birkaç evreden oluşmaktadır. Tek bir evre takip eden adımları içermektedir [29].

a. Hızlı kutu süzgeci için çok az bir hafıza kullanımı ile Viola ve Jones [39] tarafından tanıtılan Intergral Image oluşturulmaktadır.
b. Hessian tabanlı ölçek uzay piramidi (SURF detektör) oluşturarak aday özellik noktalarının aranmaktadır.
c. Elde edilen aday noktalardan yüksek kontrastlı kararlı noktaların elde edilebilmesi için maksimum olmayan bastırma evresi uygulanarak filtreleme ve azaltma işlemi uygulanmaktadır.
d. Karakteristik yönleri bulunarak her bir özelliğe yöneliminin atanması gerçekleştirilmektedir.
e. Rotasyon bağımsızlığını kazanabilmek için karakteristik yönlere dayalı özellik vektörü (SURF tanımlayıcısı) hesaplaması yapılmaktadır.
f. Işık durumları değişiminden etkilenmemek için özellik vektörü normalize edilmektedir.

Tekbir ölçekte özellik yerleşimi için Laplacian of Gaussians (LoG), SIFT algoritmasında kullanılan Difference of Gaussians (Dog) ve SURF algoritmasında kullanılan determinant of the Hessian özellik belirleyicileri kullanılabilmektedir. Ölçeklemeye karşı dayanıklılığın sağlanabilmesi için Guassian süzgeci çekirdeği farklı sigma değerleri ile kullanılmalıdır. Ayrıca bu süzgeç farklı ölçekler için farklı olacak şekilde uygulanmalıdır. Süzgeç boyutu büyüdükçe, komşu pikseller arasında küçük farklılıklar olmaktadır. Bundan dolayı hesaplama karmaşıklığı, N farklı ölçekten oluşan, octave olarak atfedilen her bir seviye piramidinde, ölçek uzayı piramidi oluşturularak azaltılabilmektedir. Her bir octave

çözünürlüğü, x ve y yönlerinde bir önceki octave değerinin yarısı kadar olmaktadır [29].

Resimde ve ölçek üzerinde ilgi noktalarının konumlandırılabilmesi için maksimum olmayan bastırma (NMS) 3 * 3 * 3 komşuluğunda, ölçek uzayı piramidine uygulanmaktadır. Hessian matrisinin determinantının uç noktaları Brown ve Lowe tarafından önerilen yöntem [40] ile ölçeğin ve resim uzayının arasına eklenmektedir.

Bir sonraki adım ilgilenilen noktanın etrafındaki dairesel bölgelerdeki bilgi ile tekrar üretilebilir yönelimi düzeltmemeyi içermektedir. Bu amaç için, ilgilenilen noktaların etrafındaki dairesel komşuluk bölgesinde x ve y istikametindeki Haar-wavelet cevapları hesaplanmaktadır. Dairesel bölge ve örnek adımların boyutu, belirlenen özellik noktasındaki ölçekle orantılı olarak belirlenmektedir. Wavelet cevapları hesaplanıp, ilgi noktasındaki Gaussian ağırlıklandırıldığında cevaplar apsis boyunca yatay cevap kuvveti ve ordinat boyunca dikey cevap kuvveti şeklinde uzayda vektör olarak temsil edilmektedirler. Ardından dominant yönelim 60^o'lik açıyı kapsayan kayan yönelim penceresi ile bütün cevapların toplamı hesaplanarak tahmin edilmektedir. Bu pencere ile yatay ve dikey cevaplar toplanmaktadır. En uzun vektör ilgi noktasında yönelimi vermektedir.

Son olarak her bir özellik için bir tanımlayıcı oluşturulmaktadır. Bu adım ilgilenilen noktayı merkez kabul eden bir kare alan inşa edilerek ve bir önceki adımda seçilen yön esas alınıp yönlendirilerek gerçekleştirilmektedir. Bu alan düzenli olarak 4 * 4 kare alt alanlara bölünmektedir. Her bir alt alan için 4 karakteristik değer, 5 * 5 örnek noktalardan hesaplanmaktadır. Bu örnek noktalarda hesaplanan *dx* ve *dy* Wavelet cevapları ilgilenilen nokta merkezli Gauusian ve her bir alt alanın toplanması ile ağırlandırılarak özellik vektörü için girdilerin ilk kümesi oluşturulmaktadır. Yoğunluk değişiminin polaritesi hakkında bilgi alınabilmesi için cevapların mutlak değerlerinin $|dx|$ ve $|dy|$ toplamı çıkarılmaktadır. Böylece her bir alt alan 4 boyutlu tanımlayıcı vektöre sahip olmaktadır, $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ (64 uzunluklu 4 * 4 alt alanlar için tanımlayıcı vektör). Wavelet cevapları ışıklandırma etkisine karşı dayanıklı hale gelmektedirler. Tanımlayıcıların birim vektörüne çevrilmesi ile yoğunluk değişimine karşı duyarlı hale gelinmesi başarılmaktadır [29].

2.3.3. ORB

ORB algoritması FAST algoritması ve BRIEF algoritmasının hibridi olan bir algoritmadır. Algoritma FAST algoritmasını kullanarak anahtar noktaları bulur ve BRIEF algoritmasını temel alan bir yöntemle bu anahtar noktalarından öznitelikleri çıkarır. ORB algoritmasının diğer algoritmalara göre en önemli ve etkin yanı resmin farklı yönlerde çevrilmesinden ve resmin gürültülü olmasından etkilenmeyişidir. Ayrıca bilgisayarla görü alanında popüler olan SIFT algoritmasından da iki kat daha hızlı çalışmaktadır [41].

ORB algoritması;

- FAST algoritmasına hızlı ve doğru yönelme bileşeni eklemektedir.
- Döndürülmüş BRIEF özelliklerinin verimli bir şekilde hesaplamaktadır.
- Döndürülmüş BRIEF özelliklerinin değişimini ve ilgileşimini analiz etmektedir [35].

FAST algoritması

FAST algoritması gerçek zamanlı sistemlerde SIFT, HARRIS gibi algoritmaların çok işlemci zamanı aldığı gerçeğini ortaya koyarak, gerçek zamanlı sistemler için geliştirilmiş hızlı bir köşe bulma algoritmasıdır. Bu algoritmanın örnek bir şeması aday noktanın etrafında on altı pikselin dairesine (üç piksel yarıçaplı) dayanmaktadır. Eğer dairedeki on iki devamlı pikselin kümesi eşik değeri ile nokta pikselin yeğinlik değeri toplamından daha aydınlık ya da daha karanlık ise bu nokta köşe olarak sınıflandırılmaktadır. Daha sonra eğitim kümesi üzerinde makine öğrenmesi ile karar ağacı oluşturulmaktadır. Böylece FAST algoritmasının köşe nokta sınıflandırıcısı elde edilmiş olunmaktadır [33].

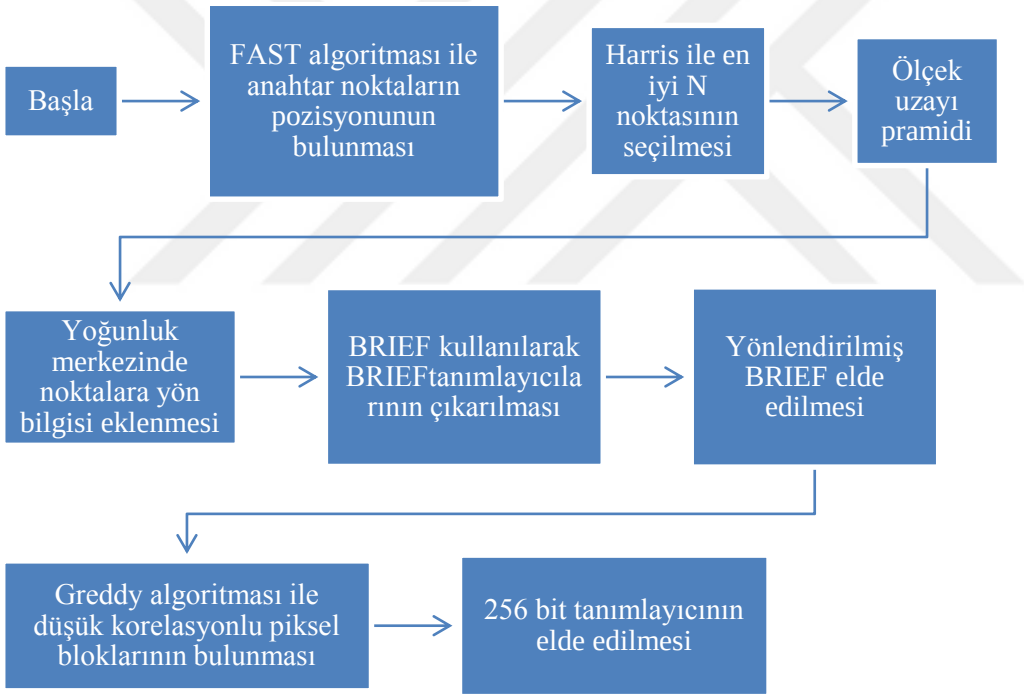
BRIEF algoritması

Özellik noktalarının yönleri ile birlikte elde edilmesinden sonra, BRIEF tanımlayıcısı tarafından geliştirilen tanımlayıcılar çıkarılmaktadır. İkili kodlama yöntemi ile BRIEF özellik noktalarının etrafında belirleyicileri çıkartmaktadır. Bu belirleyiciler SIFT ve SURF algoritmalarından daha basit ve daha az yer kapsamaktadırlar [34].
Bu yaklaşım yoğunluk farkı testinden elde edilen ilk ikili (binary) resim tanımlayıcısıdır. Bu teste göre, eğer belirli bir yerde bulunan yoğunluk değeri diğer bir belirli yerdeki yoğunluk değerinden büyükse 1 değilse 0 değerini almaktadır. BRIEF tanımlayıcıları önceden tanımlı

parçadaki bir anahtar nokta etrafında yoğunluk farkı testini rastgele seçilen bir numarada (128, 256 veya 512) fakat sabit yerlerde bulunan çiftlere uygulayarak şekillendirmektedir. BRIEF tanımlayıcıları aydınlanma değişimlerine karşı dayanıklı iken dönme ve ölçeklemeye karşı dayanıklı değildir [42].

ORB, FAST tanımlayıcısını uygulayarak belirgin potansiyel nokta konumlarını resim piramidinde bulmaktadır. Harris köşe hesabına göre bulunan noktaları sıralamakta ve en üst noktalar kümesini belirgin noktalar olarak almaktadır. Noktaların yönleri yoğunluk merkezi kullanılarak hesaplanmaktadır. ORB tanımlayıcısı, BRIEF tanımlayıcısını geliştirmekte ve bağlı ikililerin yoğunluklarını karşılaştırarak ikili dizi vektörünü oluşturmaktadır [35].

ORB algoritmasının akış diyagramı Şekil 2.12'de gösterilmektedir.



Şekil 2.12. ORB Algoritması Akış Diyagramı

2.3.4. BRISK

BRISK, SURF algoritmasına göre daha iyi performans gösteren ve işlemci zamanı olarak daha az zaman alan bir algoritmadır. Daha hızlı çalışmasını FAST algoritmasında kullanılan anahtar nokta bulma yöntemini uygulaması ve her bir anahtar nokta komşuluğunda bulunan piksel yoğunluklarının karşılaştırılmasında bit dizlerini kullanması sağlamaktadır [31].

Bir resimden anahtar noktaların çıkarımında algoritma aşağıdaki iki adımı uygulamaktadır.

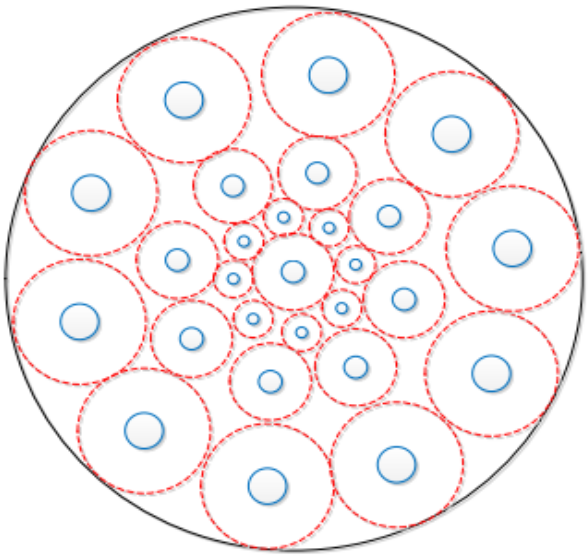
Ölçek uzayı anahtar nokta çıkarımı

İlgili noktalar ölçek uzay ve resim boyunca çıkıntı kıstasları kullanılarak saptanmaktadırlar. Hesaplamanın verimliliğini artırmak için anahtar noktalar piramidin oktave seviyesinde belirlenmektedir. Her bir anahtar noktanın bulunduğu konumu ve ölçeği sıralı domainde karesel fonksiyon ile elde edilmektedir.

Anahtar nokta tanımlayıcısı

BRIEF algoritmasının aksine, BRISK tanımlayıcıdaki bitleri üretmek için önceden tanımlı örnek desen içermektedir. Bu karşılaştırmalar Şekil 2.13'de görülmektedir, kısa ikililer dairesel desen oluşturmaktadır. Noktaların kısa ikilileri belirlenen eşik değerinin altında uzaklığa sahip olanlar tanımlayıcı olarak kullanılmaktadırlar, uzun ikililer ise daha büyük uzaklığa sahiptirler ve dönüş bilgisi için kullanılmaktadırlar.

BRIEF tanımlayıcısının aksine BRISK dönmeye karşı dayanıklıdır. Bu durum hesaplama karmaşıklığına neden olmaktadır fakat hareket eden nesnelerin daha iyi takip edilmesini sağlamaktadır. Dönme ilk olarak örnek alan için gradyan kullanılarak hesaplanmaktadır. Bu işlem tamamlandığında, örnek noktalar ve karşılaştırmalar bu dönmeye göre geliştirilmektedir. Bunun anlamı, bir nesne döndüğünde, orijinal de olduğu gibi gradyan üretmesidir. Örnekleme gradyan boyunca yapıldığından, dönmüş ya da orijinal anahtar nokta için bit dizisi aynı olmaktadır ve bu durum BRISK algoritmasını dönmeye karşı dayanıklı yapmaktadır [31].



Şekil 2.13. BRISK Örnekleme Deseni [31]

2.4. Stereo Eşleme

Bir sahnenin 3 boyutlu yapısını elde edebilmek için farklı bakış açılarından çekilmiş stereo görüntülerinde, görülen her noktanın uzaydaki koordinatlarının bulunması gerekmektedir. Bir noktanın 3 boyutlu uzaydaki yerinin tespiti için bir görüntü üzerindeki her bir noktanın diğer görüntü üzerindeki eşinin bulunması gereklidir. Bu açıdan stereo eşleme problemi aslında bir arama problemi olarak düşünülebilmektedir [43].

Stereo eşleme problemi, üç boyutlu bir noktanın farklı görüntü düzlemlerinde karşılık geldiği noktayı bulmaya çalışmaktadır. Eşlenen noktaların konumları arasındaki fark, stereo uzaklık (disparity) değerini vermektedir. Görüntü düzlemindeki her bir nokta farklılıklarını temsil eden bir başka görüntü elde edersek buna da stereo farklılık haritası (disparity map) adı verilmektedir. Stereo farklılık haritasında açık renkli bölgeler yani fark değeri yüksek olan bölgeler kameraya yakın olan bölgeleri, koyu olan bölgeler yani fark değeri düşük olan bölgeler ise kameraya uzak olan bölgeleri temsil etmektedir [43].

Şekil 2.14. Tsukuba Stereo Görüntüleri (a) Sol Kamera Görüntüsü (b) Uzaklık Haritası [43]

Görüntü eşleme algoritmaları alan tabanlı ve özellik tabanlı olmak üzere ikiye ayrılarak sınıflandırılabilmektedir [44].

Özellik tabanlı eşlemelerde resimlerden köşeler, dış hatlar gibi ilgilenilen özellikler bulunarak bunlar farklı resimlerde eşleştirilmektedirler. Eşleştirme işlemi hızlı olmaktadır çünkü özellik olarak sadece küçük sayıda resim pikselleri seçilmektedir. Bu tür eşleşmedeki eksiklik yanlış eşleşmelerin oluşması ve eşleşme noktalarının seyrek olarak dağılmış olmasıdır. Yine de özellik tabanlı eşleme nesne tanıma, robotik haritalama, görüntü birleştirme, işaret tanıma gibi uygulamalarda yaygın bir şekilde kullanılmaktadır [44].

Belirlenen özellik noktalarının eşleştirilmesinde kullanılabilecek yaklaşımlardan birisi FLANN (Fast Library for Approximate Nearest Neighbour Search) [45] tabanlı eşleştirmedir. Bu yaklaşım yüksek boyutlu özellikler ve büyük veri setleri için hızlı en yakın komşu araması için optimize edilmiş algoritmaların birleşmesinden oluşmaktadır. FLANN bulunan özellik noktalarının tanımlayıcılarını girdi olarak almaktadır. Bu kütüphane, sadece veri seti ve kullanıcı tarafından sağlanan seçilmiş hassaslığa göre en yakın komşu araması için otomatik olarak doğru algoritmanın ve parametrelerin seçilmesinde avantaj sağlamaktadır. FLANN ikinci resimde belirlenen özelliklerle ilk resimde belirlenen özellik noktalarının eşlenmesi için uygulanmaktadır ve ters olarak FLANN bu işlemin tersi şeklinde de uygulanmaktadır. Bundan dolayı kalan noktalar iki operasyonda da aynı şekilde eşleşen noktalar olmaktadır [46].

3. BENZER ÇALIŞMALAR

Nesnelerin uzaklıklarının hesaplanması için yöntemler aktif ve pasif olarak ikiye ayrılabilmektedir. Aktif yöntemler hesaplama işlemini nesneye sinyaller göndererek yaparken pasif yöntemler sadece nesnelerin pozisyon bilgisini elde etmek için kullanmaktadırlar [47]. Pasif yöntemler arasında derinlik hesabı için kullanılabilecek en iyi yöntemlerden birisi stereo görüştür[48]. Isobe ve diğerleri tarafından yapılan çalışmada stereo kamera kullanılarak mobil robotlar için insanı takip eden sistemin geliştirilmesi tartışılmıştır. Hedef insan üzerine yerleştirilen renkli alan esas alınarak 3 boyutlu derinlik bilgisini elde etmektedirler [49].

Tek kamera kullanılarak yapılan bir çalışmada mobil telefon kamerası kullanılarak stereo kamera düzeneği oluşturmuşlardır. Telefonun yer değiştirmesi sonucu iki resim çekilmekte ve bu resimler yardımı ile merkezde bulunan nesnenin uzaklık hesabı yapılmaktadır. Resimlerdeki özellik noktalarını SURF algoritması kullanarak bulmaktadırlar. Deneysel sonuçlara göre iki kamera arasındaki mesafenin artması ile daha yüksek doğrulukta sonuçlar elde etmişlerdir [15].

Ogura ve diğerleri tarafından yapılan çalışmada stereo kamera kullanılarak uzak mesafe hesaplama yöntemi sunulmuştur. Yaptıkları çalışama da gemi ile tesisi arasındaki kontrolü otomatik olarak sağlamaktadırlar. Mesafe hesaplama sisteminin verimliliğini gemi üzerine yerleştirdikleri iki kamerayı kullanıp test etmişlerdir. Yaptıkları deneylerde hedef nesneyi veri tabanlarında bulunan nesnelerle karşılaştırarak bulmaktadırlar. 20 ile 100 metre arasına yerleştirilmiş hedef nesnelerin mesafelerini yüzde bir altında hata ile hesaplamakladırlar [50].

Stereo görüş sistemlerinde nesnelerin iki boyutlu resimlerinden üç boyutlu modellerinin oluşturulabilmesi için sadece dijital kameralar kullanılabilmektedir. Bu şekilde temassız olarak üç boyutta nesnelerin ölçümleri gerçekleştirilebilmektedir. Lazer tarama gibi özel cihazlar üç boyutlu sahnenin oluşturulmasında fazladan bilgiler sunmaktadırlar fakat bu cihazların kullanılması pahalı olabilmektedirler. Comlekciler ve diğerleri ortognatik cerrahi operasyonlar için yapay, temassız ölçüm tabanlı, ölçüm hata oranlarını en aza indirmeyi amaçlayan bir çalışma gerçekleştirmişlerdir. Çalışmada diş cerrahisinde kullanılabilecek

yeni bir yöntem sunmaktadırlar. Yapay zekâ yöntemleri kullanılarak operasyon zamanlarında çene pozisyonlarının belirlenme doğruluğunu artırmayı amaçlamışlardır [51].

Birkaç ayna ve tek kamera kullanılarak stereo kamera sistemini oluşturabilen basit teknikler bazı araştırmacılar tarafından önerilmiştir [52-57]. Nurmu ve Nandi tarafından gerçekleştirilen çalışmada tek kamera kullanılıp nesnenin kamera sistemine olan uzaklığı ile fark haritası çıkarılmaktadır. Önerdikleri sistemde yeterli kadar uzunlukta eğrilik yarıçapı olan iki tümsek ayna kullanmışlardır. Geleneksek stereo kamera sistemine benzer şekilde bu iki ayna yardımı ile aynı sahnenin farklı açılardan resimleri çekilmektedir. Tek kamera kullanılması ölçümleme ve düzeltme işlemlerini kolaylaştırmaktadır. Sistemin yüksek maliyetli stereo kameralarla karşılaştırılabilecek yeterli performans sağladığını, ucuz ve robotik uygulamalarda kullanılabilir olduğunu savunmaktadırlar [17].

Bir diğer yapılan çalışmada iki ilişkili kamera arasındaki uzaklık stereo resim çiftleri ve kamera ölçümleme sonucu elde edilen bilgiler kullanılarak tespit edilmektedir [58]. Boonkwang ve Saiyod tarafından yapılan çalışmada 3 boyutlu stereo tekniği kullanılarak robotlar için uzaklık hesaplaması yapılmaktadır. İki kamera ile odaklanılan nesnenin mesafesi, kameralar ve nesne arasındaki açı ile trigonometrik formüller kullanılarak hesaplanmaktadır. Yapılan deneyeler sonucu elde edilen hata oranının yüzde %10'un altında olduğunu gözlemlemişlerdir [59].

Stereo kamera kullanılarak nesnenin uzaklığını ve boyutunu hesaplayan bir çalışmada ise sabit sahneye ölçümü yapılacak nesne eklenip resimleri çekilmektedir. Çekilen resimlerden sahnenin ilk hali çıkarılarak nesnenin tespiti yapılmaktadırlar. Nesnenin resimlerdeki yer değiştirmesi ve piksel değerlerini kullanarak ölçümleri gerçekleştirmektedirler. Mesafe hesabında ±25cm, boyut hesabında ±3cm hata ile sonuçlar elde etmektedirler [7].

Jian ve arkadaşları tarafından gerçekleştirilen çalışmada elma toplama robotlarının pozisyon belirlemedeki doğruluğunu artırmak amacıyla bir yöntem geliştirilmiştir. Nişanlanan elmalar kullanılarak, işaret tespiti yapılıp elmaların uzamsal pozisyonlarını elde etmektedirler. Görüntüleri iki kamera arası uzaklığı 39.3mm olan Logitech stereo kamera ile almaktadırlar ve öbek eşleme algoritması ile resimleri eşleştirmektedirler. Elde ettikleri sonuçlar 0.63% hata oranı ile elma pozisyonlarını belirlediklerini göstermektedir [60]. Xia ve diğerleri tarafından gerçekleştirilen çalışmada ise stereo görüş tabanlı üç boyutlu yaprak

pozisyonu hesaplama yöntemi önerilmiştir. İki kameralı stereo görüş sistemini yukarı ve aşağı hareket edebilen robot koluna bağlı tek kamera yardımı ile gerçekleştirmişlerdir. Robot kolu yardımı ile hareket eden kameradan elde edilen resimlerde Birchfield's algoritmasını kullanılarak fark haritasını çıkarıp derinlik bilgisine ulaşabilmektedirler. Elde ettikleri kamera koordinat bilgilerini robot koordinat sistemine çevirip yaprakların pozisyonlarına ulaşmaktadırlar [61].

Diğer bir yapılan çalışmada tüm yönlü stereo görüş kullanılarak sürücülere yardımcı bir sistem oluşturulmuştur. Bu sistem sürücülere sürüş sırasında karşılaşabilecekleri beklenmedik durumlar için bilgi vermektedir. Tüm yönlü kamera 360 derece görüş yaptığından araç çevresindeki yol durumlarını kontrol edebilmektedir. Öndeki aracın tespiti için Histogram of oriented gradient (HOG) algoritmasını kullanmaktadırlar. Aracın sağ arka kısmı iki kamera tarafından da tespit edildiğinde uzaklık hesabı işlemini gerçekleştirmektedirler. Yapılan deneyler sonucu 25cm den az bir hata ile uzaklık tespitini gerçekleştirmişleridir [62]. Lai ve arkadaşları tarafında yapılan diğer bir çalışmada ise araba çarpmalarının önüne geçebilmek amacıyla öndeki nesneni belirlenmesini amaçlamaktadırlar. Stereo kameradan alınan resim dizileri kullanılarak statik arka plan çıkarımı ile önde bulunan nesnenin tespiti yapılmaktadır. Tespit ettikleri nesnenin üç boyutlu pozisyonu stereo görüş teknikleri ile yönünü ise bir sonra çekilen resim yardımı ile bulmaktadırlar [63].

Shin ve arkadaşları tarafından yapılan çalışmada mobil stereo kameralar ve uygulamaları için gerçek zamanlı derinlik aralığı algoritması sunulmuştur. Önerdikleri algoritma, farklı eşleme pencere boyutları özellikleri üzerinde basit derinlik birleştirme tekniği kullanarak verimli bir şekilde derinlik gürültülerini kontrol etmektedir. Algoritmanın uygulanması sonucu dinamik derinlik aralığını düşük hesaplama karmaşıklığı ile elde etmişlerdir [64].
Bu çalışma ile tek kameralı mobil bir cihaz kullanılarak stereo görüş sistemi yardımcı bir alete ihtiyaç duyulmadan gerçekleştirilmektedir. Oluşturulan stereo kamera sistemi ile nesnelerin kameraya olan uzaklıkları hesaplanmaktadır. Mobil cihaz ile nesnelerin mesafesi kabul edilebilir bir hızla bulunmaktadır. Çekilen resimlerdeki ilgi noktalarının tespiti ve betimlemesi için kullanılan algoritmalar değiştirilerek uygulama tekrarlanmakta, hız ve doğruluk karşılaştırılması yapılmaktadır.

4. TEK KAMERALI STEREO GÖRÜŞ İLE DERİNLİK HESABININ YAPILMASI

Bu tezde, nesnelerin uzaklıklarını hesaplamaya yardımcı olan stereo görüş sistemi, stereo kamera donanımına sahip olmayan tek kameralı mobil cihaz kullanılarak oluşturulmuştur. Nesnelerin uzaklığı, aynı sahnenin kameranın yatayda el ile hareket ettirilmesi sonucu çekilen iki resmi kullanılarak hesaplanmaktadır. Kameranın hareket etme mesafesi ve hareket yönü cihaz içerisindeki algılayıcılar sayesinde tespit edilmektedir. İki resimdeki özellik noktalarının çıkarılıp tanımlanması için SURF algoritması kullanılmıştır. SURF'ün kullanılma sebebi bilgisayarla görü üzerinde çalışan kimselerce sıklıkla kullanılıyor olması ve doğruluğunun ispatlanmış olmasıdır [32]. İki resimde de çıkarılıp tanımlanan özellik noktaları eşleştirilmektedir. Yanlış eşlenen noktalar tespit edilip, eleme işlemi gerçekleştirilmektedir. Şekil 4.1'de sistemin gerçekleştirilme adımları anlatılmıştır.
Mesafe hesaplama işleminde Eş. 2.1'de belirtilen formül kullanılmaktadır.
Bu eşitliğin sağlanabilmesi için;

- Resimleri çekerken kullanılan kameranın odak uzaklığının (focal length),
- İki kamera arası mesafenin (baseline),
- Farklılık (disparity) değerinin bilinmesi gerekmektedir.

Hesaplama işlemi bu üç değere bağlı olduğundan, bu değerlerin doğru bir şekilde tespit edilmesi gerekmektedir.

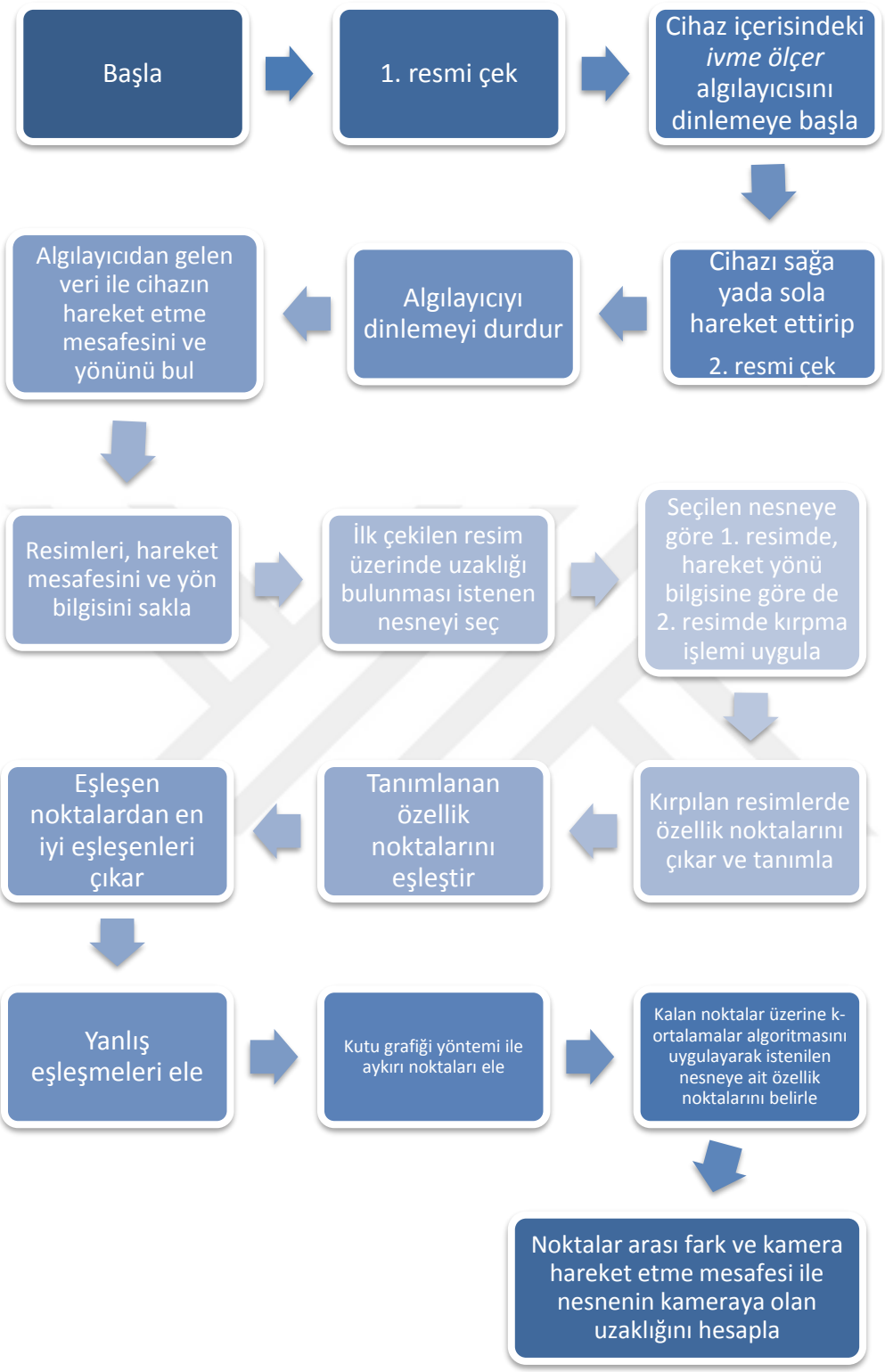
Çalışma gerçekleştirilirken Android işletim sistemine sahip mobil cihaz kullanıldığından, cihazın odak uzaklığı işletim sisteminin sunduğu fonksiyonlar yardımı ile belirlenmektedir. Sistem gerçekleştirilirken tek kamera kullanıldığından kameralar arası mesafe sabit değil, değişken olmaktadır. Bundan dolayı bu değerin her seferinde yüksek doğrulukla belirlenmesi gerekmektedir. Farklılık değeri iki resimde de aynı yere karşılık gelen noktalar arasındaki farka denilmektedir ve işlemin en önemli parametrelerindendir. Bu değer iki resimden çıkarılan özellik noktaları kullanılarak tespit edilmektedir.

Bu çalışmanın amacı mobil cihazlarda mesafe hesaplama işlemini yüksek doğrulukla ve hızlı bir şeklide gerçekleştirebilmektedir. Bu amaç doğrultusunda, kameranın hareket etme

mesafesi ile farklılık değerinin yüksek doğruluk ile bulunması gerekmektedir. Ayrıca, cihazın algılayıcılarından gelen veriler anlık olduğundan hesaplama işleminde geçen süre farklılık değerinin bulunması ile yani özellik noktalarının tespit edilip eşlenmesi ile doğru orantılı olmaktadır.

İşlemlerin gerçekleştirilmesinde en büyük zorluk algılayıcılardan gelen verilerin yorumlanmasında karşılaşılmıştır. Algılayıcılardan dinlenen değerler çok hassas olduğundan kameranın yataydaki hareket etme mesafesini doğru bir şekilde hesaplayabilmek için cihazın yatayda sabit bir hızla hareket ettirildiği varsayımı yapılmaktadır.

Sistem mobil cihaz üzerinde tasarlandığından ve mobil cihazlar bilgisayarlara göre hesaplama yapma kabiliyeti bakımından daha zayıf olduğundan dolayı hesaplama karmaşıklığının en aza indirilmesi hız açısından çok önemli olmaktadır. İşlemlerin doğruluğu ve hızı özellik noktalarının bulunup tanımlanması ile ilişkili olduğundan, uygulama bölüm 2.3'de değinilen, ilgi noktası tespiti ve betimleyicisi olan, literatürde yaygın olarak kullanılan ORB ve BRISK algoritmaları ile tekrarlanmış, hız ve doğruluk karşılaştırılması yapılmıştır. Ayrıca daha doğru sonuçların alınıp alınamayacağının belirlenmesi için uygulama üç ve dört resim çekilerek tekrarlanmıştır.



Şekil 4.1. Uygulama gerçekleştirme adımları

4.1. Uygulama Yazılımı

Bu bölümde yazılımın gerçekleştirilmesi için kullanılan teknolojiler ve yazılımın

gerçekleştirilme adımları açıklanmaktadır.

4.1.1. Kullanılan teknolojiler

Yazılım Android işletim sistemi ile çalışan 3GB RAM' e sahip, Exynos 5 Octa işlemci ve 1.3 GHz Cortex A7 ile çalışmakta olan mobil cihaz ile geliştirilmiştir. Cihaz 2560 x 1600 (WQXGA) çözünürlüğüne sahiptir.

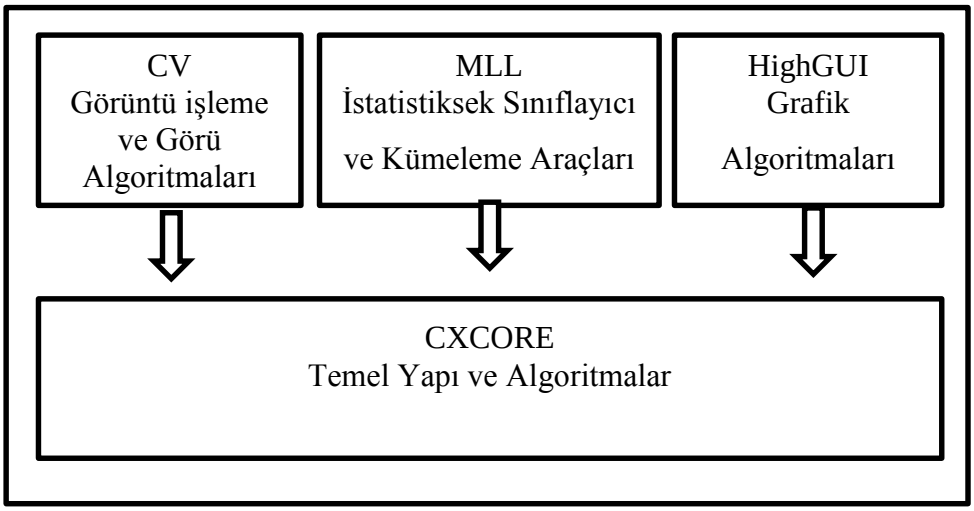
Bu çalışmada mobil Android uygulama geliştirebilmek için Google tarafından geliştirilen, içerisinde Java Development Kit (JDK), Android Virtual Device (AVD) ve mobil uygulama için gerekli ortam ayarlarının oluşturulduğu Android Studio uygulaması kullanılmıştır [65, 66].

Ayrıca bu tez kapsamında, Intel'in açık kaynaklı görüntü işleme kütüphanesi olan OpenCV (Open Source Computer Vision) kütüphanesinden yararlanılmıştır [67].

OpenCV (Open Computer Vision)

Intel tarafından geliştirilmiş Windows, Linux, Mac OS X, PSP (PlayStation Portable) platformları üzerinde çalışabilen, C diliyle yazılmış, açık kaynak kodlu bir "Bilgisayarla Görme" kütüphanedir. Amacı bir resim ya da video içindeki anlamlı bilgileri çıkarıp işleyebilmektir [68].

Bünyesinde bulundurduğu fonksiyonların birçoğu platformdan bağımsız olarak çalışmaktadır. 2.0 sürümünden sonra, C ara yüzüne ek olarak C++ ara yüzü de eklenmiştir. OpenCV akademik ve ticari kullanımı ücretsizdir [68].



Şekil 4.2. OpenCV Bileşenleri

OpenCV kütüphanesi, beş temel bileşenden oluşmaktadır [69].

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 araçlarını içeren bir bileşendir.

HighGui

Form gibi bileşenleri oluşturmak için grafik arabirimidir. Aynı zamanda resim ve videoları görüntülemek için giriş/çıkış fonksiyonlarım içeren kütüphanedir.

CxCore

OpenCV'de bulunan cvPoint, cvSize, cvMat, cvHistogram gibi veri yapılarını içerisinde bulunduran kütüphanedir.

4.1.2. Yazılım gerçekleştirme adımları

Bu bölümde yazılımın gerçekleştirilmesi adım adım anlatılacaktır.

Odak uzaklığının belirlenmesi

Uygulamada tek seferlik kullanılan cihazın odak uzaklığı bulunmaktadır. Odak uzaklığının bulunmasında *android.hardware.camera2.CameraCharacteristics* sınıfı kullanılmaktadır. Bu sınıfta bulunan *LENS_INFO_AVAILABLE_FOCAL_LENGTHS* alanı ile cihazın odak uzaklığı belirlenmektedir [70].

Resimlerin çekilmesi

Nesnelerin uzaklıklarının belirlenebilmesi için aynı sahnenin cihazın yatayda hareket ettirilmesi ile iki resminin çekilmesi gerekmektedir. Ayrıca iki resim arasında kameranın ne kadar hareket ettirildiğinin de hesaplanması gerekmektedir. Resimlerin çekilme işlemini gerçekleştirebilmek için Android Studio aracılığıyla TakePhotoActivity isminde yeni bir

Activity sınıfı oluşturulmuştur. Bu sınıf ile resimlerin çekilmesi, kaydedilmesi, iki resim arasında algılayıcılar yardımı ile kameranın hareket etme mesafesinin ve yönünün hesaplanması işlemleri gerçekleştirilmektedir.

Bu sınıfta başlangıç değeri 0 olan bir sayaç tutulmaktadır ve resim çekilmeden önce ve sonra sayacın değeri 1 arttırılmaktadır. Kullanıcı tarafından resimler çekilmek istendiğinde ilk olarak bu sayacın değeri kontrol edilmektedir. Eğer sayacın değeri 0 ise ilk resmin, 1 ise ikinci resmin çekileceği anlaşılmaktadır. Sayacın değerine göre çekilecek olan resimlerin adları "FIRST" veya "SECOND" olarak atanmakta ve sayacın değeri bir arttırılmaktadır. Daha sonra, resmin çekilebilmesi için aşağıda bulunan kod parçası çalıştırılmaktadır. Bu kısım resmin çekilebilmesi için kamerayı açmakta ve *f* parametresi ile resmin cihazda kaydedilmesi istenen yerin yolunu almaktadır.

```
startActivityForResult(new Intent(
        MediaStore.ACTION_IMAGE_CAPTURE).putExtra(
        MediaStore.EXTRA_OUTPUT, Uri.fromFile(f)),
    Take_Photo);
```

Kullanıcı tarafından resim çekildikten sonra kod *onActivityResult()* metodu ile devam etmektedir. Bu metot içerisinde sayacın değeri tekrar kontrol edilmektedir. Eğer sayacın değeri 1 ise ilk resmin çekildiği, 2 ise ikinci resmin çekildiği anlaşılmaktadır. Sayacın değeri 1 ise yer çekiminden bağımsız ivme değerlerini veren *Lineer İvmeölçer* algılayıcısı dinlenmeye başlanmaktadır. Eğer sayacın değeri 2 ise ikinci resmin de çekildiği anlaşıldığından algılayıcının dinlenmesi bırakılmaktadır.

Bu kısımda iki resmin çekilmesi arasında dinlenilen algılayıcı hakkında kısa bilgi verilmekte ve mesafe hesaplama işleminin nasıl yapıldığı anlatılmaktadır.

Algılayıcılar ve kamera hareket etme mesafesinin hesaplanması

Birçok Android işletim sistemine sahip cihaz algılayıcılar ile donatılmıştır. Bu algılayıcılar hareketi, yönelimi ve çeşitli çevresel durumları ölçmektedirler. Algılayıcılar yüksek duyarlılık ve doğrulukla ham veri sağlama özelliğine sahiptirler. Cihaz hareketinin üç boyutlu monitörlenmesi, pozisyon bilgisi veya cihaz yakınındaki çevresel değişimlerin

gözlemlenmesinde algılayıcıların kullanılması yararlı olmaktadır. Android platformu algılayıcıları üç ana kategori altında toplamaktadır.

a. Hareket Algılayıcıları

Bu algılayıcılar ivmelenme ve rotasyon gücünü üç eksen boyunca ölçmektedirler. Bu kategori, ivmeölçer, yer çekimi algılayıcısı, jiroskop ve dönel vektör algılayıcılarını içermektedir.

b. Çevresel Algılayıcılar

Bu algılayıcılar, yakındaki hava sıcaklığı, basınç, ışık durumu ve nem gibi çeşitli çevresel parametreleri ölçmektedirler. Bu kategoride, barometre, ışıkölçer ve termometreler bulunmaktadır.

c. Pozisyon Algılayıcıları

Bu algılayıcılar cihazın fiziksel pozisyonunu ölçmek için kullanılmaktadırlar. Bu kategoride, yönelim algılayıcısı ve manyetometre bulunmaktadır.

Android algılayıcı çatısı

Android algılayıcı çatısı ile algılayıcılara ve onların sağladığı ham verilere ulaşmak mümkün olmaktadır. Bu yapı *android.hardware* paketinin bir parçasıdır ve aşağıda anlatılan sınıflar ile ara yüzden oluşmaktadır.

- SensorManager; Bu sınıfı kullanarak algılayıcıların bir nesnesi oluşturulabilmektedir. Bu sınıf birçok fonksiyon ile beraber, algılayıcıların dinlenmesini başlatabilmekte ve bitirebilmektedir.
- Sensor; Bu sınıf kullanılarak hangi algılayıcı dinlenilmek isteniyorsa o algılayıcının nesnesi oluşturulabilmektedir. Ayrıca algılayıcıların kapasiteleri de bu sınıf aracılığı ile belirlenebilmektedir.
- SensorEvent; Bu sınıf kullanılarak algılayıcı olayları ile ilgili bilgi elde edilebilmektedir.
- SensorEventListener; Bu ara yüz iki metottan bildirim alınmasını sağlamaktadır. Bunlar algılayıcı değeri ve doğruluğu değiştiğinde çağrılmaktadırlar [71].

Lineer ivmeölçer

Lineer ivmeölçer, her bir ekseni temsil edecek şeklide üç boyutlu vektör ile ivme bilgisi sağlamaktadır. Bu algılayıcı ivme bilgisini yerçekimini çıkarılmış halde sunmaktadır. Algılayıcıdan bir nesne oluşturulması aşağıdaki kod parçası ile yapılmaktadır.

```
private SensorManager mSensorManager;
private Sensor mSensor;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);
```

Sonuç olarak; bu algılayıcı takip eden ilişkiye göre çalışmaktadır:
Lineer ivmeölçer = ivmeölçer – yerçekiminden kaynaklanan ivme
Lineer ivmeölçer algılayıcısında cihaz hareketsiz dik konumda tutularak 3 eksende de başlangıç değeri ölçülmektedir. Daha sonra cihaz hareketinde hesaplama yapılırken bu başlangıç değerleri yanlış ölçümlerin önüne geçmek için ölçülen değerlerden çıkarılmaktadır [72].

Algılayıcılar dinlenmeye başladığında, uygulamada bir zamanlayıcı başlatılarak iki resim çekilmesi arası geçen süre hesaplanmaktadır. Başlangıç zamanından bitiş zamanına kadar algılayıcıdan gelen değerin ikinci dereceden integrali alınarak yer değiştirme bilgisine, yani kameranın hareket etme mesafesine ulaşılabilmektedir [73]. Gelen değerlerin belirlenen eşik değerinin pozitif halinin üzerinde ise sağa doğru, eşik değerinin negatifinin altında ise sola doğru hareket ettiği anlaşılmaktadır. Bu uygulamada eşik değeri olarak 0,2 m/s^2 olarak belirlenmiştir.

Farklılık değerinin hesaplanması

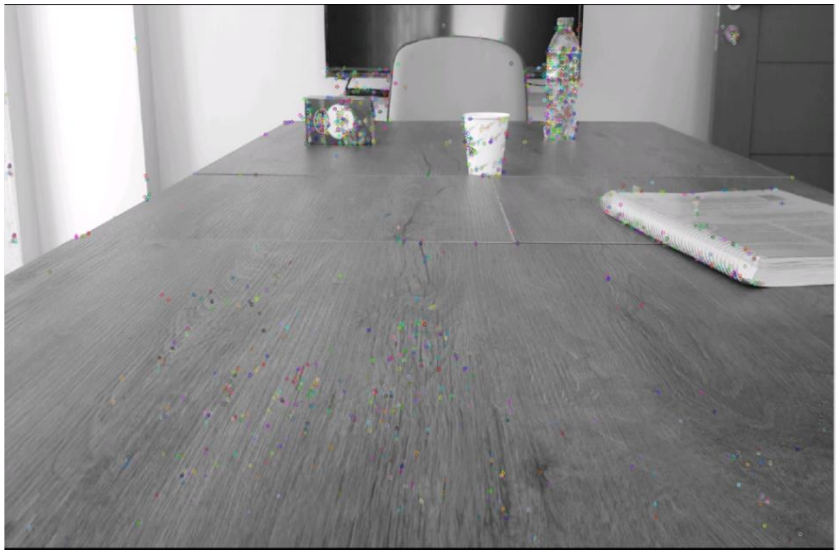
Farklılık değerinin hesaplanabilmesi için çekilen resimlerdeki özellik noktalarının çıkarılıp tanımlanması gerekmektedir. Bu işlemi gerçekleştirebilmek için yeni bir Activity sınıfı oluşturulmuştur. Bu sınıfa, bir önceki sınıfta çekilen resimlerin cihazda kayıtlı bulunduğu yerlerin yol bilgisi, cihazın hareket etme mesafesi ve yönü gönderilmektedir. Bu sınıf içerisinde OpenCV fonksiyonları kullanılmaktadır. Çekilen resimlerin grayscale olarak

okunma işlemi, anahtar noktalarının çıkarılma işlemi ve çıkarılan bu noktaların tanımlama işlemi gerçekleştirilmektedir. Bu noktaların belirlenip tanımlanması için SURF algoritması kullanılmaktadır. Bu sınıfın kodları EK-1'de bulunmaktadır.

Resim 4.1. Sol resim

Resim 4.2. Sağ resim

Resim 4.1 ve Resim 4.2'de örnek bir uygulamada çekilen birinci resim ve kameranın sağa hareketi sonrası çekilen ikinci resim gösterilmektedir. Bu resimler üzerine SURF algoritması uygulandığında Resim 4.3 ve Resim 4.4'de gösterildiği gibi özellik noktaları tespit edilmektedir.



Resim 4.3. Sol resim özellik noktaları

Resim 4.4. Sağ resim özellik noktaları

Örnek olarak; Resim 4.3'deki resimde 9241 Resim 4.4'deki resimde 10498 özellik noktası tespit edilmiştir. Farklılık değerinin tespit edilebilmesi için bu iki resimde aynı yeri gösteren özellik noktalarının doğru bir şekilde eşlenmesi gerekmektedir. Mobil cihazların hesaplama kabiliyetleri düşünüldüğünde bu kadar yüksek sayıda noktanın tespit edilip, tanımlanıp sonrasında eşlenmesi çok zaman alan bir işlem olmaktadır. Bundan dolayı işlem yapma kapasitesi kısıtlı mobil cihazlarda, bu hesaplamaları kabul edilebilir bir zamanda gerçekleştirebilmek için resimlerde bulunması istenen noktaların aranacağı alanın daraltılması gerekmektedir.

Resimlerde noktaların aranacağı alanın kısıtlayabilmek amacıyla uygulama gerçekleştirilirken resimlerin çekilmesi sonrası kullanıcıya ilk çekilen resim gösterilmektedir. Burada kullanıcıdan uzaklığı hesaplanması istenen nesne üzerinde dokunması beklenmektedir. Dokunulan noktanın koordinat bilgisi alınmakta ve bu değer diğer gönderilen bilgilere ek olarak, resimlerin özellik noktalarının bulunacağı sınıfa yollanmaktadır.

Burada resimlerin okunma işlemi gerçekleştirildikten sonra dokunulan koordinat bilgisine göre resimlere kırpılma işlemi uygulanmaktadır. İlk resimde dokunulan yeri merkez alacak şekilde 300 * 300 piksellik kare bir alan kırpılmaktadır.

İkinci resimde, resmin çekilmesi için kamera hareket yönü dikkate alınarak bir kırpılma işlemi gerçekleştirilmektedir. Eğer kamera ilk resim çekilmesi sonrası sağa hareket ettirilip resim çekilmiş ise nesneler ilk resme göre daha solda, eğer kamera sola doğru hareket ettirilmişse nesneler ilk resme göre daha sağda olmaktadır. Bu durum göz önüne alındığında ikinci resim den özellik noktalarının aranması için;

- Kamera sağa doğru hareket etmiş ise, dokunulan noktanın x koordinat değeri ile resmin başlangıcı arası ve yüksekliği 300 piksel olan dikdörtgen alan,
- Kamera sağa doğru hareket etmiş ise, dokunulan noktanın x koordinat değeri ile resmin sonu arası ve yüksekliği 300 piksel olan dikdörtgen alan,

alınarak kırpılmıştır. Bu durum Şekil 4.3 ve Şekil 4.4'de gösterilmiştir. Kırmızı dikdörtgen içinde kalan alanlar kırpılmıştır.

Bu şekilde iki resimde de noktaların aranacağı alan daraltılmakta ve uygulamada hesaplamaların yapılma süresi düşürülmektedir.

a b

Şekil 4.3. Kamera hareket yönü: sağ (a) birinci resim (b) ikinci resim

a b

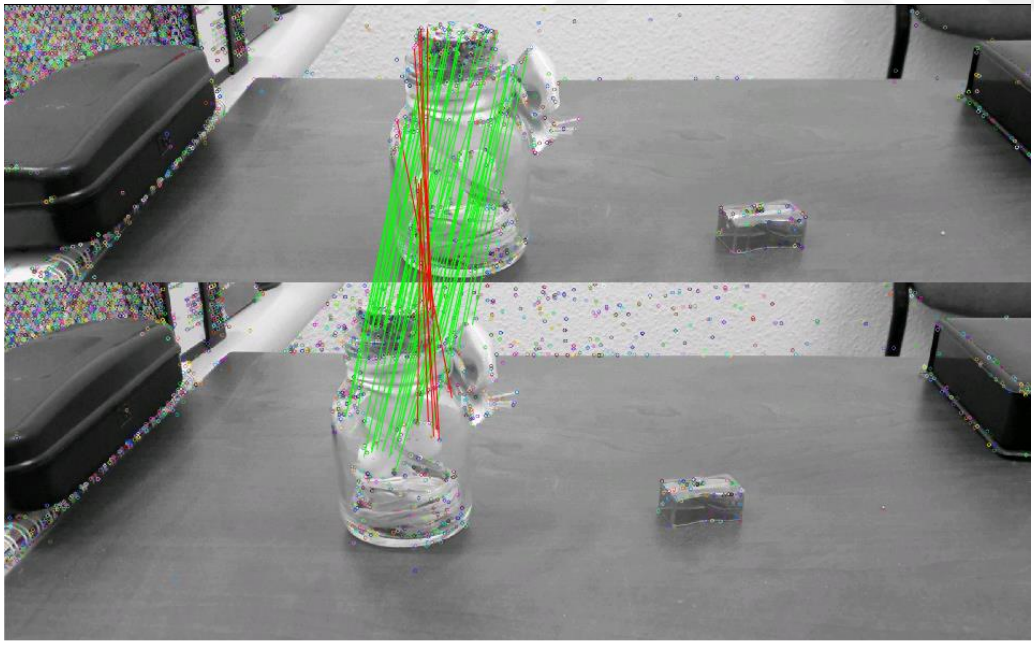
Şekil 4.4. Kamera hareket yönü: sol (a) birinci resim (b) ikinci resim

Noktaların eşlenmesi

İki resimden çıkarılan noktaların tanımlanması sonrası eşleme işlemi gerçekleştirilmektedir. Bu işlemde iki resimde de aynı yeri temsil eden noktalar eşleştirilmektedir. Eşleme işleminde 2.4'de bahsedilen FLANBBASED eşleme kullanılmıştır. Bu işlem sonrası eşleşen noktalar arsında bir değer döndürülmektedir. Bu değer ne kadar küçük olursa, noktalar o kadar iyi eşleşmiş anlamı çıkarılmaktadır.

En iyi eşleşen noktaların çıkarılması için, eşleşme sonrası eşleşen noktaların döndürdükleri değerlerden en küçük olanı yani en iyi eşleşeni bulunmaktadır. Bulunan değerin 2,5 katı arasında kalan noktalar en iyi eşleşen noktalar olarak seçilmektedir. Farklılık değerinin hesaplanması işlemine bu noktalar kullanılarak devam edilmektedir.

Eşleme işlemi sonrası yanlış eşleşen noktaların elenmesi gerekmektedir. Bu eleme işlemi için ilk olarak seçilen en iyi eşleşen noktalar arasında, ikinci resimdeki noktaların X koordinat değerlerinden, birinci resimde eşleştiği noktaların X koordinat değerleri çıkarılmaktadır. Çıkarma sonucu eğer kamera sağa hareket etmişse negatif, sola hareket etmişse pozitif değer gelmesi beklenmektedir. Beklenen değerlerin dışında kalan noktalar yanlış eşleştiklerinden dolayı elenmektedir. Bu durum Şekil 4.5'de görselleştirilmiştir. Burada, iki resim çekilirken kamera sağa hareket ettiğinden ikinci resimde, birinci resme göre nesneler daha solda kalmaktadırlar. Bundan dolayı eşleşen noktalarda, ikinci resimdeki noktaların daha solda ve *X* koordinat değerlerinin daha küçük olması beklenmektedir. Şekil 4.5'de kırmızı ile belirtilen ve ilk resme göre daha sağda kalan yanlış eşleşmelerin elenmesi gerekmektedir.



Şekil 4.5. Sağ ve sol resimlerde eşleşen noktalar

Bu işlemin ardından, kalan noktalar kullanılarak, her bir nokta için geçici olarak uzaklık hesabı gerçekleştirilmektedir. Bu hesaplamalar sonucu elde edilen içinden aykırı değerler kutu-grafiği yöntemi kullanılarak elenmektedir.

Kutu grafiği

Bir seriye ait değerlerin küçükten büyüğe doğru sıralanması sonrası, seriyi değer sayısı yönünden iki eşit kısma ayıran değere medyan, dört eşit kısma ayıran değere kartil (quartiles) adı verilmektedir. İstatistikte, değişim aralığının (bir serideki en büyük değer ile en küçük değer arasındaki fark) serinin iki ucunda yer alan aşırı değerlerden etkilenmemesi için Kartiller Arası Farktan (KAF/ interquartile range/ iqr) yararlanılır. Bir veri grubunun iqr değeri, Üçüncü çeyrek kartil ve birinci çeyrek kartil değerleri arasındaki fark alınarak hesaplanır. İqr, verilerin ortasındaki %50'sinin yer aldığı aralığın genişliğidir. Bu fark, küçükten büyüğe doğru sıralanmış gözlem değerlerinden yarısının ne genişlikte bir kısmını kapladığını belli eder ve değeri ne kadar büyük olursa değişkenliğin o kadar fazla olduğunu, ne kadar az ise orta %50'lik gruptaki değişkenliğin o kadar az olduğunu gösterir [74].

İqr değeri, olasılık dağılımının basit grafik gösterimleri olan kutu grafiklerinin çizilmesinde kullanılır. Kutu Grafiği (Boxplot / Box and Whisker Plot) [75], bir örneklemenin dağılımının grafikle gösterilmesidir. Kutu grafiği, veri yığınının ne kadar simetrik olduğunu, sayıların yayılımının ne olduğunu, diğerlerinden uzakta olan verilerin varlığını, verilerin yoğunluğunun nerede toplandığını, verilerin aralarında boşluklar olup olmadığını ve aykırı gözlem değerleri hakkında bilgi verir [74].

Bununla birlikte dağılımdaki aykırı değerleri tanımlamak için Alt Sınır Değer (ASD) = Q1-1.5 iqr ve Üst Sınır Değer (ÜSD)= Q3 + 1.5 iqr olarak belirlenir. Burada Q1 birinci kartil (%25'lik değer = Q1), Q3 Üçüncü kartil (%75'lik değer = Q3) değerlerini belirtmektedir. Alt ve üst sınır değerleri içerisinde kalmayan veriler aykırı değerler olarak düşünülmekte ve elenmektedir.

Bu eleme işleminden sonra, kalan noktalar üzerinde uzaklığının hesaplanması istenen nesne ile alakası olmayan noktalarında elenmesi gerekmektedir. Bu işlem için ilk olarak noktalar üzerine k-ortalamalar kümeleme algoritması uygulanmaktadır.

K-Ortalamalar algoritması ve en iyi küme seçimi

Hiyerarşik olmayan kümeleme yöntemlerinden en çok kullanılanı k-ortalamalar tekniği, MacQueen tarafından bulunmuş olup, küme sayısının belli olduğu durumlarda birbirine en

yakın değerlere sahip elemanları aynı kümede toplamayı amaçlamaktadır. Bu yöntemde her iterasyonda yeni bir küme merkezi oluşturulmakta ve bir eleman yeniden hesaplanan yeni merkeze daha yakın ise o kümeye taşınmaktadır [76, 77].

K-Ortalamalar tekniği, n birim, p değişken ve k küme için aşağıdaki adımları izleyerek birimleri kümelere ayırmaktadır [76, 77].

a) Elde edilen bilgilere göre ilk k gözlemin her birinin p değişken değerleri birer küme ortalama vektörü olarak kabul edilir. Tüm birimlerin küme ortalamalarına olan uzaklıkları hesaplanır.
b) Geriye kalan n-k gözlemin her biri, ortalaması en yakın olan kümeye atanır ve her atamadan sonra küme ortalamaları genellikle Öklid uzaklığı kullanılarak yeniden hesaplanır.
c) Küme içi varyansın minimum ve kümeler arası varyansın maksimum olduğu kümeleme yapısına ulaşıncaya kadar tüm birimler k kümeye atanmaya devam eder. Yinelemeli yaklaşımla uygun kümeleme sağlanır. Her birimin bu küme ortalama vektörlerine göre değişik aşamalarda değişik kümelerde yer alması sağlanır.
d) Küme içi kovaryans matrisinin minimum olduğu koşul sağlanıncaya ve yakınsama ölçütüne eşit ya da daha küçük varyans farkına ulaşıncaya kadar parçalama işlemine devam edilir [76, 77].

Uygulamada kalan noktalar üzerine k değeri 3 olacak şekilde k-ortalamalar algoritması uygulanmaktadır. Bu algoritma uygulanırken iki kümenin başlangıç küme merkezi rastgele seçilirken kalan kümenin küme merkezi için kullanıcı tarafından bir önceki adımda dokunulan nokta alınmaktadır. Bu şekilde uzaklığı bulunması istenen nesneye ait noktaların daha kaliteli bir şekilde bulunulacağı düşünülmektedir. Algoritmada sonlandığında, uygulama tarafından küme merkezleri kontrol edilmektedir. Eğer dokunulan nokta olarak küme merkezi atanan kümenin küme merkezi değişmemiş ise bu küme en iyi küme olarak alınmaktadır. Eğer bu kümenin başlangıç olarak atanan küme merkezi değişmiş ise küme merkezi dokunulan noktaya en yakın olan küme en iyi küme olarak alınmaktadır.

Seçilen en iyi küme elemanları üzerinden her bir nokta için farklılık değeri hesaplanmaktadır. Bu farklılık değerlerinde son olarak aykırı değerler bulunup elenmektedir. Kalan bu değerlerin ortalaması alınarak istenen nesne için farklılık değerine

ulaşılmış olunmaktadır.

Böylece nesnenin uzaklığının hesaplanabilmesi için gerekli olan bütün parametreler elde edilmiş olunmaktadır. Bu parametreler ve Eş. 2.1'deki eşitlik yardımı ile nesnenin uzaklığı hesaplanmaktadır.

Üç ve dört resim için uygulamanın tekrarlanması

Uygulamanın daha iyi sonuç verip veremeyeceğinin denenmesi için ikiden fazla resim çekilerek tekrarlanmıştır. Bu işlemi gerçekleştirebilmek için uygulama kodlarında küçük değişiklikler yapılmıştır. İlk olarak resimlerin çekilmesi işlemini gerçekleştiren Activity sınıfındaki sayacın alacağı değerler tekrar gözden geçirilmiştir. Sayaç üç resim çekildiğinde üç değerine dört resim çekildiğinde ise dört değerine ulaşacağından bu durumları yönetmek için bu sınıfa önceki bölümde anlatıldığı gibi koşul ifadeleri eklenmiştir.

Uygulamanın bir sonraki adımında uzaklığı hesaplanmak istenen nesnenin seçilmesi için ilk resim gösterilmektedir. Daha sonra, dokunulan yerin koordinat bilgileri ve kameranın hareket etme yönü dikkate alınarak resimlere kırpılma işlemi uygulanmaktadır. İlk resim için kırpılma işlemi önceki bölümde anlatıldığı gibi yapılmaktadır. Diğer resimler için gerçekleştirilecek kırpma işlemi ise bir önceki bölümde ikinci resme uygulanan kırpma işlemi ile aynı olmaktadır. Üç resim için ikinci ve üçüncü, dört resim için ikinci, üçüncü ve dördüncü resimlere bu işlem uygulanmaktadır.

Resimlere daha sonra SURF algoritması ayrı ayrı uygulanarak özellik noktaları bulunmakta ve tanımlanmaktadır. Eşleştirme işlemi ve bu işlemden sonra uygulanan eleme işlemi önceki bölümde anlatıldığı gibi gerçekleştirilmektedir. Eşleştirme işlemine resimler ikişerli olarak tabi tutulmaktadırlar. Örneğin üç resim çekilmiş ise 1. ve 2. resim, 1. ve 3. resim, 2. ve 3. resim eşleştirilmektedir.

Son olarak her bir eşleştirme için elde edilen farklılık değerleri ve resimler çekilirken kameranın hareket etme mesafeleri kullanılarak uzaklık değerleri hesaplanmaktadır. Elde edilen bu sonuçların ortalamaları alınarak nesnenin kameraya olan uzaklık değeri çıkarılmaktadır.

4.2. Deneysel Sonuçlar

Deneylerin gerçekleştirilmesi için farklı nesneler farklı uzaklıklara yerleştirilmiştir. Bu şekilde oluşturulan sahnelerde kameranın hareket etme mesafesi 3cm, 5cm, 10cm ve 15cm olacak şekilde ayarlanmıştır. Sahnede bulunan nesnelerin mesafeleri ayrı ayrı ölçülmüş ve doğruluk karşılaştırılması yapılmıştır.

Ayrıca bir sahne için çekilen resim sayısı artırılarak deneyler tekrarlanmıştır. Son olarak, özellik noktalarının bulunup, tanımlanmasında kullanılan algoritmalar değiştirilerek tekrar çalıştırılmış, performans ve hız karşılaştırılmaları gerçekleştirilmiştir.

Cep Telefonu

Kumanda

Plastik Su Şişesi

Çay Kutusu

Resim 4.5. Ölçümlerde kullanılan nesneler

Birinci deney düzeneği

İlk deney düzeneğinde nesnelerin farklı uzaklıklara yerleştirilmesi sonucu, kamera hareket etme mesafesi değiştirilerek ölçümler gerçekleştirilmiştir. Bu düzenek ile farklı nesnelerin farklı uzaklıkta bulunmasının ve kameranın iki resim çekilirken yaptığı hareket mesafesinin doğruluğa nasıl etki ettiği gözlemlenmiştir. Bu ölçümler yapılırken özellik belirleme ve tanımlama algoritması olarak SURF algoritması kullanılmıştır.

Çizelge 4.1'de dört farklı nesnenin farklı uzaklıklara yerleştirilmesi ile gerçekleştirilen ölçümler verilmiştir. Her bir ölçüm kameranın hareket etme mesafesinin değiştirilmesi ile

tekrarlanmıştır. Deneyler gerçekleştirilirken plastik su şişesi, cep telefonu, televizyon kumandası ile çay kutusu nesneleri kullanılmıştır. Bu nesneler sırası ile 60 cm, 100 cm, 150 cm ve 180 cm uzaklıklara ayrı ayrı yerleştirilerek ölçümler gerçekleştirilmiştir. Her bir ölçüm beş kere tekrarlanış ve ortalaması alınarak Çizelge 4.1' e eklenmiştir.

Çizelge 4.1. Farklı uzaklıkta bulunan nesnelerin mesafe hesaplama sonuçları

Kamera Hareketi	Nesne	ÖLÇÜLEN UZAKLIK (cm)			
		60 cm	100 cm	150 cm	180 cm
3 cm	Su şişesi	64,854	103,998	140,652	154,318
	Telefon	57,605	105,256	159,782	155,256
	Kumanda	63,108	95,865	161,365	169,389
	Çay Kutusu	63,56	93,897	142,568	186,658
5 cm	Su şişesi	62,991	105,336	143,235	175,025
	Telefon	68,652	104,698	145,02	164,652
	Kumanda	65,525	104,652	140,782	185,235
	Çay Kutusu	66,629	107,896	139,524	160,569
10 cm	Su şişesi	62,896	106,563	155,812	187,563
	Telefon	64,989	94,569	156,897	188,019
	Kumanda	64,315	96,896	158,215	186,163
	Çay Kutusu	64,015	96,215	157,896	187,537

Çizelge 4.1. (devam) Farklı uzaklıkta bulunan nesnelerin mesafe hesaplama sonuçları

15 cm	Su şişesi	64,256	103,028	154,036	187,968
	Telefon	56,569	104,563	158,856	189,685
	Kumanda	56,528	106,986	156,365	192,968
	Çay Kutusu	58,192	109,105	154,157	191,152

Çizelge 4.1'de elde edilen ölçüm sonuçları 3 başlık halinde yorumlanmıştır.

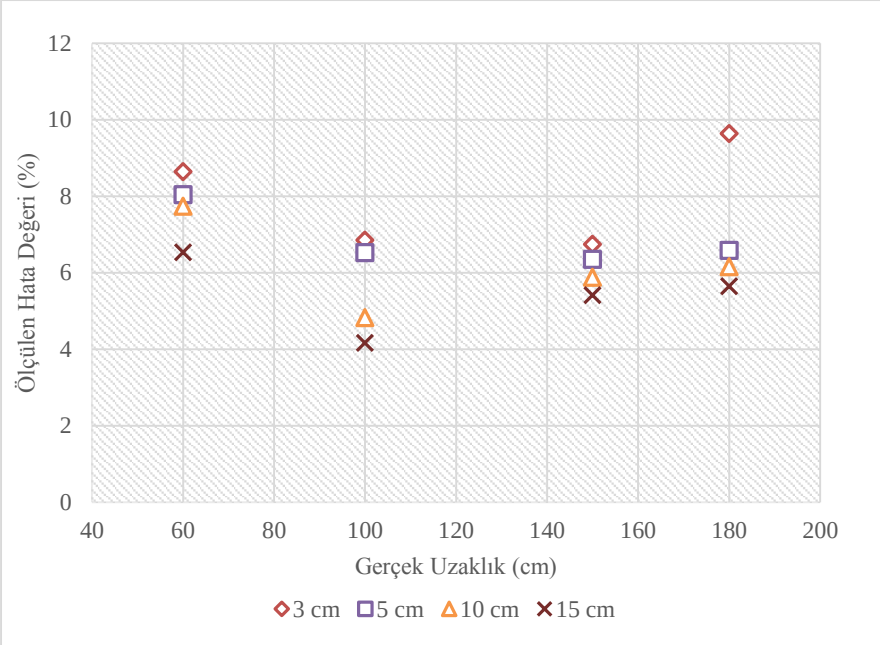
Kamera hareket etme mesafesine göre

Çizelge 4.2'de kamera hareketi 3cm, 5cm, 10cm, 15 cm olduğu durumlarda 60 cm, 100 cm, 150 ve 180 cm uzaklıkta bulunan nesnelerin ölçülen uzaklık değerleri ortalaması verilmiştir. Bu çizelge ile kameranın hareket etme mesafesine göre elde edilen doğruluk değerleri karşılaştırılmıştır. Kamera hareket mesafesinin artmasının ya da azalmasının hesaplama doğruluğuna olan etkisi gözlemlenmiştir.

Çizelge 4.2. Kamera hareket etme mesafesine göre ölçümler

Kamera Hareketi(cm) / Gerçek Uzaklık(cm)	3	5	10	15
60	65,184	64,821	64,641	63,919
100	106,851	106,523	104,821	104,164
150	160,119	159,523	158,821	158,119
180	197,353	191,845	191,075	190,162

Şekil 4.6'de kamera hareketine göre hata oranları bulunmaktadır. Gerçek uzaklığı 60cm, 100cm, 150cm ve 180cm olan nesnelerin kameranın 3cm, 5cm, 10cm, 15cm hareket ettirilmesi ile elde edilen hata oranları verilmiştir. Oranlara göre kameranın her bir farklı hareketinde, hata oranı nesnelerin 100 cm ile 150 cm de bulundukları durumlarda azalmaktadır. Kamera hareket mesafesinin artması ile daha düzgün, az değişken ve daha yüksek oranlarda bir doğruluk elde edildiği gözlemlenmiştir. Kamera hareket etme mesafesinin artması nesnelere ait farklılık değerlerinin de daha yüksek doğruluk ile hesaplanmasını sağlamıştır. Bu durum mesafe hesaplamalarında daha iyi sonuçların alınmasını sağlamıştır.



Şekil 4.6. Kamera hareket ettirilme mesafesine göre hata yüzde değerleri

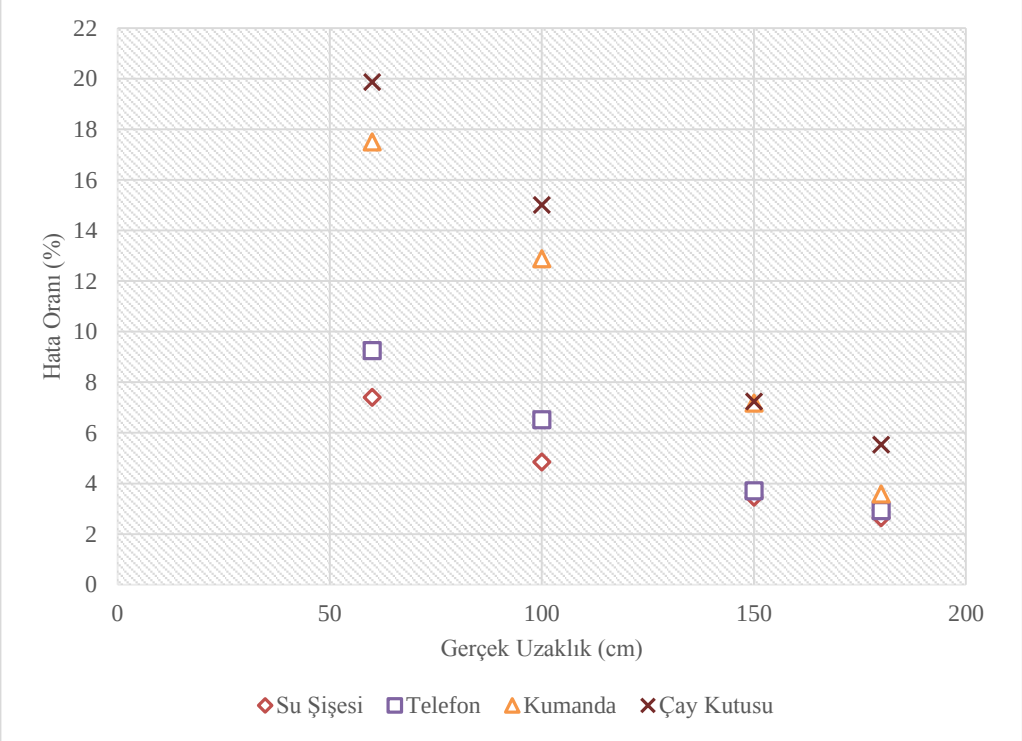
Mesafe ölçümünde kullanılan nesnelere göre

Çizelge 4.3'de ölçümlerde kullanılan dört farklı nesnenin 60cm, 100cm, 150cm ve 180cm uzaklıklarında hesaplanan mesafe değerleri ortalaması verilmiştir. Bu çizelge ile farklı nesnelerin aynı uzaklık değerlerinde farklı uzaklıklarda ölçüldüğü gözlemlenmiştir. Şekil 4.7'de Çizelge 4.3'de ki verilere göre hata oranları hesaplanıp gösterilmiştir. Buna göre plastik su şişesinde en az hata oranı elde edilirken, çay kutusunda ise hata oranı en yüksek çıkmıştır. Ayrıca bütün nesneler için kameraya olan uzaklık mesafesi arttıkça hata oranında azaldığı gözlemlenmiştir. Bu durum resimlerdeki nesnelerden elde edilen özellik noktalarının düzgün bir şekilde eşlenip eşlenmemesi ile alakalı bir durumdur. Plastik su

şişesinde noktaların eşeleme oranı en yüksek ve bu eşleme sonrası elenen nokta sayısı en az olmaktadır. Bu durum bu nesnenin uzaklığının daha yüksek doğruluk ile hesaplanmasını sağlamaktadır.

Çizelge 4.3. Nesnelere göre ölçülen uzaklık değerleri

Nesne / Gerçek Uzaklık(cm)	Su Şişesi	Telefon	Kumanda	Çay Kutusu
60	64,445	65,555	70,5	71,925
100	104,855	106,525	112,887	115,017
150	155,197	155,575	160,775	160,875
180	184,774	185,275	186,45	189,975



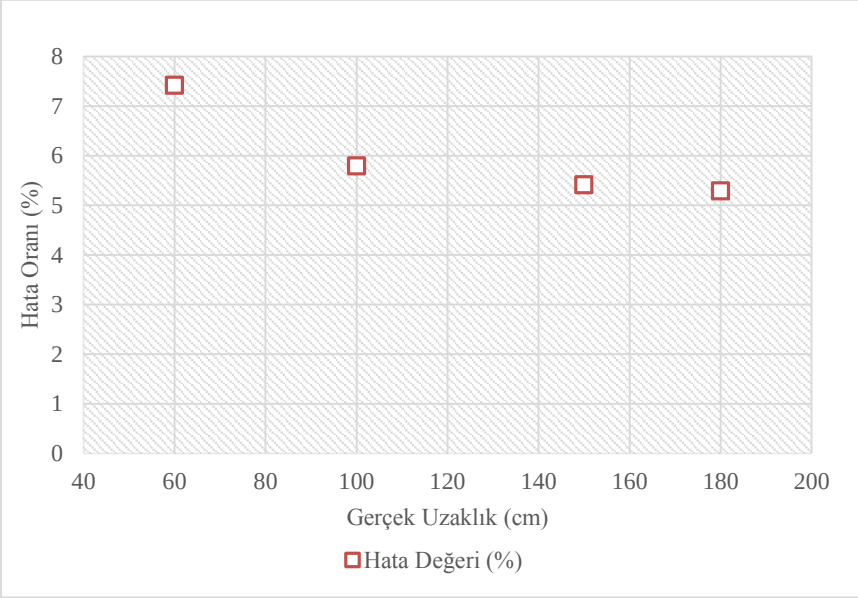
Şekil 4.7. Nesnelere göre uzaklık ölçümü hata oranları

Nesnelerin bulundukları uzaklıklara göre

Çizelge 4.4'de 60cm, 100cm, 150cm ve 180cm uzaklık değerlerinde bulunan bütün nesnelerin elde edilen hesaplama sonuçlarının ortalaması alınarak verilmiştir. Bu çizelge ile nesnelerin kameraya olan uzaklık değerlerinin doğruluğa olan etkisi gözlemlenmiştir. Şekil 4.8'da ise Çizelge 4.4'de bulunan verilere göre hata oranları görselleştirilmiştir. Bu değerlere göre nesnelerin kameraya olan uzaklığı arttıkça daha az hata oranları elde edildiği gözlemlenmiştir.

Çizelge 4.4. Nesnelerin kameraya olan uzaklık ölçüm değerleri

Gerçek Uzaklık (cm)	Ölçülen Uzaklık (cm)	Hata değeri (Yüzde)
60	63,585	7,6417
100	105,556	5,556
150	158,585	5,723
180	189,025	5,0139



Şekil 4.8. Hesaplanan uzaklık mesafelerinde hata değerleri

İkinci deney düzeneği

Bu deney düzeneği ile resimlerdeki özellik noktalarını çıkarıp, tanımlayan algoritmalar değiştirilerek uygulama tekrarlanmıştır. Kameranın hareketi ile çekilen iki resim arasında özniteliklerin çıkartma algoritmalarından elde edilen sonuçların doğruluğu ve bu algoritmaların çalışma süreleri uygulamanın doğruluğunu ve çalışma süresini yansıtacaktır. Yapılan deneyler ile algoritmalardan elde edilen veriler kullanılarak uzaklık hesabı ve doğruluk karşılaştırılması gerçekleştirilmiştir.
Deneylerde SURF, ORB ve BRISK algoritmaları kullanılmıştır. ORB ve BRISK daha yeni ikili (binary) algoritmalardır. Literatürde gerçekleştirilen çalışmalarda bu iki algoritmanın SURF algoritmasına yakın doğrulukta sonuçlar sağlayıp, çok daha hızlı bir şekilde çalıştıkları belirtilmiştir.

Çizelge 4.5'de SURF, ORB ve BRISK algoritmalarının aynı sahnede çekilen resimlere uygulanması sonucu elde edilen değerler görülmektedir. Bu çizelgede aynı sahnenin 5 defa çekilmesi ile noktaların hesaplanması sonucu geçen sürenin milisaniye cinsinden ortalaması ile hesaplama yapılırken kullanılan iki resimde bulunan özellik noktası sayılarının ortalaması verilmiştir. SURF algoritmasının diğer iki algoritmaya göre daha fazla özellik noktası bulduğu gözlemlenmiştir. Ayrıca bu çizelgeye göre ORB algoritması en hızlı SURF algoritması ise en yavaş çalışan algoritmadır. Fakat bulunan özellik nokta sayısı en çok SURF algoritmasında elde edilmiştir.

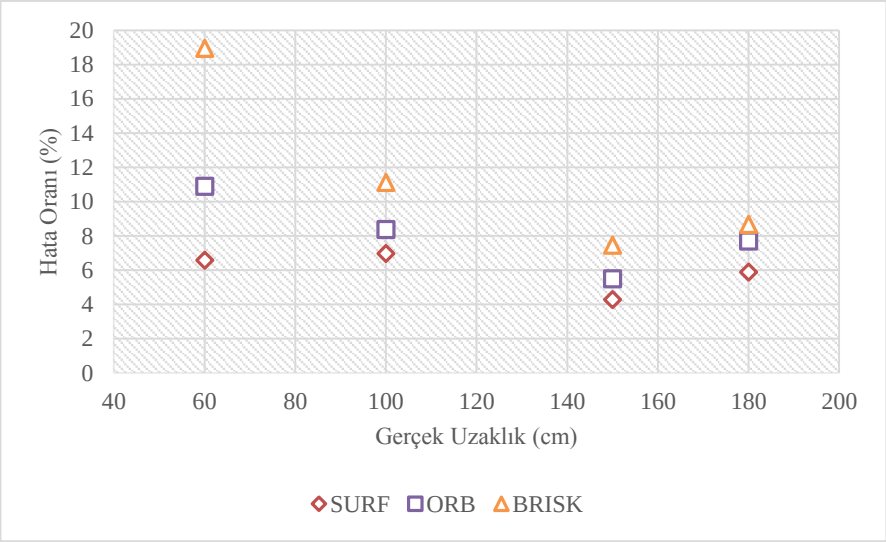
Çizelge 4.6'da ise nesnenin 60cm, 100cm, 150cm ve 180cm uzaklıkta bulunması ile algoritmaların hesapladığı uzaklık değerleri verilmiştir. Bu değerlere göre en iyi başarımın SURF algoritması ile elde edildiği görülmüştür. Tablodaki bilgilere göre SURF algoritması diğer algoritmalara oranla hem daha çok nokta bulmuş bulunan noktalar daha yüksek oranda eşleşmiştir. Bu durum nesneler için farklılık değerinin daha yüksek doğruluk ile hesaplanmasını ve bu algoritmanın diğer algoritmalardan daha iyi sonuç vermesini sağlamıştır. Sonuçlar göz önüne alındığında, her bir algoritma için kabul edilebilir bir doğruluk elde edildiği söylenebilmektedir.

Çizelge 4.5. Uygulamanın farklı algoritmalar kullanılarak çalıştırılması ile bulunan özellik nokta sayısı ve uygulamanın çalışma zamanı

Algoritma	Hesaplama Zamanı (ms)	Resimlerde Bulunan Anahtar Nokta Sayısı		Eşlenen Nokta Sayısı
		Resim 1	Resim 2	
SURF	1869	254	287	49
ORB	193	55	71	12
BRISK	328	33	41	9

Çizelge 4.6. Farklı algoritmaların farklı uzaklıklardaki nesnelerin uzaklıklarını hesaplama sonuçları

Nesne Uzaklığı(cm) / Algoritma	60	100	150	180
SURF	63,95	106,981	156,431	189,598
ORB	66,54	108,37	158,251	193,858
BRISK	71,37	111,126	161,2	195,63



Şekil 4.9. Algoritmalara göre ölçülen uzaklık değerlerinin hata oranları

Üçüncü deney düzeneği

Bu deney düzeneği ile daha doğru sonuçların alınıp alınmayacağının tespit edilebilmesi amacıyla nesnelerin uzaklıklarının hesaplanması için aynı sahnenin üç ve dört resmi çekilerek uygulama gerçekleştirilmiştir. Ölçümler sonucu elde edilen değerler Çizelge 4.7'de gösterilmiştir.

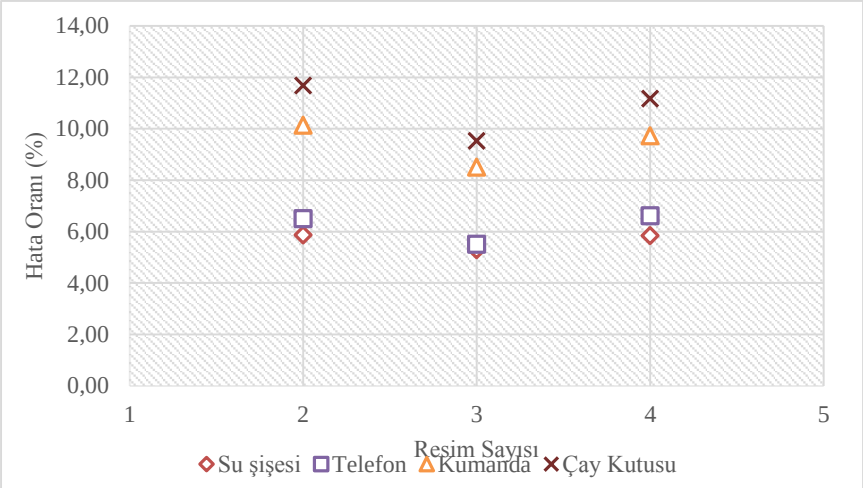
Çizelge 4.7. Çekilen resim sayısı arttırılması sonucu ölçülen uzaklık değerleri

Resim Sayısı	Nesne	UZAKLIK			
		60 cm	100 cm	150 cm	180 cm
2	Su şişesi	63,045	103,152	153,89	185,556
	Telefon	64,889	105,006	154,889	186,128
	Kumanda	72,888	109,889	163,881	187,125
	Çay Kutusu	71,898	117,258	161,658	191,785
3	Su şişesi	64,002	104,996	154,665	184,556
	Telefon	64,008	10,886	155,225	187,012

Çizelge 4.7. (devam) Çekilen resim sayısı arttırılması sonucu ölçülen uzaklık değerleri

	Kumanda	67,558	111,002	158,663	184,563
	Çay Kutusu	68,258	108,236	161,021	188,258
4	Su şişesi	64,589	104,536	156,238	186,259
	Telefon	67,256	106,526	154,896	187,834
	Kumanda	68,032	105,996	158,829	190,256
	Çay Kutusu	70,008	112,332	160,891	189,945

Ölçülen değerler sonucu iki, üç ve dört resim çekilmesi ile elde edilen uzaklık değerlerinin nesne cinslerine göre yorumlanması ile elde edilen hata oranları Şekil 4.10'de verilmiştir. Grafikteki verilere göre uzaklık hesabı için çekilen resim sayılarında üç resim için en iyi başarım elde edilirken, dört resim ile yapılan ölçümlerin iki resim ile yapılan ölçümlere yakın sonuçlar verdiği gözlemlenmiştir. Üç resim çekilmesi ile elde edilen düşük hata oranı resimlerde bulunan anahtar noktalardan elde edilen farklılık değerinin hesaplanması ile ilgili bir durumdur. Resimlerin her biri ayrı ayrı kırpılmakta ve anahtar noktalar bulunmakta ve bulunan anahtar noktalar ikili olarak eşleştirilip ortalaması alınmaktadır. Üç resimde eşleşen noktalar ile hesaplanan farklılık değeri diğerlerine oranla daha yüksek olmakta ve daha iyi sonuçlar vermektedir. Ayrıca su şişesi ve telefon nesneleri için elde edilen sonuçların diğer iki nesneye oranla daha az hata oranına sahip olduğu görülmektedir. Kullanılan nesnelerin neden birbirlerinden farklı doğruluk oranlarına sahip olduğundan birinci deney düzeneğindeki ikinci bölümde bahsedilmektedir.



Şekil 4.10. Ölçümlerde kullanılan nesnelere göre hata oranları

5. SONUÇ VE ÖNERİLER

Stereo görüş sistemlerinde aralarında belirli uzaklık bulunan iki kamera kullanılarak aynı sahnenin farklı açılardan iki resmi çekilmektedir. Tek kameralı stereo görüş sisteminde ise kamera yatayda hareket ettirilerek sahnenin resimleri çekilmekte ve bu resimler üzerinde işlemler gerçekleştirilmektedir. Geleneksel stereo görüş sistemlerinde iki kamera arası uzaklık önceden bilinmekte fakat tek kameralı sistemlerde bu uzaklık değişken olacağından her seferinde hesaplanması gerekmektedir.

Bu çalışma ile mobil bir cihaz kullanılarak tek kameralı stereo görüş sistemi bir sahnede bulunan nesnelerin kameraya olan uzaklıklarının hesaplanabilmesi için oluşturulmuştur. Gerçekleştirilen uygulamada kameranın yataydaki hareketi mobil cihaz içerisindeki algılayıcılar yardımı ile hesaplanmaktadır. Çekilen resimler üzerine özellik noktaların tespiti için SURF (Speeded Up Robust Feature) algoritması uygulanmıştır.

Çalışmanın test edilebilmesi için plastik su şişesi, cep telefonu, çay kutusu ve kumandadan oluşan dört farklı nesnenin bulunduğu bir sahne kullanılmıştır. Nesneler bir sahneye her biri aynı anda 60cm, 100cm, 150cm ve 180cm uzaklıklarımdan birinde bulunacak şekilde sırası ile yerleştirilmiş ve kameraya olan uzaklıkları ölçülmüştür. Ölçümler kameranın yataydaki hareket etme mesafesi değiştirilerek tekrarlanmıştır. Bu şekilde kamera hareketinin, farklı nesnelerin ve nesnelerin bulundukları uzaklık mesafesinin uygulama doğruluğuna etkileri gözlemlenmiştir. Ayrıca, daha hızlı ve doğru sonuçların alınıp alınamayacağının görülebilmesi amacıyla çekilen resimlerdeki anahtar noktaları bulmak için kullanılan algoritma değiştirilerek uygulama tekrarlanmıştır. Son olarak çalışmada, sahnenin iki resmi yerine üç ve dört resmi çekilerek hesaplamalar gerçekleştirilmiş doğruluk karşılaştırılması yapılmıştır.

Birinci deney düzeneği ile elde edilen sonuçlara göre nesne cinsinin, iki resim çekilirken kameranın hareket ettiği mesafenin ve nesnelerin bulundukları uzaklıkların hesaplama doğruluğuna ayrı ayrı etkileri olduğu gözlemlenmiştir. Yapılan deneyler sonucu kullanılan nesneler arasında en yüksek doğruluk plastik su şişesi ile elde edilmiştir. Bu sonuç resimlerdeki nesnelerden elde edilen özellik noktalarının düzgün bir şekilde eşlenip eşlenmemesi ile alakalı bir sonuçtur. Kullanılan algoritma ile plastik su şişesinde diğer

nesnelere oranla daha çok ve benzersiz nokta bulmaktadır. Böylece, plastik su şişesinde noktaların eşeleme oranı en yüksek, bu eşleme sonrası elenen nokta sayısı en az olmaktadır ve daha az hata ile farklılık değeri hesaplanmaktadır. Bu durum bu nesnenin uzaklığının diğer nesnelerden daha yüksek doğruluk ile hesaplanmasını sağlamaktadır. Ayrıca deneyler daha yüksek baz doğrusunun daha iyi sonuçlar ortaya koyduğunu göstermektedir. Elde edilen sonuçlara göre baz uzaklığının artması ile hesaplanan farklılık değerlinin doğruluğunun da arttığını gözlemlenmiştir. Yüksek doğruluk ile hesaplanan bu farklılık değerleri hesaplamalarda daha iyi sonuçların alınmasına olanak sağlamıştır. Ayrıca bu sonuç baz değerinin kısa olduğu durumlarda farklılık değerindeki küçük değişimlerin büyük hata oranlarına neden olduğunu da göstermektedir.

İkinci deney düzeneğinde, uzaklık hesabının yapılması için çekilen resimlerden özellik noktalarının tespit edilip tanımlanması için kullanılan algoritmaların değiştirilmesi ile hız ve doğruluk karşılaştırılması gerçekleştirilmiştir. SURF algoritmasının diğer iki algoritmaya göre çok daya yavaş, ORB algoritmasının ise BRISK algoritmasından daha hızlı çalıştığı gözlemlenmiştir. ORB ve BRISK daha yeni ikili (binary) algoritmalardır. Bu algoritmalar ikili tanımlama temelli olmaları ve ikili eşleştirmelerinin Hamming uzaklık kıstası ile yapılabiliyor olması bu hızı sağlayan en önemli etkenlerdir. Ayrıca SURF algoritması diğer iki algoritmadan çok daha yüksek sayıda anahtar nokta bulmaktadır. Resimlerde bulunan bu anahtar noktalarının eşlenme doğruluğunun SURF algoritmasında daha yüksek olduğu ve bu eşleşmelerde diğer iki algoritmadan daha az oranda nokta elendiği tespit edilmiştir. Bu durum, algoritmalardan elde edilen özellik noktalarının kullanılması ile gerçekleştirilen uzaklık hesaplarında ise SURF algoritmasının diğer iki algoritmadan daha yüksek doğruluk değerini vermesine neden olmaktadır. Elde edilen sonuçlar göz önüne alındığında, eğer bir uygulamanın hızlı çalışması isteniyorsa ORB algoritması, hızdan ziyade daha yüksek bir doğruluk elde edilmesi bekleniyorsa SURF algoritmasının kullanılması önerilmektedir.

Üçüncü deney düzeneği ile nesnelerin için uzaklık hesaplamalarında çekilen resim sayının artması ile bulunan ve karşılaştırılan anahtar noktalarının artmasını sağlamaktadır. Resimlerin her biri ayrı ayrı kırpılmakta ve anahtar noktalar bulunmakta ve bulunan anahtar noktalar ikili olarak eşleştirilip ortalaması alınmaktadır. Üç resimde eşleşen noktalar ile hesaplanan farklılık değeri diğerlerine oranla daha yüksek olmakta ve daha iyi sonuçlar vermektedir. Daha iyi hesaplanan farklılık değeri daha düşük hata oranının alınmasını sağlamıştır.

Gerçekleştirilen tez çalışmasındaki uygulama ile günlük hayatımızdaki mobil cihazlarda çevremizdeki nesnelerin kameraya olan uzaklıkları hesaplanabilmektedir. Çekilen resimlerde gerçekleştirilen kırpılma işlemi ile resimlerde istenilen alan işlenmektedir. Bundan dolayı, yapılan işlemin karmaşıklığı ve mobil cihazların hesaplama yapma kabiliyetleri düşünüldüğünde uygulama hızlı bir şekilde çalışmaktadır. Mesafe hesaplamanın doğruluğu mobil cihaz içerisindeki algılayıcılardan gelen değerlere doğrudan bağlı olduğundan, cihazın sabit bir hızla hareket ettirildiği kabulü ile uygulama kabul edilebilir bir doğruluk ile işlemleri gerçekleştirmektedir.

KAYNAKLAR

1. Bhatia, A., Ghosh, P., and Venkatesh, K. (2013). Multistereo system design. *Image Information Processing (ICIIP),*148-153.

2. Hamzah, R. A., Aziz, K. A. A., and Shokri, A. (2012). A pixel to pixel correspondence and region of interest in stereo vision application. *Computers & Informatics (ISCI),* 193-197.

3. Fan, X., Wang, X., & Xiao, Y. (2014). A shape-based stereo matching algorithm for binocular vision. *Security, Pattern Analysis, and Cybernetics (SPAC)*, 70-74.

4. Chen, Y., Chen, Y., Yuan, Z., and Zhang, Y. (2006). Multi-Stereo Vision Tracking for AR System. *IEEE International Conference on Information Acquisition*, 326-331.

5. Lim, Y.-C., Lee, C.-H., Kwon, S., and Jung, W.-Y. (2008). Distance estimation algorithm for both long and short ranges based on stereo vision system. *Intelligent Vehicles Symposium*, 841-846.

6. Zheng, L.-W., Chang, Y.-H., and Li, Z.-Z. (2010). A study of 3D feature tracking and localization using a stereo vision system. *Computer Symposium (ICS)*, 402-407.

7. Mustafah, Y. M., Noor, R., Hasbi, H., and Azma, A. W. (2012). Stereo vision images processing for real-time object distance and size measurements. *Computer and Communication Engineering (ICCCE)*, 659-663.

8. Song, X., Wu, Y., Yang, L., and Liu, Z. (2013). Object position measuring based on adjustable dual-view camera. *In Multimedia and Expo Workshops (ICMEW)*, 1-6.

9. Yoo, I.-S., and Seo, S.-W. (2015). Object distance estimation based on frequency domain analysis using a stereo camera, 343-344.

10. Ranftl, R., Gehrig, S., Pock, T., and Bischof, H. (2012). Pushing the limits of stereo using variational stereo estimation. *In Intelligent Vehicles Symposium (IV)*, 4, 401-407.

11. Haritaoglu, I., Harwood, D., and Davis, L. S. (1998). W4S: A real-time system for detecting and tracking people in 2 1/2D. *Computer Vision—ECCV'98*, 877-892.

12. Beymer, D., & Konolige, K. (1999). *Real-time tracking of multiple people using stereo*. Proc. of IEEE frame rate workshop.

13. Harville, M. (2004). *Stereo person tracking with adaptive plan-view templates of height and occupancy statistics*. Image and Vision Computing, 22(2), 127-142.

14. Morency, L.-P., Gupta, R. (2003). *Robust real-time egomotion from stereo images*. Image Processing, *ICIP*, 2, 719-722.

15. Holzmann, C., & Hochgatterer, M. (2012). Measuring distance with mobile phones using single-camera stereo vision. *Distributed Computing Systems Workshops (ICDCSW)*, 32, 88-93.

16. Ogura, T., Mizuchi, Y., Kim, Y.-B., and Choi, Y. (2015). Highly accurate stereo-based measuring and tracking system for vessel control. *In Control, Automation and Systems (ICCAS)*, 15, 763-768.

17. Murmu, N., Nandi, D. (2014). Low cost distance estimation system using low resolution single camera and high radius convex mirrors. *Advances in Computing, Communications and Informatics (ICACCI*), 993-1003.

18. Li, W., Zhao, Y., Lu, S., and Chen, D. (2015). Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing. *Communications Magazine*, 53(3), 89-97.

19. Hardware, R.-P. M. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE pervasive Computing*, 8(4), 14-23

20. Hussain, K. M., Zepherin, R. A. R., ShanthaKumar, M., and Abirami, S. (2015). Anaglyph 3Dimesional Image Processing using NI-LabVIEW. *International Journal for Innovative Research in Science and Technology*, 1(8), 108-112.

21. Temiz, M. S. (2011). *Video Görüntülerinden Hareketli Nesnelerin Çikarilmasi Ve Hareket Yörüngelerinin Belirlenmesi*, Yayımlanmamış Doktora Tezi, İstanbul Teknik Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.

22. Bradski, G., and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. (First Edition). Sebastopol: O'Reilly Media, 415.

23. Davis, J., Ramamoorthi, R., and Rusinkiewicz, S. (2003). Spacetime stereo: A unifying framework for depth from triangulation. Computer Vision and Pattern Recognition, 2, 359-366.

24. İnternet: Sarı, Mustafa. 3D Görüntü İşleme. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.mavis.com.tr%2Fblog%2F%3Fcat%3D1&date=2016-08-29, Son Erişim Tarihi: 29.08.2016.

25. Khadka, B., Dhonju, D. L., and Shrestha, (2008). A. Stereo Vision: An Introductory Approach. *Binsankhadka*, 12-16.

26. Özgündüz, E., Karslıgil, M. E., & Şentürk, T. (2009). Stereo vision based real-time obstacle distance and dimension estimation using look-up table. *In2009 IEEE 17th Signal Processing and Communications Applications Conference*, 17, 636-639.

27. Güvendik, C. (2014). *Fpga Based Stereo Camera Depth Map Generation*. Yayımlanmamış Yüksek Lisans Tezi, Dokuz Eylül Fen Bilimleri Enstitüsü, İzmir.

28. Kytö, M., Nuutinen, M., and Oittinen, P. (2011). Method for measuring stereo camera depth accuracy based on stereoscopic vision. *IS&T/SPIE Electronic Imaging*, 786-789.

29. Cornelis, N., Gool, L. V. (2008). Fast scale invariant feature detection and matching on programmable graphics hardware. *Computer Vision and Pattern Recognition Workshops*, 3, 1-8.

30. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.

31. Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. *Computer Vision (ICCV)*, 2548-2555.

32. Gunduz, A. E., Temizel, A., and Taskaya Temizel, T. (2013). Feature detection and tracking for extraction of crowd dynamics. *Signal Processing and Communications Applications Conference (SIU)*, 21, 1-4.

33. Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Computer Vision–ECCV*, 430-443

34. Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. *Computer Vision–ECCV*, 778-792.

35. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: an efficient alternative to SIFT or SURF. *International conference on computer vision*, 2564-2571.

36. Wu, S., Lew, M. (2013). Evaluation of salient point methods. *Proceedings of the 21st ACM international conference on Multimedia*, 21, 685-688

37. Canclini, A., Cesana, M., Redondi, A., Tagliasacchi, M., Ascenso, J., and Cilla, R. (2013). Evaluation of low-complexity visual feature detectors and descriptors. *Digital Signal Processing (DSP)*, 18, 1-7

38. Oral, M. C. G. (2016). SIFT Yöntemini Kullanarak Madeni Para Tanıma. *EEB 2016 Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 11, 1-4.

39. Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, CVPR*, 1(1), 511-518.

40. Brown, M., Lowe, D. G. (2002). Invariant Features from Interest Point Groups. *In BMVC*, 1-10.

41. Yu, L., Yu, Z., and Gong, Y. (2015). An Improved ORB Algorithm of Extracting and Matching Features. International Journal of Signal Processing, *Image Processing and Pattern Recognition*, 8(5), 117-126.

42. Hartmann, J.-M., Klussendorff, J. H., & Maehle, E. (2013). A comparison of feature descriptors for visual SLAM. *Mobile Robots (ECMR)*, 56-61.

43. Özcan, M. O., Sert, E., Taşkın, D., ve Taşkın, C. OpenCV ile Stereo Görüntülerden Derinlik Kestirimi, *ab.org*, 13, 1-4.

44. Wang, Z., Kieu, H., Nguyen, H., and Le, M. (2015). Digital image correlation in experimental mechanics and image registration in computer vision: similarities, differences and complements. *Optics and Lasers in Engineering*, 65, 18-27.

45. Muja, M., Lowe, D. G. (2009). Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP*, (1)2, 331-340.

46. Chermak, L., Aouf, N. (2012). *Enhanced feature detection and matching under extreme illumination conditions with a hdr imaging sensor*. *Cybernetic Intelligent Systems (CIS)*, 11, 64-69.

47. Walcher, H. (2014). *Position sensing: angle and distance measurement for engineers*. (Fifth Edition). Boston: Elsevier, 176.

48. Farhad, M., Farhad, S., and Ahmad, M. (2013). Indoor security system design and implementation using depth information. *Informatics, Electronics & Vision (ICIEV)*, 1-5.

49. Isobe, Y., Masuyama, G., and Umeda, K. (2015). *Target Tracking for a Mobile Robot with a Stereo Camera Considering Illumination Changes*. *Proc. of the 2015 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, 702-707.

50. Ogura, T., Mizuchi, Y., Hagiwara, Y., Kim, Y., Suzuki, A., Choi, Y., and Watanabe, K. (2013). Distance measurement system using stereo camera for automatic ship control. *In Control, Automation and Systems (ICCAS)*, 1020-1024.

51. Comlekciler, I. T., Gunes, S., & Irgin, C. (2016). Artificial 3-D contactless measurement in orthognathic surgery with binocular stereo vision. *Applied Soft Computing*, 41, 505-514.

52. Gu, F., Nakata, T., and Bao, Y. (2010). Arbitrary 3D image generation using a single camera and a spin mirror. *SICE Annual Conference*, 2364-2370.

53. Jaramillo, C., Guo, L., and Xiao, J. (2013). A single-camera omni-stereo vision system for 3D perception of micro aerial vehicles (MAVs). *Industrial Electronics and Applications (ICIEA)*, 8, 1409-1414.

54. Kim, H., Lin, C., Song, J., and Chae, H. (2005). Distance measurement using a single camera with a rotating mirror. *International Journal of Control Automation and Systems*, 3(4), 542.

55. Nishimoto, Y., Shirai, Y. (1987). A feature-based stereo model using small disparities. *Proc. Computer Vision and Pattern Recognition*, 192-196.

56. Shen, Y., Peng, P., and Gao, W. (2012). 3D reconstruction from a single family camera. In *Advanced Computational Intelligence (ICACI), 2012 IEEE Fifth International Conference on*, 108-112.

57. Teoh, W., Zhang, X. (1984). An inexpensive stereoscopic vision system for robots. *Robotics and Automation. Proceedings*, 186-189.

58. Sroba, L., Grman, J. (2015). Determination of camera displacement using image stereo pair. *Computer Science and Information Technologies (CSIT*), 95-99.

59. Boonkwang, S., Saiyod, S. (2015). Distance measurement using 3D stereoscopic technique for robot eyes. *7th International Conference on Information Technology and Electrical Engineering (ICITEE), 232-236.*

60. Jian, L., Chengyan, Z., and Shujuan, C. (2012). Positioning Technology of Apple-Picking Robot Based on OpenCV. *Digital Manufacturing and Automation (ICDMA),* 618-621.

61. Xia, C., Li, Y., Chon, T.-S., and Lee, J.-M. (2009). A stereo vision based method for autonomous spray of pesticides to plant leaves. *Industrial Electronics,* 909-914.

62. Seo, D., Park, H., Jo, K., Eom, K., Yang, S., and Kim, T. (2013). Omnidirectional stereo vision based vehicle detection and distance measurement for driver assistance system. *In Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, 5507-5511.

63. Lai, Y., Huang, Y., and Hwang, C. (2016). Front moving object detection for car collision avoidance applications. *IEEE International Conference on Consumer Electronics (ICCE),* 367-368.

64. Shin, H., Sohn, K. W. (2012). Real-time depth range estimation and its application to mobile stereo camera. *IEEE Consumer Communications and Networking Conference (CCNC),* 5-9.

65. İnan, M., Biçek, E. Restoran Yönetim Süreçlerine Mobil Yaklaşım: Bir Android Uygulama. *Akademik Bilişim Konferansı,* 10 18.

66. İnternet: Google. Android Developers. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com&date=2016-08-29, Son Erişim Tarihi: 29.08.2016.

67. İnternet: OpenCV. OpenCV for Android. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fopencv.org%2Fplatforms%2Fandroid.html&date=2016-08-29, Son Erişim Tarihi: 29.08.2016.

68. Tenekeci M. E., Gümüşçü, A., Baytak, A., ve Aslan, E. (2013). Görüntüden OpenCV ile Duygu Analizi. *Akademik Bilişim Konferansı*, 14, 1-5.

69. Sert, E., Taşkın, D., Taşkın, C., Topçubaşı, N., ve Köprücü, İ. (2012). OpenCV ile Kamera Kalibrasyonu. *Akademik Bilişim Konferansı*, 12, 1-7.

70. İnternet: Android Developers. Camera Characteristics. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Freference%2Fandroid%2Fhardware%2Fcamera2%2FCameraCharacteristics.html&date=2016-08-29, Son Erişim Tarihi: 29.08.2016.

71. İnternet: Android Developers. Sensors Overview. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Fguide%2Ftopics%2Fsensors%2Fsensors_overview.html&date=2016-08-29, Son Erişim Tarihi: 29.08.2016.

72. İnternet: Android Developers. Motion Sensors-Using the Linear Accelerometer. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdeveloper.android.com%2Fguide%2Ftopics%2Fsensors%2Fsensors_motion.html%23sensors-motion-linear&date=2016-08-29, Son Erişim Tarihi: 29.08.2016.

73. Seifert, K., Camacho, O. (2007). Implementing positioning algorithms using accelerometers. *Freescale Semiconductor*, 1-13.

74. Acar, M. (2009). *Heyelanların İzlenmesinde Esnek Hesaplama Yöntemleri*, Doktora Tezi, İstanbul Teknil Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 39-43.

75. Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S., & Kavsek, B. (2000). Informal identification of outliers in medical data. *In Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, 20-24.

76. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8), 651-666.

77. Atalay, A., Tortum, A. (2010). Türkiye'deki İllerin 1997-2006 Yılları Arası Trafik Kazalarına Göre Kümeleme Analizi. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 16(3), 1997-2006.

EKLER

EK-1. Derinlik hesabı

```
File file1 = new File(Environment.getExternalStorageDirectory(), "AutoExperiment2/" +
        imgPath1);
File file2 = new File(Environment.getExternalStorageDirectory(), "AutoExperiment2/" +
        imgPath2);
                Mat image1, image2;

try {
   if (file1.exists() && file2.exists()) {
      // Resimleri Grayscale olarak okuma

      /**
       * 1. çekilen resim image2 içersine alınıyor.
       * 2. çekilen resim image1 içersine alınıyor.
       *
       * */
      image1 = Imgcodecs.imread(file1.getAbsolutePath(),
       Imgcodecs.IMREAD_GRAYSCALE);
      image2 = Imgcodecs.imread(file2.getAbsolutePath(),
       Imgcodecs.IMREAD_GRAYSCALE);


      /**
       * Resimlerde dokunulan koordinata göre kırpma işlemi uyulanıyor
       * İlk resimde dokunulan koordinat çevresi kırpılırken ikinci resimde kameranın
          hareket yönüne göre
       * dokunulan korrdinattan önceki ya da sonraki kısım kesiliyor.
       * Böylece gereksiz yerlere bakılmamış olunup işlemlerin daha hızlı yapılması
          sağlanıyor.
       * */
      Rect roi = new Rect(X, Y, WIDTH, HEIGHT);
      Mat cropped = new Mat(image1, roi);
      Mat outputImage1 = cropped.clone();

      // neTaraf true, hareket sağa
      // neTaraf fase, hareket sola
      int x = 0;

      if (neTaraf) {
         x = X - 400;
         if (x < 0) x = 0;
         roi = new Rect(X - (300 + WIDTH), Y, X, HEIGHT); // WIDHT = X + 300
         roi = new Rect(x, Y, 500, HEIGHT); // WIDHT = X + 300
      } else {
         roi = new Rect(X, Y, 3264 - X, HEIGHT); // WIDHT = 3264 - X
      }
      cropped = new Mat(image2, roi);
      Mat outputImage2 = cropped.clone();
```

EK-1.(devam) Derinlik hesabı

```
FeatureDetector SURF = FeatureDetector.create(FeatureDetector. SURF);

keyPoints = new MatOfKeyPoint();
logokeyPoints = new MatOfKeyPoint();


SURF.detect(outputImage2, keyPoints); // 2. çekilen resim için
SURF.detect(outputImage1, logokeyPoints); // 1. çekilen resim için


Size keyPoint = keyPoints.size();
Size keyPointLogo = logokeyPoints.size();

System.out.println(keyPoint);
System.out.println(keyPointLogo);

DescriptorExtractor SurfExtractor = DescriptorExtractor
      .create(DescriptorExtractor. SURF);



Mat descriptors = new Mat();
Mat logoDescriptors = new Mat();


SurfExtractor.compute(outputImage2, keyPoints, descriptors);
SurfExtractor.compute(outputImage1, logokeyPoints, logoDescriptors);

/**
 * İki resimin karşılaştırma işlemi burada yapılıyor
 *
 * */
gm = new MatOfDMatch();
matches = new MatOfDMatch();
good_matches = new LinkedList<>();

double max_dist = 0;
double min_dist = 1000;

matcher = DescriptorMatcher.create(DescriptorMatcher. FLANBASED);
try {
   matcher.match(descriptors, logoDescriptors, matches);

} catch (Exception e) {
   e.printStackTrace();
}
```

EK-1.(devam) Derinlik hesabı

```
        double x1 = keypoints_sceneList.get(good_matches.get(i).trainIdx).pt.x;
        double x2 = keypoints_objectList.get(good_matches.get(i).queryIdx).pt.x;

        x1 += X;
        x2 += x;
        double fark = x1 - x2;
        if (!neTaraf)
            fark = x2 - x1; // Eğer neTaraf değişkeni false ise sola hareket var, true ise sağa
      hareket var demektir.

        if ((fark) >= 0) {
            ort += (fark);
            farkList.add(fark);// ilk elenen noktalar sonrası disparity değerleri
            count++;
            // objList değerleri ekleniyor

            point = new
      exp1.sensor.oda114.sensorapp6.kmeans.Point(keypoints_objectList.get(good_match
      es.get(i).queryIdx).pt.x + x,
      keypoints_objectList.get(good_matches.get(i).queryIdx).pt.y + Y);
            objKMeans.getPoints().add(point);

            point = new
      exp1.sensor.oda114.sensorapp6.kmeans.Point(keypoints_sceneList.get(good_matche
      s.get(i).trainIdx).pt.x + X,
      keypoints_sceneList.get(good_matches.get(i).trainIdx).pt.y + Y);
            sceneKMeans.getPoints().add(point);
        }
      }
      sceneKMeans.init(X + 150, Y + 150);
      sceneKMeans.calculate();
int baseline = Baseline;
      output = (0.34 * baseline) / enYakınNoktaDisparity * 10000;
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : MUMCU, Ali
Uyruğu : T.C.
Doğum yeri : TRABZON
Medeni hali : Bekâr
Telefon : 0 (312) 582 31 14
e-mail : ali.mumcu@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi /Bilgisayar Mühendisliği	Devam Ediyor
Lisans	Anadolu Üniversitesi /Bilgisayar Mühendisliği	2013
Lise	Affan Kitapçıoğlu Lisesi	2006

İş Deneyimi

Yıl	Yer	Görev
2013-Halen	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

GAZİ GELECEKTİR..