

**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**(YÜKSEK LİSANS TEZİ)**

**SÜRÜ ZEKÂSI YAKLAŞIMI  
İLE PANORAMİK GÖRÜNTÜ  
OLUŞTURMA**

**Fatih DAĞDEVİREN**

**Tez Danışmanı: Doç. Dr. Muhammed Gökhan CİNSDİKİCİ**

**Uluslararası Bilgisayar Anabilim Dalı**

**Sunuş Tarihi: 14.08.2015**

**Bornova-İZMİR**

**2015**

Fatih DAĞDEVİREN tarafından yüksek lisans tezi olarak sunulan “Sürü Zekâsı Yaklaşımı ile Panoramik Görüntü Oluşturma” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 14 Ağustos 2015 tarihinde yapılan tez savunma sınavında aday oybirliği ile başarılı bulunmuştur.

**Jüri Üyeleri:**

**İmza**

<b>Jüri Başkanı</b>	: Doç.Dr.Muhammed Cinsdikici	.....
<b>Raportör Üye</b>	: Doç. Dr. Geylani Kardeş	.....
<b>Üye</b>	: Doç. Dr. Bahadır Karasulu	.....

## EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

### ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi / Doktora Tezi olarak sunduğum “Sürü Zekâsı Yaklaşımı ile Panoramik Görüntü Oluşturma” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

.... / .... / 20..

İmzası

Adı-Soyadı

**ÖZET****SÜRÜ ZEKÂSİ YAKLAŞIMI KULLANARAK PANORAMİK  
GÖRÜNTÜ OLUŞTURMA**

DAĞDEVİREN, Fatih

Yüksek Lisans Tezi, Uluslararası Bilgisayar Anabilim Dalı

Tez Danışmanı: Doç. Dr. Muhammed Gökhan CİNSDİKİCİ

Ağustos 2015, 54 sayfa

Görüntü alabilen cihazlar yardımıyla, içinde bulunulan ortamın tamamını görüntülemek amacıyla önce parça parça resimler çekilmektedir. Ardından çekilen resimler birleştirilerek Panoramik görüntü elde edilmektedir. Parçaların birbirlerine birleştirilmeleri için kullanılan algoritmalar ne kadar hassas ortak belirleyebiliyorsa, kalitesi o derece yüksek sonuç görüntüleri elde edilmiş olmaktadır.

Panoramik görüntü oluşturmak için ilk olarak çakıştırılacak resimlerin ortak noktalarını bulmak gerekmektedir. Literatürdeki çeşitli önemli nokta tespit algoritmaları ile bu işlem gerçekleştirilmektedir. İkinci adımda olarak tespit edilen önemli noktaların özniteliklerinin oluşturulması ve oluşturulan özniteliklerin eşlenmesi gerçekleştirilmektedir. Son adımda ise yanlış eşleşen öznitelikler elenmekte ve kalan öznitelikler kullanılarak resimler birleştirilmektedir.

Bu tez çalışmasında literatüre katkı sağlayacak yeni bir öznitelik eşleme yöntemi üzerinde durulmaktadır. Tespit edilen önemli noktalar, Sürü Zekâsı yaklaşımlarından Yapay Arı Kolonisi algoritması kullanılarak doğru bir biçimde eşleştirilmektedir. Yanlış eşleşen noktaların elenmesi içinse RANSAC algoritması kullanılmaktadır. Son olarak kalan noktalar üzerinden birleştirme işlemi yapılmaktadır.

**Anahtar sözcükler:** Panoramik görüntü, Sürü Zekâsı Yaklaşımı, Önemli nokta Tespit ve Eşleme Algoritmaları

**ABSTRACT**

**PANORAMIC IMAGE CREATION BY USING PARTICLE SWARM  
OPTIMIZATION**

DAĞDEVİREN, Fatih

MSc in International Computer Department  
Supervisor: Assoc. Prof. Dr. Muhammed Gökhan CİNSDİKİCİ  
August 2015, 54 pages

By using devices which has capability of taking photo, pieces of photos are taken for displaying whole scene. After, photos are stitched to obtain panoramic view of that scene. The algorithms which find common points more sensitive by using photos which is a part of same scene gives high quality pictures.

Finding common points is the first step of creating panoramic image. This is done by using some algorithms in literature. The second step is to create features around every common point and match them. Last step is eliminating outlier points which can be classified as wrong matched points and stitch images by using remaining points.

In this thesis study, as a contribution to the literature a new feature matching method is proposed. Common points between images are matched by using Artificial Bee Colony Algorithm which is member of Particle Swarm Optimization. After matching points, RANSAC algorithm is used to eliminate false matches. At last stitching is done by using remaining points.

**Keywords:** Panoramic view, Particle Swarm Optimization, Key Point Detection and Matching

## TEŞEKKÜR

Yüksek lisans eğitimim süresince ve tez çalışmamda benden desteğini ve ilgisini esirgemeyen, bilgi ve deneyimleriyle yol gösteren tez danışmanım Sayın Doç. Dr. Muhammed Gökhan Cinsdikici'ye sabrından, özverisinden ve cesaretlendirmesinden dolayı çok teşekkür ederim. Onun değerli fikirleri ve yardımları sayesinde bu tez ortaya çıkmıştır. Uluslararası Bilgisayar Enstitüsü Müdürü Sayın Prof. Dr. Mehmet Emin Dalkılıç ve öğretim üyelerine, yüksek lisansım süresince bana destek olan UNIPA A.Ş.'deki çalışma arkadaşlarıma, hayatımın her döneminde benden desteğini esirgemeyen aileme sonsuz teşekkürlerimi sunarım.

**İÇİNDEKİLER**

	<u>Sayfa</u>
ÖZET .....	vii
ABSTRACT .....	ix
TEŞEKKÜR .....	xi
ŞEKİLLER DİZİNİ .....	xv
ÇİZELGELER DİZİNİ .....	xvii
KISALTMALAR DİZİNİ .....	xix
1. GİRİŞ .....	1
2. LİTERATÜR ÖZETİ .....	3
3. TEMEL BİLGİLER .....	11
3.1 Önemli Nokta Tespit Yöntemleri .....	11
3.1.1 Harris köşe nokta tespit algoritması .....	11
3.1.2 SIFT önemli nokta tespit algoritması .....	14
3.2 Öznitelik Tanımlama Yöntemleri .....	16
3.2.1 Basit öznitelik tanımlayıcısı .....	16
3.2.2 SIFT öznitelik tanımlayıcısı .....	17
3.2.3 Yerel ikili örüntü öznitelik tanımlayıcısı .....	18
3.3 Öznitelik Eşleme Yöntemleri .....	19
3.3.1 Öklid uzaklığı kullanarak eşleme .....	20

## İÇİNDEKİLER (devam)

### Sayfa

3.3.2 Öklid ve uzaklık oranı kullanarak eşleme .....	20
3.3.3 Normalizasyon uygulanmış çapraz ilişkilendirme metodu.....	20
3.4 Aykırı nokta tespit yöntemleri .....	21
3.4.1 RANSAC algoritması .....	21
4. TEZ YÖNTEMİ.....	23
4.1 Noktaların Tespiti .....	24
4.2 Öznitelik Oluşumu Ve Eşlenmesi.....	29
4.3 Aykırı Noktaların Elenmesi.....	33
5. TARTIŞMA VE DENEYSEL SONUÇLAR .....	35
6. SONUÇ.....	47
KAYNAKLAR DİZİNİ.....	49
ÖZGEÇMİŞ.....	53
EKLER .....	..



## ŞEKİLLER DİZİNİ

<u>Şekil</u> .....	<u>Sayfa</u>
3.1 Piksel etrafındaki yoğunluk değişimlerinin gruplanması .....	11
3.2 Öz değerlere göre alanların sınıflandırılması .....	13
3.3 Ölçeksel uzay ve Dog piramidi elde edilmesi .....	15
3.4 Uç noktaların hesaplanması .....	15
3.5 Öznitelik oluşturulması.....	16
3.6 16x16'lık alanların 4x4'lük alanlara bölünmesi .....	18
3.7 Yerel ikili örüntü operatörleri.....	19
3.8 Yerel ikili örüntü ikilik tabanı dizisi oluşumu ve onluk gösterimi.....	19
4.1 Sürü Zekâsı ile resim birleştirme akış diyagramı .....	31
4.2 (A) Referans Resim-1 (B) Karşılaştırılacak Resim-1 .....	33
4.3 Sürü Zekâsı yaklaşımı ile noktaların eşlenmesi-1 .....	33
4.4 Ransac uygulanmamış resim .....	34
4.5 Ransac algoritması ile aykırı noktaların elenmesi sonrası.....	34

## ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
5.1 Seri 1'e ait başarılı birleştirme sayıları grafiği.....	36
5.2 Seri 2'ye ait başarılı birleştirme sayıları grafiği.....	37
5.3 (A) Referans Resim-2 (B) Karşılaştırılacak Resim-2 .....	38
5.4 Özniteliklerin sürü zekâsı yaklaşımı ile eşlenmesi .....	38
5.5 Ransac Sonucu-2.....	39
5.6 Birleşme sonucu-2.....	39
5.7 (A) Referans Resim-3 (B) Karşılaştırılacak Resim-3 .....	40
5.8 Pso ile noktaların Eşleşmesi-3 .....	40
5.9 Ransac algoritması sonrası-3 .....	40
5.10 Birleşme sonucu-3.....	40
5.11 (A) Referans Resim-4 (B) Karşılaştırılacak Resim-4 .....	41
5.12 Pso ile noktaların eşlenmesi-4.....	41
5.13 Ransac algoritması sonrası-4 .....	41
5.14 Birleşme sonucu-4.....	42
5.15 (A) Referans Resim-5 (B) Karşılaştırılacak Resim-5 .....	42

## ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
5.16 Pso ile noktaların eşleşmesi-5.....	43
5.17 RANSAC algoritması sonrası-5 .....	43
5.18 Birleşme sonucu-5 .....	43



**ÇİZELGELER DİZİNİ**

<u>Çizelge</u>	<u>Sayfa</u>
4.1 Serilere ait bilgiler .....	24
4.2 Seri 1'e ait farklı baskılama alanı ve eşik değerleri sonucu ortaya çıkan nokta sayısı .....	25
4.3 Seri 2'ye ait farklı baskılama alanı ve eşik değerleri sonucu ortaya çıkan nokta sayısı .....	25
4.4 Seri 5'e ait farklı baskılama alanı ve eşik değerleri sonucu ortaya çıkan nokta sayısı .....	25
4.5 Seri 1'e ait SIFT algoritmasının nokta tespit ölçüsü .....	28
4.6 Seri 2'e ait SIFT algoritmasının nokta tespit ölçüsü .....	28
4.7 Seri 5'e ait SIFT algoritmasının nokta tespit ölçüsü .....	28
5.1 Seri 1'e ait birleştirme sonuçları.....	35
5.2 Seri 2'ye ait birleştirme sonuçları.....	36
5.3 Performans sonuçları ve algoritmik karmaşıklıklar.....	44



**KISALTMALAR DİZİNİ**Kısaltmalar

NCC	Normalized Cross Correlation
PSO	Particle Swarm Optimization
SIFT	Scale-Invariant Feature Transform
RANSAC	Random Sample Consensus
RR	Referans Resim
KR	Karşılaştırılacak Resim
SURF	Speeded Up Robust Features
ORB	Oriented Fast and Rotated BRIEF
FAST	Features From Accelerated Segment
BRIEF	Binary Robust Independent Elementary Features





## 1.GİRİŞ

Teknolojinin gelişimi ile birlikte, birçok yenilik insanların kullanımına sunulmuştur. İnternet kullanımının yaygınlaşması sosyal platformların kullanımını ortaya çıkarmıştır. Sosyal platformlarda insanların paylaşım yapmaları çeşitli teknolojik aletlerin kullanımını sıklaştırmıştır. Bu teknolojik aletlerin en yaygın olarak tercih edileni fotoğraf çekme özelliği bulunan makinalardır. Sosyal platformlarda insanların içinde buldukları mekânın tamamının paylaşılması istenmesi üzerine panoramik görüntü oluşturma ihtiyacını doğurmuştur. Bu ihtiyacın giderilmesi için çeşitli çalışmalar yapılmış ve panoramik görüntü oluşturmak için farklı yöntemler ortaya sunulmuştur.

Panoramik görüntü kadrage sığmayan bir görüntünün yatayda yan yana dikeyde üst üste olacak şekilde alınan birden fazla görüntüyü birleştirmek için yapılan çalışmadır. Panoramik görüntü oluşturmak için yapılan çalışmalarda, resimler 2 farklı yöntem kullanılarak birleştirilmektedirler. Bu yöntemlerden ilki kalıp eşleme kullanarak panoramik görüntü oluşturmaktır. Bu yöntemde referans resimdeki bir resim kalıbı karşılaştırılacak olan resimde aranır ve resim birleştirme işlemi eşlenen alanlar üzerinden gerçekleştirilmektedir. İkinci olarak kullanılan yöntem, resimler üzerinde çeşitli algoritmalar uygulanarak bulunan önemli noktalar üzerinden birleştirme şeklindedir. Bu yöntemde tespit edilen her bir önemli noktaya ait öznitelikler oluşturulmaktadır. Eşleme işlemi öznitelikler üzerinden gerçekleştirilmektedir. Bu yöntemde yanlış eşleşmelerin elenmesi de yapılmaktadır. Son olarak ise kalan noktalar üzerinden resimler birleştirilip panoramik görüntü oluşturulmaktadır.

Bu tez çalışmasında sunduğumuz yöntem, referans resim üzerinde önemli noktaların, karşılaştırılacak olan resim üzerindeki kenar noktaların tespit edilmesinden sonra bu noktaların Sürü Zekâsı yaklaşımlarından Yapay Arı Kolonisi algoritması kullanılarak eşlenmesinin sağlanmasıdır. Eşlenen noktalar arasından yanlış eşleşmelerin elenmesi için RANSAC algoritması kullanılmaktadır. Son olarak kalan noktalar üzerinden resimler birleştirilip panoramik görüntü elde edilmektedir.

Çalışmanın ikinci bölümünde literatürde daha önce yapılan benzer çalışmalar kronolojik sırayla anlatılmaktadır. Bu çalışmalarda kullanılan yöntemler ve elde edilen sonuçlara yer verilmektedir.

Üçüncü bölümde ise tez çalışmasında kullanılan yöntemlere ait temel bilgilere yer verilmektedir.

Dördüncü bölümde tez çalışmasında sunulan özgün yöntem ayrıntılı olarak ele alınmaktadır. Öncelikle yöntemin ilk adımı olan önemli noktaların tespiti anlatılmaktadır. Ardından noktaların karakteristik yapısını belirten özneliklerin oluşturulması ve eşlenmesi anlatılmaktadır. Yöntemin son adımındaysa yanlış eşleşen noktaların elenmesi ve eşyazımlılık matrisi anlatılmaktadır.

Beşinci bölümde, tez çalışmasından elde edilen sonuçlara yer verilmektedir.

Son bölümde, önerilen yöntemin geliştirilebilmesi için gelecekte yapılabilecek çalışmalar yanı sıra mevcut durumu ve olumlu edinimleri hakkında bilgilere de yer verilmektedir.

## 2.LİTERATÜR ÖZETİ

Panoramik görüntü oluşturma'nın amacı, birbirinden farklı resimlerin ortak noktaları üzerinden birleştirip tek tek bir resim elde etmektir. Bu doğrultuda ilk olarak referans alan, nokta veya noktaların ve karşılaştırılacak resmin belirlenmesi gerekir. Literatürde bu işlem için çeşitli sayıda yöntem kullanılmaktadır. Kalıp eşleme ilk yaklaşım olarak kabul edilir. Kalıp eşleme yönteminde karşılaştırılacak olan resim kalıbı, karşılaştırılması yapılacak olan resim üzerinde aranır. Her bir arama sonucunda elde edilen değerler kıyaslanıp en uygun değer ve alan belirlenip eşleme yapılır. Karşılaştırılacak resim üzerinde köşe noktalarının tespiti bir diğer yaklaşımdır. (Deepak Parajuli et al, 2012), çalışmasında önemli noktaların tespiti için Harris köşe tespit algoritmasından yararlanmıştır. Bu şekilde elde edilen noktaların karşılaştırmasını yapıp birbirlerine yakın olan noktaları eşleştirmiştir. (Aathreya S. Bhat et al., 2013), çalışmasında önemli noktaların tespiti için SIFT min-max metodunu kullanıp elde edilen noktaları ise SIFT öznitelik tanımlayıcısı ile eşleştirmiştir.

Bu yöntemde önemli noktaların tespiti için Harris ve SIFT algoritmaları kullanılmaktadır. Elde edilen her bir köşe noktası için SIFT algoritmasının temeli olan histogram dizileri oluşturulmaktadır. Karşılaştırılacak noktaların belirlenmesi aşamasında Sürü Zekâsı yaklaşımlarından Yapay Arı Kolonisi algoritmasından yararlanılmaktadır. Yanlış eşleşmeleri ortadan kaldırmak içinse RANSAC algoritması kullanılmaktadır.

Jignesh N Sarvaiya ve arkadaşları (2009), kalıp eşleştirme (İng. *Template Matching*) yöntemini kullanarak resim birleştirme (İng. *Image Stitching*) üzerine bir çalışma ortaya koymaktadır. Bu yöntemde kalıp (İng. *Template*) ile karşılaştırılacak olan alanların belirlenmesi klasik yöntem sayılabilecek bir yöntemle belirlenmiştir. Bu yöntemde kalıp resim üzerinde sol en üstten başlayarak her bir piksel etrafında oluşturulan alanla karşılaştırılmıştır. Karşılaştırılmanın yapılması için normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) algoritması kullanılmış ve kalıp ile en yakın değere sahip alan eşleştirilmiştir. Çalışmadaki deneysel sonuçlara bakarak resim üzerindeki noktaların etrafında alanların belirlenmesi ve karşılaştırması hesaplama açısından uzun sürebilmesinin bir dezavantaj olmasının yanına normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) algoritmasının kalıp eşleştirme için ideal bir algoritma olabileceği bir avantaj olarak belirtilmiştir.

Xiaoxiang Liu (2009), resimlerde kalıp (İng. *template*) eşleştirmek için sürü zekâsı yaklaşımını kullanarak yeni bir yöntem ele almaktadır. Bu yöntem herhangi bir görüntü kalıbının resimdeki eşleniğinin bulunmasının daha hızlı bir biçimde gerçekleşecek şekilde çözüm sunmaktadır. Xiaoxiang Liu, karşılaştırılması yapılacak resim üzerindeki noktaların belirlenmesi için Sürü Zekâsı yaklaşımını algoritmasından (İng. *Particle Swarm Optimization: PSO*) yararlanmaktadır. Bu şekilde Jignesh'in ortaya koyduğu çalışmasındaki yöntem gibi teker teker karşılaştırmak yerine resim üzerine rastgele dağıtılan noktaların Sürü Zekâsı yaklaşımındaki sürünün hareketlerini taklit ederek ziyaret ettiği kalıplar üzerinden normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) karşılaştırma metodunu kullanarak en iyi kaynağa yönelim sağlanmıştır. Deneysel sonuçlarda kalıp eşleştiricisinin sonuçlarının tatmin edici olduğunu gözlemlenmekte ve PSO algoritmasının bu yöntemdeki başarısı yüksek derecede verimlilik olarak tespit edilmesine karşın PSO parametrelerinin gerekli seviyede belirlenmemesi durumunda eşleşme işleminin başarısız olduğu tespit edilmiştir.

Lu Xuan ve arkadaşları (2009) ortaya koydukları çalışmada, önemli noktaların tespitinde SURF algoritmasından yararlanmaktadır. Önemli noktaların eşlenmesi için ilk olarak referans resimdeki her bir noktanın karşılaştırılacak resimdeki diğer noktalarla arasındaki Öklid uzaklığı (İng. *Euclidean Distance*) hesaplanmaktadır. Bu yöntemde bir sonraki adımda referans resimdeki her bir nokta için karşılaştırılacak resimdeki en yakın iki değer oranına bakılmaktadır. Sonuç olarak elde edilen oran eşik değerinden düşükse referans resimdeki nokta ile değer olarak kendisine en yakın nokta eşleştirilir. Eşleştirme sonucunda yanlış eşleşen noktaların elenmesi için RANSAC algoritmasından yararlanılmaktadır. RANSAC algoritması sonucunda elde edilen noktalar ve eşleri kullanılarak resimler birleştirilip panoramik görüntü elde edilmektedir. Bu yöntemde çalışmalarda uygulanan basamakların başarılı sonuçlar verdiği ve SURF algoritmasının SIFT algoritmasına göre daha hızlı çalıştığı belirtilmektedir. Fakat bu çalışmada resimlerdeki önemli noktaların tespitinde ışıktan kaynaklı değişimlerdeki tespit etme oranının SURF algoritmasında SIFT algoritmasına göre daha düşük olduğu belirtilmektedir.

Lei Yang ve arkadaşlarının (2011), "Feature based Image Mosaic Algorithm" adlı çalışmasında üç farklı önemli nokta tespit algoritmasını karşılaştırarak resimleri birleştirme işlemi gerçekleştirmiştir. İlk önemli nokta tespit algoritması olarak Harris kullanılmaktadır. Harris algoritmasının diğer

algoritmalarla göre daha basit hesaplamalar gerektirdiđi tespit edilmiř fakat resimler üzerindeki rotasyon deđiřimlerinden etkilendiđi ve aynı ilgi noktalarını bulmada pasif kaldıđı gözlenmiřtir. İlgi noktası tespitinde ikinci bir algoritma olarak SIFT minimum-maksimum algoritmasından yararlanılmıřtır. Gözlemler sonucunda SIFT algoritmasının resimler üzerindeki deđiřimlerden etkilenmediđi ve bunun yanında SIFT algoritmasının diđer algoritmalarla göre daha yavař çalıştıđı gözlemlenmiřtir. Son bir metot olarak SURF algoritmasından yararlanılmıřtır. SURF algoritmasının SIFT algoritmasından daha hızlı olduđu yapılan gözlemler sonucu ortaya çıkmıřtır. Fakat SURF algoritması resimler üzerindeki ışık kaynađından kaynaklanan deđiřimlerde zayıf olduđu tespit edilmiřtir. Elde edilen ilgi noktaların eřleřtirilmesi için en yakın komřu algoritmasından yararlanılmıřtır ve elde edilen yanlış eřleřmeler RANSAC ile elenmiř bu sayede noktalar üzerinden resimler birleřtirilmiřtir.

Deepak Parajuli (2012) çalışmasında, resimler üzerinde tespit edilen noktaların eřleřtirilmesi için bir yöntem sunmaktadır. Bu yöntemde köře noktalarının tespiti için Harris köře tespit algoritmasından faydalanılmıřtır. Deepak çalışılmasında, Harris algoritmasını kullanarak resimler üzerindeki önemli noktaların tespitinde rotasyon ve öteleme (İng. *translation*) deđiřmezliklerini sađlamaktadır. Bu çalışmada bir sonraki adımda histogram eřitleme (İng. *Histogram equilization*) metodu kullanılarak resimlerin üzerindeki ışıktan kaynaklanabilecek deđiřimlerin ortadan kaldırılması sađlanılmıřtır. Son olarak noktaların etrafında belirlenen büyüklükte alanlar oluşturularak basit bir öznitelik tanımlayıcı oluşturulmuřtur. Oluřturulan özniteliklerin karřılařtırılması için Öklid uzaklıđı formülü kullanılmakta ve en yakın deđere sahip noktalar eřleřtirilmektedir. Fakat yapılan eřleřmeler sonucunda Öklid uzaklıđını kullanarak bu yöntemde öznitelik tanımlayıcısının ölçek deđiřmezliđinin sađlanamamıř olduđunun farkına varılmıřtır.

Cansın Yıldız (2012), çalışmasında resimler üzerindeki önemli noktaların tespiti için SIFT önemli nokta tespit algoritmasından faydalanılmaktadır. Elde edilen noktaların eřleniklerinin bulunması için SIFT öznitelik tanımlayıcısı kullanılmaktadır. Bu yöntemde eřleřmeler sonucunda yanlış eřlenen noktalara RANSAC uygulamakta ve yanlış eřleřmeler elenmektedir. Uygulanan yöntemler sonucunda karřılařtırılan 2 resmin birleřtirilmesi iřleminin verimli bir şekilde gerçekeřtirilmesine karřın SIFT yöntemlerinin diđer yöntemlere göre daha yavař çalıştıđı gözlemlenmiřtir.

Aathreya S. Bhat ve arkadaşlarının (2013) yaptıkları çalışmada, kalıp eşleştirme (İng. *template matching*) yönteminin tek başına uygulanmasının dezavantajlarından bahsedilmektedir. Çalışmada, önemli noktaların çıkarılmasına dayalı bir metodun daha iyi performans çıkaracağı üzerinde durulmaktadır. Bu önemli noktalarının çıkarılması amacıyla, Cansın Yıldız'ın çalışmasında ortaya konan SIFT metoduna benzer bir uygulama ile min-max yaklaşımı kullanılmaktadır. Elde edilmiş önemli noktaların eşleştirilmesi için ise SIFT öznelik tanımlayıcısından (İng. *descriptor*) yararlanılmaktadır. Başarılı sonuçlar elde edilmesine karşın, bu çalışmada, uygulanan metodun yavaş olduğu gözlenmekte ve yanlış eşleşen noktalara ilişkin herhangi bir performans bildirimi yapılmamaktadır.

Russol Abdelfatah ve arkadaşlarının (2013) çalışmasında, panoramik görüntü oluşturmak için ilk olarak önemli noktaların tespitinde Harris algoritması kullanılmaktadır. İki resimdeki noktalar normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) tanımlayıcısı ile kıyaslatılarak maksimum eşleşme değerine sahip noktaları eşleştirilmektedir. Bu çalışmada, normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) tanımlayıcısı SIFT tanımlayıcısı kadar etkili olmadığı ve yanlış eşleşmelerin daha çok görülebileceği bir tanımlayıcı olduğu belirtilmektedir. Bundan dolayı bu yöntemde RANSAC algoritması kullanılarak yanlış eşleşmeler ortadan kaldırılmaktadır. Yapılan çalışmalar sonucunda bu yöntemin başarılı bir şekilde eşleştirme yaptığı görülmüştür.

Niu Jing ve arkadaşlarının (2013), çalışmasında görüntü birleştirme yönteminin 3 basamağı olduğunu ifade etmiştir. Bu basamaklardan ilki olan görüntü ön işleme (İng. *Image preprocessing*) basamağında görüntüler üzerindeki bozulmaların giderilmesi yönünde çalışmalar yapılmıştır. İkinci adımda ise resim üzerindeki önemli noktaların bulunması ve eşleştirilmesi üzerinde çalışmıştır. İlgili noktalarının tespiti için ilk olarak görüntüler üzerinde çakışık bölgelerin (İng. *Overlapped region*) tespiti yapılmaktadır. Bu şekilde ilgi noktalarının hesaplanması sırasında bu bölgelerden yararlanılmakta ve bu sayede yapılacak olan hesaplamaların sayısının azaldığı gözlemlenmiştir. Çakışan bölgeler üzerinden önemli noktaları tespitinde SURF algoritması kullanılmaktadır. Bu yöntemde SURF algoritmasının SIFT algoritmasına göre daha az hesaplama yaptığı savunulmakta ve yöntemin bu sayede ilgi noktalarını daha hızlı tespit ettiği sonucu savunulmaktadır. İlgili noktalarının eşleştirilmesi için Lu Xuan'ın çalışmasında kullandığı Öklid uzaklıklarının (İng. *Euclidean distance*) oranından

yararlanılmaktadır. Son adım olarak yanlış eşleşmelerin elenmesi için RANSAC algoritması kullanılmaktadır ve kalan noktalar üzerinden resimler birleştirilmektedir.

Hemlata Joshi (2013), çalışmasında Harris ve SIFT önemli nokta saptayıcıları kullanılmaktadır. Çalışmalar sonucunda Harris algoritmasının hesaplamalarının SIFT algoritmasına göre daha basit olduğu fakat SIFT algoritmasının resimler üzerindeki değişimlere daha dayanıklı olduğu tespit belirtilmektedir. Tespit edilen ilgi noktalar en yakın komşu algoritması ile eşleştirilmiştir ve yanlış eşleşmeler RANSAC ile elenmiştir. Son adım olarak elde edilen noktalar üzerinden resimler birleştirilmiştir. Bu çalışmada SIFT önemli noktası saptayıcısı ile yapılan birleştirme işlemlerinin daha iyi olduğu belirtilmektedir.

Chen Kaili ve arkadaşları (2014), “Image Stitching algorithm Research Based on OpenCV” adlı çalışmalarında Harris önemli nokta tespit metoduna dayalı bir yöntem sunmuştur. Bu yöntemde köşe noktaları Harris ile tespit edilip eşleştirilmeleri normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) tanımlayıcısı kullanılarak yapılmıştır. Normalizasyon uygulanmış çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) tanımlayıcısı kullanımı resimlerdeki ışıktan kaynaklı değişimleri ortadan kaldırmıştır. Elde edilen noktalar RANSAC algoritması sayesinde süzgeçten geçirilip yanlış eşleşmeler ortadan kaldırılmıştır. Son kalan noktalar üzerinden resimler birleştirilmektedir.

Manisha D. Wasnik (2014), çalışmasında nokta tespitinde ve eşleştirilmesi için SIFT min-max önemli nokta tespit algoritması ve SIFT tanımlayıcısı kullanılmaktadır. Bu yöntemde SIFT min-max algoritmasının karmaşık ve işlem çokluğundan bahsedilmektedir. Bu karmaşıklık ve işlem çokluğunu ortadan kaldırmak için bu çalışmada basitleştirilmiş SIFT önemli nokta tespit algoritması kullanılmaktadır. Elde edilen noktalar SIFT öznelik tanımlayıcısı kullanılarak eşleştirilmektedir. Eşleştirilen noktalar üzerinden iki resim birleştirilmektedir. Bu çalışmada basitleştirilmiş SIFT algoritmasının hızlı ve efektif olduğu sonucuna varılmasına karşın yanlış eşleşen noktaların elenmesi için herhangi bir işlem yapıldığı belirtilmemektedir.

Wai Chong ve arkadaşları (2014), çalışmalarında üç farklı önemli nokta saptayıcısını karşılaştırarak resimlerde birleştirme işlemi uygulamaktadır.

Çalışmada kullanılan resimler yağmurlu havalarda çekilmiştir ve bu şekilde Harris, SURF ve SIFT önemli nokta saptayıcılarının çekilen resimler üzerindeki önemli noktaların tespit sayıları üzerinden bir çalışma gerçekleştirmiştir. Bu çalışmalar sonucunda Harris algoritması diğer algoritmalara göre daha hızlı ve basit, SURF algoritmasının SIFT algoritmasına göre daha hızlı olduğu tespit edilmiştir fakat elde edilen resimler yağmurlu havalarda tedarik edildiğinden dolayı Harris ve SURF algoritmasının ilgi noktası saptama derecelerinde düşüş, SIFT algoritmasındaysa saptanan noktaların sayısında bir değişiklik gözlenmemiştir. Her üç metottan elde edilen ilgi noktaları SIFT öznitelik tanımlayıcısıyla eşlenmektedir. Yanlış eşleşmeler RANSAC ile elenmektedir. Bu yöntemde sonuç olarak SIFT önemli nokta saptayıcısının diğer saptayıcılara göre daha yavaş ve işlem gerektiren bir gereksinim duymasına karşın resimler üzerindeki bozulmalardan diğer saptayıcılara göre daha az etkilendiği görülmüştür.

Rupali Chandratre ve arkadaşları (2014), yaptıkları çalışmada önemli noktaların tespitinde SIFT algoritmasından daha hızlı bir algoritma olan Harris algoritması kullanılmaktadır. Elde edilen önemli noktalar arasında ilişkilendirme (İng. *Correlation*) tespit edilip noktalar eşleştirilmektedir. Chen Kaili ve arkadaşlarının yönteminde elde edilen önemli noktalar için kullanılan ilişkilendirme tanımlayıcısında noktaların etrafında oluşturulan bölgeler normalize edilmektedir. Fakat bu yöntemde kullanılan ilişkilendirme tanımlayıcısında buna benzer bir işlem yapılmamakta ve bundan dolayı bu tanımlayıcı ışığa karşı dayanıklı sonuçlar verememektedir. Çalışmada, elde edilen yanlış eşlenmiş noktaların elenmesi için RANSAC algoritması kullanılmaktadır. Bu yöntemde yapılan deneyler sonucunda Harris ile RANSAC algoritmalarının birlikte çalıştıkları zaman daha iyi performans gösterdikleri belirlenmiştir.

Ashwini ve arkadaşları (2014), yaptıkları çalışmada önemli noktaların tespiti için birden fazla algoritmadan yararlanmaktadır. Yapılan çalışmalar sonucunda resimlerde rotasyon ve ölçekten kaynaklanan değişimlerin gözlenmesi durumunda SIFT yoksa Harris algoritmasından yararlanılması sonucuna varılmıştır. Önemli noktaların eşleştirilmesinde SIFT öznitelik tanımlayıcısı (İng. *descriptor*) kullanılmaktadır. Yanlış eşleşmelerin elenmesi için popüler bir algoritma olan RANSAC algoritması kullanılmaktadır. Elde edilen noktalar üzerinden ise resimler birleştirilmektedir.



Ebstam Adel ve arkadaşlarının (2014), “Image Stitching based on Feature Extraction: A Survey” isimli çalışmasında resim birleştirme işlemi SIFT, SURF ve ORB algoritmaları kullanılarak üç farklı yoldan elde edilmiştir. Aykırı noktaların elenmesi RANSAC algoritması ile gerçekleştirilmektedir. RANSAC uygulandıktan sonra kalan noktalar ile resimler birleştirilmektedir. Ebstam Adel ve arkadaşları bu çalışmada sonuç olarak SIFT algoritmasının rotasyon ve ölçek değişmezliğini sağladığını, ayrıca resimdeki gürültüden kaynaklı bozulmalara daha dayanıklı olduğunu fakat SIFT algoritmasının çalışma süresinin yüksek olduğunu belirtmektedir. SIFT algoritmasına alternatif olan SURF algoritmasının ise daha hızlı çalıştığı belirtilmektedir. Son olarak ORB algoritmasının SIFT gibi rotasyon ve ölçek değişmezliklerini sağladığı, bunun yanında hız bakımından SIFT algoritmasından daha hızlı olduğu fakat resimdeki gürültüden kaynaklı bozulmalara daha dayanıksız olduğu tespit edilmiştir.

Mahammadi Nigar ve arkadaşlarının (2012) ortaya koydukları çalışmada köşe noktası tespiti için FAST algoritması kullanılmaktadır. FAST algoritması ile tespit edilen noktalar BBF metodu ile eşlenmektedir. Bu yöntemde aykırı verilerin elenmesi hakkında bir bilgi verilmemektedir. Çalışmanın sonucunda FAST köşe tespit algoritmasının Harris, Susan ve SIFT algoritmalarına göre daha hızlı çalıştığı tespit edilmiştir. Ayrıca FAST algoritmasının video işleme konularında diğer algoritmalara göre daha fazla performans göstereceği belirtilmektedir.



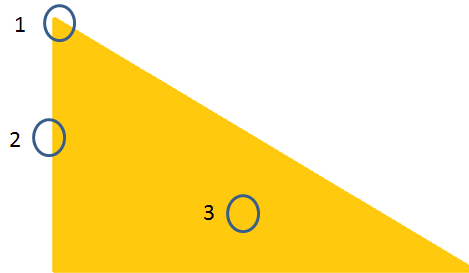
### 3.TEMEL BİLGİLER

Bu bölümde literatürde adı geçen önemli noktaların belirlenmesinde Harris önemli nokta saptama algoritması, SIFT önemli nokta saptayıcısından bahsedilmektedir. Önemli noktaların tespitinden sonraki adım olan öznelik oluşturma işleminde Basit öznelik tanımlayıcısı, SIFT öznelik tanımlayıcısı, Yerel İkili Örüntü tanımlayıcısı detaylı olarak anlatılmaktadır. Öznelikleri kullanıp özneliklerin birbirlerine olan yakınlıklarını ölçen Öklid uzaklığı (İng. *Euclidean Distance*), Öklid ve uzaklık oranı formülü (İng. *Euclidean with Distance Ratio*), Normalize edilmiş çapraz ilişkilendirme (İng. *Normalized Cross Correlation*) yöntemleri ele alınmaktadır. Aykırı noktaların bulunması için RANSAC algoritması kullanılmaktadır. Bu bölümde yukarıda bahsedilen algoritmalar hakkında temel bilgiler ayrıntılı olarak açıklanmaktadır.

#### 3.1 Önemli Nokta Tespit Yöntemleri

##### 3.1.1 Harris köşe nokta tespit algoritması

Harris algoritması köşe noktalarının tespitinde kullanılan algoritmalarından biridir. Bu yöntemde bir pikselin köşe (İng. *Corner*) noktası olup olmadığını belirlemek için piksel etrafındaki yoğunluk değişimlerine bakılmaktadır (Harris and Stephens, 1988).



Şekil 3.1 Piksel etrafındaki yoğunluk değişimlerinin gruplanması

Şekil 3.1 de 3 farklı yoğunluk grubu gözükmemektedir. 1 numaralı gruba ait alanlar köşe (İng. *Corner*) noktası olarak kabul edilmektedir. Bu bölgelerdeki yoğunluk değişimleri her yönde diğer gruplara göre daha fazla olduğu gözlemlenmektedir. 2 numaralı gruba ait alanlar kenar (İng. *Edge*) noktalarıdır. Bu gruba ait alanlar üzerindeki yoğunluk değişimleri 1 numaralı gruba kıyasla daha az miktardadır ve tek yönlüdür. 3 numaralı gruba ait alanlarda yoğunluk

değişimleri çok düşük olarak gözlemlenmektedir. Harris algoritmasında Şekil 3.1 de belirtilen alanların tespiti için resimlerin Gauss ile bulanıklaştırılmasından sonra elde edilen  $x$  ve  $y$  türevlerinden (İng. *Gradient*) yararlanılmaktadır (Abdelfatah and Haitham, 2013). Sonuç olarak Harris algoritmasının formülü Denklem 3.1'deki (Harris and Stephens, 1988) gibi ortaya çıkmıştır.

$$E(u, v) = \sum_{x,y} W(x, y) [I(x+u, y+v) - I(x, y)]^2 \quad (3.1)$$

$I(u+x, v+y)$  Taylor serileri kullanılarak genişletildiğinde Denklem 3.2'de belirtilen formül ortaya çıkmaktadır.

$$E(u, v) \approx \sum_{x,y} [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad (3.2)$$

Denklem 3.2'de belirtilen  $I_x$  ve  $I_y$  resme ait  $x$  ve  $y$  türevleridir (İng. *Gradient*). Denklem 3.2 Taylor serileri kullanılarak genişletildiğinde Denklem 3.3'te belirtilen eşitsizlik ortaya çıkmaktadır.

$$E(u, v) \approx \sum_{x,y} u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2$$

$$E(u, v) \approx [u \ v] \left( \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.3)$$

Denklem 3.3' te çıkan sonuç matris formuna çevrilirse aşağıdaki sonuç ortaya çıkmaktadır.

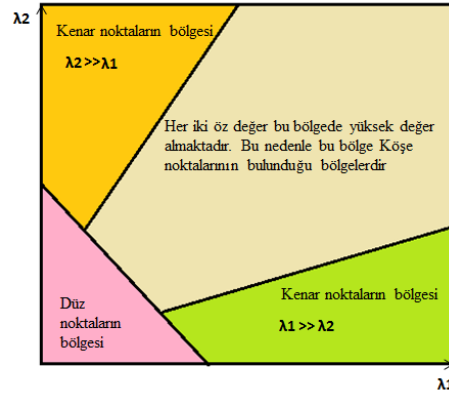
$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3.4)$$

Denklem 3.3'te belirtilen eşitsizliğin son hali:

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.5)$$

Denklem 3.5'te bulunan M matrisinin öz değerleri (İng. *Eigen Value*) hesaplanarak 3 farklı durum elde edilmektedir.

1. Her iki öz değer (İng. *Eigen Value*) küçük olması durumunda düz alan elde edilmektedir
2. Bir öz değer büyük diğeri küçük olması durumunda kenar (İng. *Edge*) elde edilmektedir.
3. Her iki öz değer büyük olması durumunda köşe elde edilmektedir.



Şekil 3.2 öz değerlere göre alanların sınıflandırılması

Matrislerde öz değer hesaplamaları zor olacağı göz önünde bulundurularak aşağıdaki formül türetilmiştir (Başak, 2015).

$$R = \det(M) - k(\text{trace}(M))^2 \quad (3.6)$$

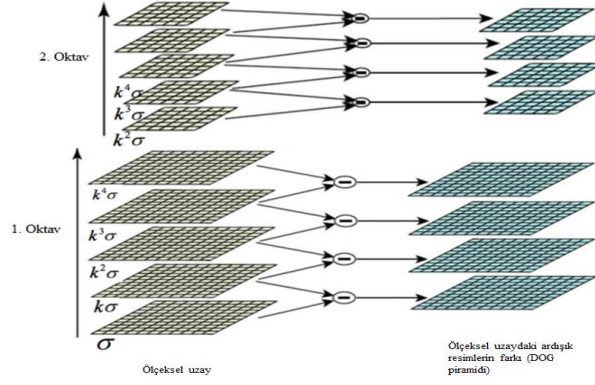
Denklem 3.6'da belirtilen  $\det(M)$  ifadesi M matrisinin determinantının kısaltmasıdır.  $\text{Trace}(M)$  ise resmin türevlerinden gelen piksel değerlerinin toplamıdır. "k" değeri Harris sabiti olarak kabul edilir ve değeri 0,4'tür (Harris and Stephens, 1988). R fonksiyonu resimdeki alanların sınıflandırılmasında büyük pay sahibidir. R fonksiyonu sonucu değeri yüksek olan alanlar köşe noktası adayları olarak belirlenmektedir.

Aday noktalar üzerinde son adım olarak maksimum olmayan noktaları baskılama (İng. *Non-Maximum Supression*) uygulanır ve kalan noktalar köşe noktaları olarak belirlenir. Kenar noktalarını baskılama işleminde noktalar etrafında belirli alanlar oluşturulur ve bu alanların içinde başka aday nokta bulunup bulunmadığı gözlenir. Bulunduğu durumda en yüksek değere sahip aday nokta köşe noktası olarak belirlenmektedir aksi takdirde alanın içinde tek bir aday nokta bulunması durumunda aday nokta köşe noktası olarak işaretlenir (Abdelfatah and Haitham, 2013).

### 3.1.1 SIFT önemli nokta tespit algoritması

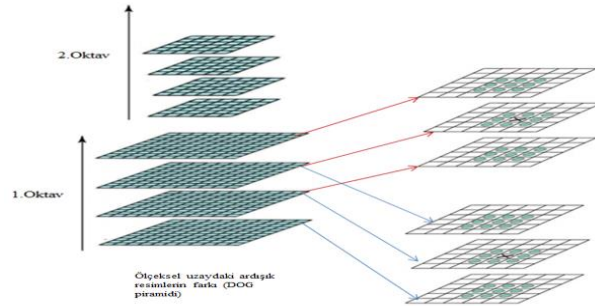
SIFT önemli nokta tespit algoritması ilk olarak 1999 yılında Lowe tarafından sunulmuştur. SIFT algoritması farklı resimlerdeki aynı objelerin tespitinde kullanılan bir algoritmadır (Hemlata, 2013). SIFT önemli nokta tespit algoritması panoramik resim elde etmek için kullanılan yaygın algoritmalarından biridir. SIFT algoritmasının tercih edilmesinin nedeni farklı resimlerde bulunan aynı noktaların resim ölçeklendirmesinden kaynaklanan değişimlere karşı dayanıklı olmasındandır (Ashwini, 2014).

SIFT algoritmasında noktaların tespiti için ilk olarak ölçeksel uzay (İng. *Scale space*) oluşturulmaktadır (Lowe, 2003). Ölçeksel uzayda aynı resmin farklı büyüklükteki halleri kademeli olarak farklı derecelerde Gauss bulanıklaştırma (İng. *Gaussian Blur*) yöntemi ile bulanıklaştırılmaktadır. Lowe tarafından önerilen ölçeksel uzay 3 oktav (İng. *Octave*) içermektedir. Oktav bir resmin belirli bir ölçekte kademeli olarak Gauss ile bulanıklaştırılmış türevlerini içermektedir. Bulanıklaştırma için kullanılan Gauss sigma değerleri her resim için  $k\sigma$  olarak belirlenmektedir. (Utkarsh, 2015), yönteminde  $\sigma$  değerini 1.6,  $k$  değerini ise  $\sqrt{2}$  olarak belirlemiştir. Farklı kademelerdeki bulanıklaştırma miktarları,  $k$  değerinin farklı kuvvetleri alınarak ortaya çıkmaktadır. Her bir oktav için 5 resim önerilmektedir. Aynı oktavdaki resimler bir önceki oktavdaki resimlerin genişlik ve yükseklik olarak yarısına eşittir.



Şekil 3.3 Ölçeksel uzay ve DOG piramidi elde edilmesi (Utkarsh, 2015)

Şekil 3.3'te ölçeksel uzay ve ölçeksel uzayın belirlenmesinden sonraki adım olan DOG piramidinin oluşturulması gösterilmektedir. DOG piramidi, ölçeksel uzayda aynı oktav içinde yer alan ardışık resimlerin farklarını yeni bir ölçeksel uzayda toplanmasıyla meydana gelmektedir (Utkarsh, 2015). Bir sonraki adım, DOG piramidinden her bir oktav için ardışık olarak alınan üç resim üzerinde uç noktaların bulunmasıdır.



Şekil 3.4 Uç noktaların hesaplanması (Utkarsh, 2015)

Şekil 3.4'te DOG piramidi kullanılarak uç noktaların bulunması gösterilmiştir. Her bir oktav içindeki resimler ardışık üçerli gruplara ayrılır. Gruplarda ikinci sırayı alan resimler üzerinde tarama işlemi başlatılır. Tarama işlemi için seçilen piksel etrafına ve komşu resimlerdeki koordinat olarak aynı noktaların etrafına 3x3 büyüklüğünde alanlar oluşturulur. Seçilen pikselin oluşturulan bölgelerin içindeki pikseller arasında en büyük ya da en küçük değere sahip olma durumu o pikselin uç noktası olmasını belirtmektedir (Chia et al., 2014). Şekil 3.4'te "X" ile belirtilen nokta seçilen piksel ve yeşil noktalar ise komşu piksellerdir. Eğer "X" pikseli 27 nokta arasından en büyük ya da en küçük değere sahip olma durumunda uç nokta olarak işaretlenmektedir.

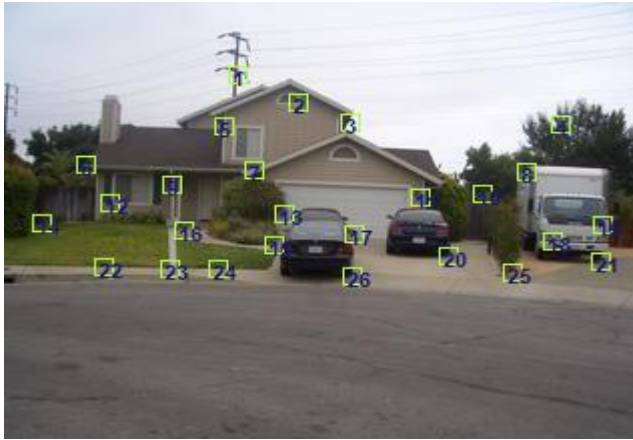
SIFT önemli nokta tespit algoritmasında, uç noktalar arasından düşük kontrast değerine sahip noktalar ve kenar noktalar elenmesi son adım olarak belirlenmiştir. Düşük kontrast değerine sahip noktalar, DOG fonksiyonu Taylor serisinin ikinci dereceden açılımı ile birlikte kullanılarak elenmektedir (Özgen vd., 2015). İkinci aşamada kenar noktaların tespiti yapılmaktadır. Kenar noktaların tespiti için  $2 \times 2$ 'lik Hessian Matrisinden ( $[D_{xx}, D_{xy}; D_{xy}, D_{yy}]$ ) faydalanılmaktadır. Geri dönüş değerinin ( $(D_{xx}+D_{yy})^2/(D_{xx}D_{yy}-(D_{xy})^2)$ ) Lowe tarafından önerilen "10" değerinden büyük olduğu noktalar elenmektedir. Bu işlemler sonucunda geriye kalan noktalar önemli nokta olarak belirlenmektedir (Utkarsh, 2015).

### 3.2 Öznitelik Tanımlama Yöntemleri

Resim birleştirme işlemlerinde öznitelikler (İng. *descriptor*), önemli noktaların tespitinden sonra iki resimdeki benzerliğin hesaplanması için oluşturulan ve noktaların karakteristik yapısını belirleyen yapılardır (Müezzinoğlu vd., 2014).

#### 3.2.1 Basit öznitelik tanımlayıcısı

Basit Öznitelik tanımlayıcısı gri seviyeye çevrilen resme ait önemli noktaların etrafında belirlenen büyüklükte alanların oluşturulması ile elde edilmektedir. Oluşturulan alanların içinde önemli nokta ve komşu noktalarına ait piksel değerleri bulunmaktadır. Özniteliklerin karşılaştırılması piksel değerleri kullanılarak yapılmaktadır (Ravi, 2015). Bundan dolayı basit öznitelik tanımlayıcısı rotasyon değişimlerinden etkilenen resimlerde beklenen sonuçları verememektedir.



Şekil 3.5 Öznitelik Oluşturulması (Visual Size, 2015)



Şekil 3.5’te numaralandırılmış sarı renkli alanlar resimdeki öznitelik alanlarıdır. Her bir öznitelik alanı merkezinde resme ait önemli noktalardan birini barındırmaktadır.

### 3.2.2 SIFT öznitelik tanımlayıcısı

SIFT öznitelik tanımlayıcısı resim birleştirme işlemlerinde yaygın olarak kullanılan bir tanımlayıcıdır. Temel olarak önemli noktaların etrafında oluşturulan alanların oryantasyon ve gradyan değerlerinin birlikte kullanılmasına dayanmaktadır (Utkarsh, 2015).

SIFT öznitelik tanımlayıcısı 2 aşamadan oluşmaktadır. İlk aşamada önemli noktaların etrafında 16x16 büyüklüğünde pencereler oluşturulmaktadır. Bir sonraki adımda oluşturulan pencerelerin içindeki piksellerin gradyan ve oryantasyon değerleri hesaplanmaktadır. Gradyan ve oryantasyon hesaplanması formülü Denklem 3.7 ve 3.8’de belirtilmektedir (Utkarsh, 2015).

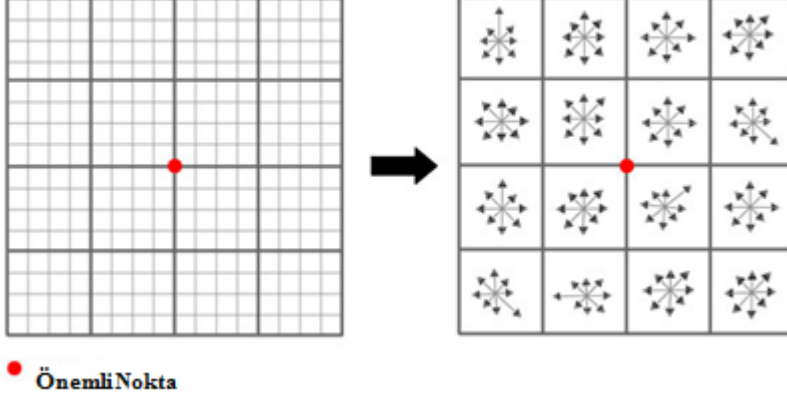
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.7)$$

$$\theta(x, y) = \arctan \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (3.8)$$

Denklem 3.7’de bir piksele ait gradyan değerinin hesaplanmasına ait formül gösterilmektedir. Denklemde  $m(x, y)$  değeri “L” resminde  $x, y$  koordinatlarına ait gradyan değerini ifade etmektedir. Denklem 3.8’de piksele ait oryantasyon değerine ait formül belirtilmiştir. Denklemde  $\theta(x, y)$  değeri “L” resminde  $x, y$  koordinatlarına ait oryantasyon değerini ifade etmektedir (Hemlata, 2013).

İkinci aşamada, ilk aşamada hesaplanan oryantasyon ve gradyan değerleri kullanılarak 128’lik SIFT histogramı oluşturulmaktadır (Peng et al., 2011). SIFT histogramı oluşturmak için ilk olarak önemli noktaların etrafında oluşturulan 16x16’lık alanlar 4x4’lük alanlara bölünmektedir. Bu şekilde 16 tane 4x4 lük alan oluşmaktadır. Bir sonraki adımda 4x4’lük her bir alan için; alanların içinde bulunan piksellerin oryantasyon değerleri  $0^\circ$ - $360^\circ$  arasında  $45^\circ$ ’lik farklarla oluşturulmuş 8 bölmeden uygun olanına yerleştirilir. Aynı bölmeye denk gelen

tüm piksellerin gradyant değerleri toplanır ve bu şekilde 4x4'lük alanlara ait histogram oluşturulur (Utkarsh, 2015).

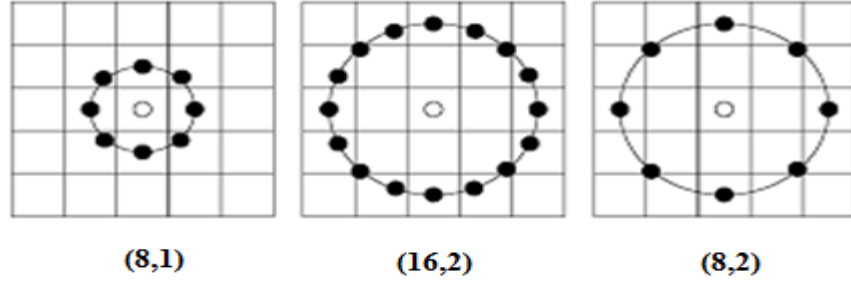


Şekil 3.6 16x16'lık alanların 4x4'lük alanlara bölünmesi (Utkarsh, 2015)

Şekil 3.6'da 16x16'lık alanların 4x4'lük alanlara bölünmesi ve 4x4'lük alanlara ait histogramların oluşturulması gösterilmektedir. Son adımda ise 4x4'lük alanlara ait histogramlar 128'lik (8x(4x4)) SIFT histogramını oluşturmak için uç uca eklenmektedir (Utkarsh, 2015). Oluşturulan 128'lik histogram önemli noktanın SIFT tanımlayıcısı olarak kabul edilmektedir.

### 3.2.3 Yerel ikili örüntü öznitelik tanımlayıcısı

Yerel ikili örüntü operatörü resim işlemede kullanılan öznitelik çıkarma metotlarından biridir. Yerel ikili örüntü öznitelik çıkarma işlemi önemli nokta merkez kabul edilerek etrafında oluşturulan belli büyüklükteki alanın içindeki komşu piksel değerleri arasındaki sıralı ikili karşılaştırmalardan oluşmaktadır. Yerel ikili örüntü operatörü her bir önemli nokta için ikilik tabanda bir dizi üretmektedir. Üretilen dizinin içine önemli nokta ve komşu piksellerin karşılaştırılmalarının sonuçları eklenmektedir. Eğer komşu piksel önemli noktadan büyükse "1", küçükse "0" eklenmektedir (Tekeli vd., 2015).



Şekil 3.7 Yerel İkili Örüntü operatörleri (Nabiyev vd., 2011)

Komşu piksel değerleri ile kıyaslanarak oluşturulan ikilik tabandaki dizi onluk tabana çevrilip önemli nokta için öznitelik tanımlayıcısı olarak kullanılmaktadır (Nabiyev vd., 2011). Şekil 3.8'de yerel ikili örüntü tanımlayıcısına ait örnek verilmektedir.

110	111	112	Karşılaştırma	1	1	1	İkili Kod: 11110000 Onlu Kod: 240
96	100	113		0	-	1	
97	98	99		0	0	0	

Şekil 3.8 Yerel İkili Örüntü ikilik taban dizisi oluşumu ve onluk gösterimi (Nabiyev vd., 2011)

Önemli noktalara ait oluşturulan Yerel İkili Örüntü tanımlayıcılarının hepsi karşılaştırma işlemi için kullanılmaz. İkili kodunda 0-1 ya da 1-0 geçiş sayısı 2 ya da daha az olanlar karşılaştırma için seçilmektedir kullanılmaktadır (Nabiyev vd., 2011). Seçilen noktaların Şekil 3.8'de ifade edildiği gibi onlu kodları hesaplanır ve karşılaştırma işlemi için bekletilir.

### 3.3 Öznitelik Eşleme Yöntemleri

Öznitelik eşleme yöntemleri, önemli noktalara ait özniteliklerin oluşturulmasından sonra iki resim arasındaki benzerliklerin tespit edilmesinde kullanılmaktadır. Bu bölümde öznitelik eşleme yöntemleri incelenmektedir.

### 3.3.1 Öklid uzaklığı kullanarak eşleme

Öklid uzaklığı formülü matematiksel olarak iki nokta arasındaki uzaklığı gösteren formüldür. Koordinatları verilen  $p(p_1, p_2)$  ve  $q(q_1, q_2)$  noktaları arasındaki uzaklık Denklem 3.9'da gösterilmektedir.

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (3.9)$$

Resim birleştirme işlemlerinde Öklid uzaklığı Denklem 3.9'da belirtilen formüle benzer şekilde kullanılmaktadır. Deepak (2015), çalışmasında birbirinden farklı iki özneteliğin yakınlığını ölçmek için özneteliklerin içindeki piksel değerlerinin farkların karelerinin toplamını almıştır. Sonuç olarak çıkan değer en düşük olduğu noktalar birbirleri ile eşleştirilmiştir. Bu eşleştirme yöntemi basit ve hızlı olmasına karşın basit öznetelik tanımlayıcısı ile kullanıldığında ışık ya da rotasyon değişimlerinden etkilenen öznetelikler arasında sağlıklı sonuçlar vermemektedir.

### 3.3.2 Öklid ve uzaklık oranı kullanarak eşleme

Öklid ve Uzaklık Oranı kullanarak eşleme yöntemi Öklid uzaklığı kullanarak eşleme yönteminin gelişmiş halidir. Bu yöntemde amaç referans resimdeki özneteliğe Öklid uzaklığı derecesi olarak en yakın iki özneteliğin oranını hesaplamaktadır. Hesaplanan oranın belli bir eşik değerinin altında olması durumunda referans resimden gelen önemli öznetelik ile diğer resimdeki en yakın değere sahip öznetelik eşlenir (LU et al., 2009).

### 3.3.3 Normalizasyon uygulanmış çapraz ilişkilendirme metodu

Normalizasyon Uygulanmış Çapraz İlişkilendirme metodu temel olarak gri seviyedeki resimler üzerinde yaygın olarak kullanılan bir algoritmadır (Chen, 2014). Bu yöntemde ilk olarak hesaplanan özneteliklerin normalizasyonları yapılmaktadır. Normalizasyon sonucunda ışıktan kaynaklı bozulmalar öznetelikler üzerinde etkisiz hale getirilmektedir (Mubarak et al., 2004). Denklem 3.10'da normalizasyon uygulanmış çapraz ilişkilendirme metoduna ait formül belirtilmiştir.

$$\frac{\sum_{(i,j) \in W} I_1(i, j) \cdot I_2(x+i, y+j)}{\sqrt[2]{\sum_{(i,j) \in W} I_1^2(i, j) \cdot \sum_{(i,j) \in W} I_2^2(x+i, y+j)}} \quad (3.10)$$

Denklem 3.10'da belirtilen  $I_1(i,j)$  referans resimdeki öznitelik noktasını,  $I_2(x+i,y+j)$  karşılaştırılması yapılacak olan resimdeki öznitelik noktasını ifade etmektedir. Hesaplanan ilişkilendirmeler sonucunda maksimum değere sahip noktalar bu yöntemde eşlenmektedir (Abdelfatah and Haitham, 2013).

### 3.4 Aykırı Nokta Tespit Yöntemleri

Özniteliklerin eşleşmesinden sonra aykırı noktaların tespiti yapılmaktadır. Bu şekilde elde kalan noktalar üzerinden resimler doğru bir şekilde birleştirilmektedir.

#### 3.4.1 RANSAC Algoritması

RANSAC algoritması resim eşleme işlemlerinde aykırı verileri elemek ve eşyazımlılık (İng. *Homography*) matrisini hesaplamak için kullanılan bir algoritmadır. Belirsiz (İng. *Non-Deterministic*) bir algoritma olduğundan dolayı her zaman en iyi sonucu vereceği garanti edilmemektedir. Algoritmanın başarısı yaptığı iterasyon sayısı ile doğru orantılıdır (Abdelfatah and Haitham, 2013).

RANSAC algoritmasının ilk aşaması aykırı verilerin elenmesidir. RANSAC algoritması her bir iterasyonda aykırı verileri hesaplamaktadır. Veri için en iyi dönüşüm parametrelerini elde ettiği zaman gerekli dönüşüm matrisini hesaplamaktadır (Müezzinoğlu vd., 2014). (Yıldız, 2012), RANSAC algoritmasına ait temel basamakları aşağıdaki gibi özetlemektedir.

**Algoritma 1. RANSAC**

---

1. Rastgele model parametrelerini belirlemek için gerekli olan 4 nokta seç
  2. Eşyazımlılık matrisi ( $H$ ) hesapla
  3. Yığın içindeki noktaların (İng. inlier) sayısını ( $T$ ) belirle
  4. Eğer,  $T =$  yığın içinde hesaplanan maksimum nokta sayısı, sonlandır
  5. Değilse, 1 ile 4 arasındaki adımları tekrarla.
- 

RANSAC algoritmasında bir diğer işlev eşyazımlılık matrisinin hesaplanmasıdır. Eşyazımlılık matrisi iki resim arasındaki dönüşüm (İng. *Conversion*) matrisidir (Yıldız, 2012). Resim birleştirme işlemlerinde son adım olarak hesaplanan eşyazımlılık matrisi, eşleşen özniteliklerin kullanılmasıyla birleşecek resimlerin 2B ortamdan 3B ortama aktarılması sonrasında sorunsuz birleşmeyi sağlamaktadır.

## 4.TEZ YÖNTEMİ

Bu tez çalışmasında resimlerin birleştirilmesi için gereken özniteliklerin eşlenmesi adımı Sürü Zekâsı yaklaşımlarından Yapay Arı Kolonisi algoritmasından faydalanılmıştır. Bunun için ilk olarak resimler üzerindeki önemli noktalar tespit edilmiş ve tespit edilen noktalar etrafında öznitelikler oluşturulmuştur.

Resimlerin birleştirilmesi işlemi 3 aşamadan oluşmaktadır:

1. Resimlerdeki noktaların tespiti,
2. Özniteliklerin oluşturulması ve Sürü Zekâsı yaklaşımı kullanılarak özniteliklerin eşlenmesi,
3. Aykırı noktaların elenmesi, eşyazımlılık matrisi hesabı ve resimlerin birleştirilmesi.

Çizelge 4.1’de tez kapsamında kullanılan veri setinin detayları verilmektedir. Buna göre, 8 ayrı set kullanılmıştır. Her setin kaç resimden oluştuğu ve referans resim (RR) ve karşılaştırılacak resim (KR) olarak o setten kaç adet ikili (RR, KR) elde edildiği çizelgede verilmektedir. Toplamda 47 resim kullanılmaktadır. Resimlerde farklı ölçekler kapsamaktadır. 31 adet 320x240, 11 adet 202x306, 5 adet 195x291 resim bulunmaktadır. Veri setindeki bu farklı boyutlar yöntemin esnek uygulama için kullanılabilmesini hedeflemektedir.

Çizelge 4.1 Serilere ait bilgiler

Seri Adı	Seri Numarası	Setteki Resim Sayısı	Setten Üretilen İkili Resim Sayısı
Cul De Sac	1	11	10
Chester Riverside	2	6	5
Chester Riverside 2	3	5	4
Family Room	4	5	4
Grass	5	5	4
Office	6	5	4
Parking Lot	7	5	4
UCSB	8	5	4

#### 4.1 Noktaların Tespiti

Resim birleştirmede ilk adım olarak noktaların tespiti yapılmaktadır. Çalışmamızda RR üzerindeki noktaların tespiti için Harris algoritması kullanılmaktadır. Ayrıca Harris algoritmasının yanında SIFT algoritmasına da değinilmektedir. KR üzerindeki noktaların tespiti için Canny kenar bulma algoritmasından yararlanılmıştır. RR üzerinde hesaplanan önemli noktaların eşleniklerinin daha hızlı bulunabilmesi açısından KR üzerinde kenar bulma işlemi gerçekleştirilmektedir.

Shrivakshan ve arkadaşlarının 2012'deki çalışmasında Canny kenar bulma algoritmasının diğer kenar bulma algoritmalarına karşı olan üstünlüğünü ele almıştır. Canny algoritmasının gürültülü resimlerde diğer algoritmalara kıyasla daha verimli sonuçlar verdiği tespit edildiğinden dolayı çalışmamızda KR üzerinde Canny kenar bulma algoritması kullanılmaktadır.

Bu tez çalışmasında RR üzerinde uygulanan Harris algoritması ile elde edilen noktaların sayısı girilen parametrelere bağlı değişmektedir. Çizelge 4.2, 4.3



ve 4.4'de, çizelge 4.1'de belirtilen resimlere ait Harris köşe nokta sayılarının parametrelere göre değişiminin sonuçları belirtilmektedir.

Çizelge 4.2 Seri 1'e ait farklı baskılama alanı ve eşik değerleri sonucu ortaya çıkan nokta sayısı

Resim	15x15 baskılama alanı, $\Omega \geq 1000$	10x10 baskılama alanı, $\Omega \geq 1000$	15x15 baskılama alanı, $\Omega \geq 750$	10x10 baskılama alanı, $\Omega \geq 750$
1	29	61	20	67
2	30	62	32	66
3	25	43	28	49
4	28	45	29	51
5	27	37	28	43
6	28	41	28	46
7	26	35	26	38
8	24	39	25	42
9	22	39	23	42
10	20	35	21	37
11	21	36	21	37

Çizelge 4.3 Seri 2'ye ait farklı baskılama alanı ve eşik değerleri sonucu ortaya çıkan nokta sayısı

Resim	15x15 baskılama alanı, $\Omega \geq 1000$	10x10 baskılama alanı, $\Omega \geq 1000$	15x15 baskılama alanı, $\Omega \geq 750$	10x10 baskılama alanı, $\Omega \geq 750$
1	35	72	36	81
2	39	76	41	81

Çizelge 4.4 Seri 5'e ait farklı baskılama alanı ve eşik değerleri sonucu ortaya çıkan nokta sayısı

Resim	15x15 baskılama alanı, $\Omega \geq 1000$	10x10 baskılama alanı, $\Omega \geq 1000$	15x15 baskılama alanı, $\Omega \geq 750$	10x10 baskılama alanı, $\Omega \geq 750$
1	32	74	33	77
2	35	79	36	84

Çizelge 4.2, 4.3 ve 4.4'te farklı serilerdeki resimlere uygulanan farklı miktarlarda baskılama alanı ve eşik değeri sonucunda ortaya çıkan nokta sayıları verilmiştir. Eşik değerinin azalması, algoritmanın nokta tespit ölçüsünü arttırmıştır. Bunun nedeni daha düşük değere sahip noktaların eklenmesidir. Bu noktalar arasında köşe noktalarından farklı olarak kenar noktaları da bulunabilmektedir. Baskılama alanı değerinin azalması ise noktaların etrafındaki kontrol edilecek alanı daraltır ve bu şekilde daha fazla nokta algoritma tarafından tespit edilmektedir.  $\Omega \geq 1000$  ve  $15 \times 15$  baskılama alanı değerlerinde hesaplama karmaşıklığını düşük tutacak, eşleşme doğruluğunu yükseltecek yeterli miktarda gürbüz noktayı çıkardığı gözlemlenmiştir. Çizelge 5.3'te belirtilen farklı yöntemlere ait sonuçların tamamı bu parametrelerle elde edilmiş olup baskılama alanı ve omega ile ilgili diğer performanslar bundan düşük olduğu için tez kapsamına alınmamıştır.

Bu tezdeki deneyler sonucunda bahsi geçen çizelgelerde görüldüğü üzere  $15 \times 15$  baskılama alanı ve  $\Omega \geq 1000$  değerlerinin tercih edilmesi gereken parametreler olduğu saptanmıştır.

RR üzerinde belirlenen noktaların gürbüz olması eşleşme esnasında daha verimli sonuçlar elde etmemize kolaylık sağlamaktadır. Bunun nedeni RR üzerindeki nokta sayısının artması çalışmamızdaki karmaşıklığı arttırmaktadır. Karmaşıklığın artması ise algoritmanın çalışma süresini ve işlem gereksinimlerini arttırmaktadır.

SIFT önemli nokta tespit algoritmasında Gaussian filtresinin  $k$  ve  $\sigma$  değerleri Utkarsh'ın belirlediği değerler baz alınarak yapılmıştır. Aday noktaların elenmesi için kullanılan kontrast değeri 0.03'ten küçük olan noktalar ve Hessian matrisine verilen tepki değerinin 10'dan büyük olduğu elenmektedir (Utkarsh, 2015). Çizelge 4.5, 4.6 ve 4.7'de Harris için kullanılan serilerin aynılarını SIFT algoritmasından geçirilip elde edilen nokta sayıları hesaplanmış olarak gösterilmektedir.

SURF algoritması SIFT algoritmasına alternatif olarak geliştirilen bir algoritmadır. SURF nokta tespit metodunda aynen SIFT'te oluşturulan ölçeksel uzay oluşturulmaktadır. SURF algoritmasında SIFT'ten farklı olarak ölçeksel uzay Gauss ile bulanıklaştırılmış resimlerden oluşmamaktadır. SURF algoritması Laplacian operatörünü tercih etmiştir. Ölçeksel uzayın oluşturulmasından sonra noktaların önemli nokta olup olmadığını tespit etmek için Hessian Matrisi

kullanılmaktadır. Hessian Matrisi aynen SIFT algoritmasında olduğu gibi yüksek değere sahip noktaları ortaya çıkarmaktadır. SURF nokta tespit metodundaki son adım nokta baskılama işlemi uygulanmasıdır. Nokta baskılama işlemi, SIFT metodunda uygulanan DOG piramidindeki oktavlar içindeki resimlerin ardışık üçerli gruplanması ve gruplardan uç değerlerin tespit edilmesi ile gerçekleşmektedir. Uç noktalar ve Hessian matris değerinin eşik değerinden yüksek olduğu noktalar seçilmektedir (Lei Yang, 2011). Lei Yang çalışmasında SURF algoritmasının SIFT'e göre daha hızlı olduğunu belirtmiştir. Bunun yanında SURF algoritmasının resimlerdeki ışıktan kaynaklı değişimlere karşı daha az tepki gösterdiğini açıklamıştır. Bu nedenden dolayı SURF algoritması çalışmamızda tercih edilmemiştir.

Köşe tespitinde kullanılan diğer algoritmalarından bir tanesi de FAST algoritmasıdır. FAST algoritması 1998 yılında geliştirilen bir köşe algılama algoritmasıdır (Traykovic et al., 1998). Köşe noktaları özneliklerin eşlenmesi için önemli olduğundan dolayı FAST algoritması kenarlar üzerinden köşe algılamasına öncelik vermiştir. Bunun nedeni, iki boyutlu yoğunluk değişimi gösterdiklerinden dolayı komşu noktalardan daha iyi ayırt edilebilmektedir (Varol, 2014). FAST algoritması oryantasyon değişkenlerini hesaplamamaktadır. Oryantasyon değişkeni yerine noktaların konumlandırılmasında Bresenham çemberinden yararlanmaktadır (Rydholm, 2015). Rydholm, çalışmasında FAST algoritmasının hızlı olmasına karşın ölçek değişmezliğini sağlamadığını belirtmektedir. Bu nedenden dolayı FAST algoritması çalışmamızda tercih edilmemiştir.

FAST algoritmasının ölçek değişmezliği ve oryantasyon değişkeninin yoksunluğu ORB algoritmasının nokta tespit metodunda ele alınmıştır. ORB algoritmasında ölçeksel uzay oluşturularak ölçek değişmezliği sağlanmaktadır. ORB momentleri kullanarak FAST noktalarına oryantasyon değişkenini sağlamaktadır (Rydholm, 2015).

Çizelge 4.5 Seri 1'e ait SIFT algoritmasının nokta tespit ölçüsü

Resim	SIFT nokta sayıları
1	101
2	107
3	93
4	80
5	85
6	93
7	73
8	79
9	73
10	85
11	65

Çizelge 4.6 Seri 2'ye ait SIFT algoritmasının nokta tespit ölçüsü

Resim	SIFT nokta sayıları
1	126
2	157

Çizelge 4.7 Seri 5'ye ait SIFT algoritmasının nokta tespit ölçüsü

Resim	SIFT nokta sayıları
1	125
2	128

Çizelge 4.5, 4.6 ve 4.7'de SIFT sonuçları gösterilmektedir. Harris nokta tespit ölçülerine göre SIFT algoritmasının nokta tespit ölçüleri daha yüksektir. Bunun nedeni SIFT algoritmasının resim üzerindeki noktaları tespit ederken aynı

resmin farklı boyutlarını kullanıyor olmasından kaynaklanmaktadır. Bu çözelgelere dayanarak SIFT önemli nokta algoritması yöntemimizde tercih edilmemiştir.

Harris algoritması, SIFT ve SURF algoritmalarında olduğu gibi farklı yollarla oryantasyon değişkenini kullanıp rotasyon değişmezliğini sağlamaktadır. Bunun yanında SIFT ve SURF algoritmalarına göre daha hızlı oluşundan dolayı bu algoritmalarının yerine tercih edilmiştir. Aynı zamanda ORB ve FAST algoritmalarına göre daha fazla parametre içermesi çalışmamız açısından daha az sayıda ve daha gürbüz noktaların çıkmasını sağlamaktadır. Bu nedenlerden dolayı Harris algoritması çalışmamızda tercih edilmiştir.

## 4.2 Öznitelik Oluşumu ve Eşlenmesi

Parçacık Sürü Optimizasyonu (İng. *Particle Swarm Optimization, PSO*) 1995'te Dr. Eberhart ve Dr. Kennedy tarafından geliştirilmiş popülasyon temelli sezgisel bir optimizasyon tekniğidir. Kuş veya balık sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir ve popüler bir optimizasyon tekniğidir (Shi, Y. and Eberhart, R. C., 1998).

PSO bir grup rasgele üretilmiş çözümlerle (parçacıkla) başlatılır ve jenerasyonlar güncellenerek en uygun değer araştırılır. Her iterasyon da, her bir parçacık iki “*en iyi*” değere göre güncellenir. Bunlardan birincisi bir parçacığın o ana kadar bulduğu en iyi uygunluk değeridir. Ayrıca bu değer daha sonra kullanılmak üzere hafıza tutulur ve “*pbest*” yani parçacığın en iyi değeri olarak isimlendirilir. Diğer en iyi değer ise popülasyondaki herhangi bir parçacık tarafından o ana kadar elde edilmiş en iyi uygunluk değerine sahip çözümdür. Bu değer popülasyon için en iyi değerdir ve “*lbest*” olarak isimlendirilir (Karaboğa, D., 2014). Derviş Karaboğa'ya ait Yapay Arı Kolonisi algoritmasına ait temel adımlar algoritma 2'de belirtilmektedir.

### **Algoritma 2.** ABC Algoritması

1. Başlangıç yiyecek kaynağı gölgelerin üretilmesi

#### **REPEAT**

2. Görevli arıların yiyecek kaynağı bölgelere gönderilmesi

3. Olasılıksal seleksiyonda kullanılacak olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması

4. Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmesi

5. Bırakılacak kaynakların bırakılışı ve kâşif arı üretimi

**UNTİL** (çevrim sayısı=Maksimum çevrim sayısı)

Algoritma 2’de belirtilen Karaboğa’ya ait ABC algoritmasında bahsedilen işçi arılar, keşfedilen belli kaynaklara ait nektarın kovana getirilmesinden sorumludur. Kâşif arılar, rastgele kaynak arayışında olan görevsiz işçi arılardır. Gözcü arılar ise işçi arılar tarafından paylaşılan bilgiyi kullanıp yeni bir kaynağa yönelen arılardır (Karaboğa, D., 2014).

Bu tez çalışmasında önemli noktaların eşlenmesi için ilk olarak noktaların SIFT öznitelikleri (İng. *descriptor*) . Eşleşme için Yapay Arı Kolonisi algoritması kullanılarak eşleme yapılmıştır.

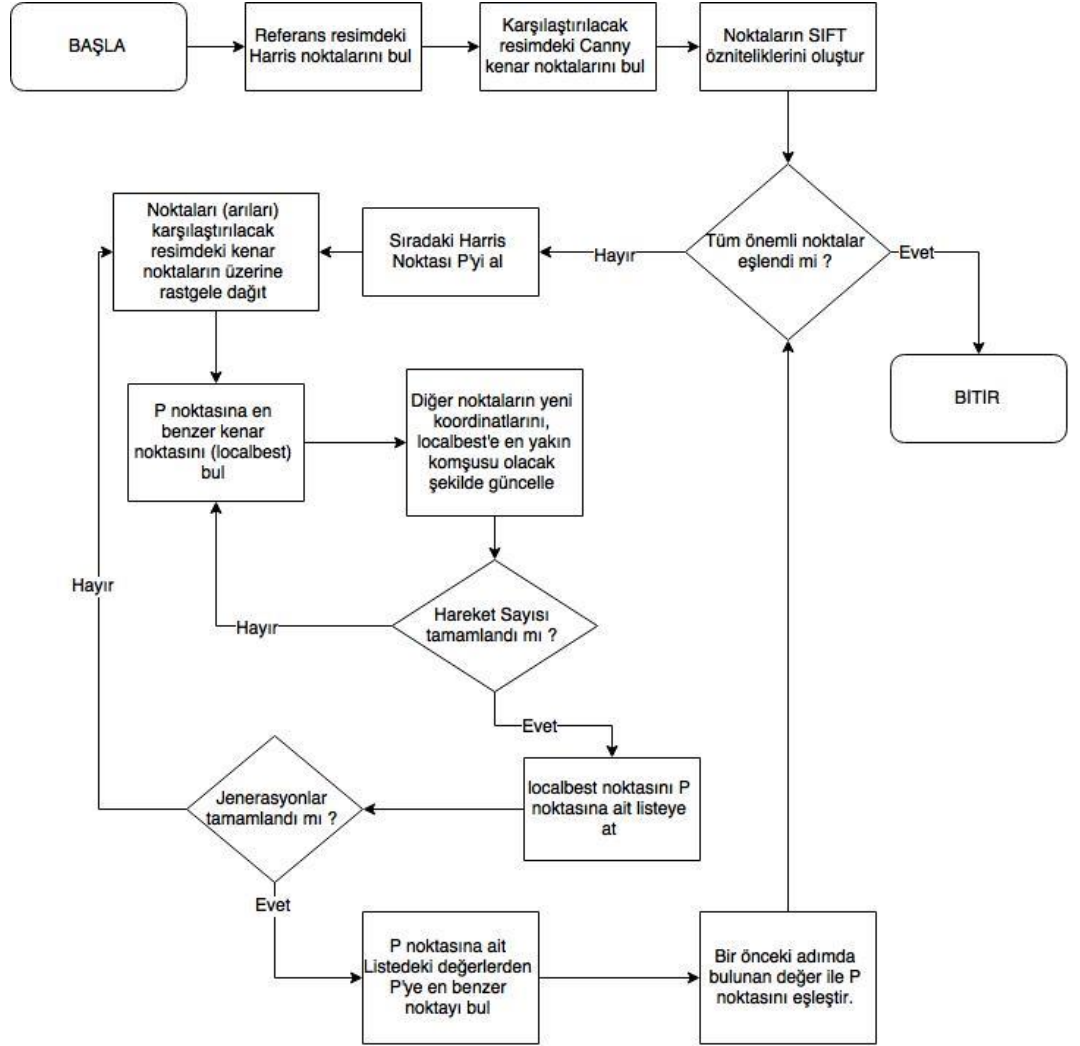
RR üzerinde Harris algoritması ile köşe noktaları çıkarılmıştır. KR üzerinde Canny kenar bulma algoritması uygulanmaktadır. Kenar noktaların ve RR üzerindeki köşe noktalarının SIFT öznitelikleri elde edilmiştir. RR ve KR üzerinde nokta eşleştirmesi için tezde sunulan ABC algoritmasının varyantı olan AB-CD algoritması uygulanmaktadır. AB-CD algoritmasına ait temel adımlar algoritma 3’te belirtilmektedir.

### **Algoritma 3. AB-CD Algoritması**

---

1. RR üzerindeki önemli noktalar KR üzerindeki Canny kenar noktalarını bul.
  2. Adım 1’de hesaplanan tüm noktaların SIFT özniteliklerini çıkar.
  3. RR üzerindeki her bir köşe noktası ( $P_i$ ) için
  4. KR üzerine noktaları rastgele olarak dağıt (sürü).
  5.  $P_i$  ile Adım 2’de oluşturulan noktalar arasında SIFT özniteliklerinin Öklid uzaklıklarını hesapla.
  6.  $P_i$  noktasına en yakın değere sahip olan noktayı yeni lbest noktası olarak işaretle.
  7. Adım 4’te dağıtılan noktaların koordinatlarını, lbest noktasına en yakın komşu noktasının koordinatları olacak şekilde güncelle.
  8. 5. ve 7. adımlar arasındaki işlemleri hareket sayısı kadar tekrarla
  9. lbest noktasını  $P_i$  noktasına ait listeye ( $P_{il}$ ) gönder.
  10. 4. ve 9. adımlar arasındaki işlemleri jenerasyon sayısı kadar tekrarla
  11.  $P_{il}$  listesinden  $P_i$  noktasına değer olarak en yakın noktayı ( $P_a$ ) bul
  12.  $P_i$  ile  $P_a$  noktalarını eşle
  13. 3. ve 12. adımlar arasındaki işlemleri tüm önemli noktalar için tekrarla
-

Algoritma 3'te belirtilen AB-CD algoritmasına ait akış diyagramı 4.1'de belirtilmektedir.



Şekil 4.1 Sürü Zekâsı ile resim birleştirme akış diyagramı

Şekil 4.1'de AB-CD algoritmasına ait akış diyagramı gösterilmektedir. Buna göre tez yönteminde ilk olarak RR üzerinde seçilen önemli nokta ele alınmaktadır. Daha sonra KR üzerindeki kenar noktalarının bulunduğu bölgelere belirli sayıda nokta (arı) rastgele olacak şekilde dağıtılmaktadır. Dağıtılan her bir nokta ile RR üzerinde seçilen nokta arasında SIFT özelliklerine ait Öklid uzaklıkları hesaplanmaktadır. KR üzerinde RR'de seçilen önemli noktaya en yakın Öklid değerine sahip nokta dağıtılan noktalar arasında en iyi nokta “*lbest*” olarak işaretlenmektedir. Bir sonraki adımda KR üzerine dağıtılan noktalar, etrafında bulunan ve *lbest* noktasına en yakın komşu noktası olan kenar noktasına

hareket etmektedirler. Her hareket sonrasında lbest noktası yeniden hesaplanmaktadır. KR üzerine dağıtılan noktaların hareket sayıları belli bir değeri geçtikten sonra durmaktadır. 4.1’de KR üzerindeki noktaların lbest noktasına en yakın olan komşusuna hareketi formülize edilmiştir.

$$V_{ij} = X_{\text{komşu}} \text{ ve } X_{\text{komşu}} = \min ( L_2(X_i, \text{lbest}) ) \quad (4.1)$$

4.1’de belirtilen denklemde  $X_i$ , KR üzerine dağıtılan noktalardan herhangi birinin konumunu belirtmektedir.  $X_{\text{komşu}}$  ise  $X_i$  noktasının lbest noktasına en yakın komşu noktasıdır.

Hareket sayıları sonlandıktan sonra lbest noktası RR üzerinde seçilen noktaya ait listede tutulmaktadır. Bu işlemlerden sonra dağıtılan noktalar kaldırılmakta ve KR üzerine yeniden başka noktalar (jenerasyon) dağıtılmaktadır. Bu işlemler aynı RR resmindeki noktada birden fazla kez tekrarlanmakta ve sonuç olarak RR üzerinde seçilen noktaya ait listede birden fazla lbest noktası bulunmaktadır. Bir sonraki adımda listedeki lbest noktaları arasından RR üzerinde seçilen noktaya en yakın değer eşlenmektedir.

En yakın değerın tespit edilmesi kaba kuvvet eşleme (İng. *Brute force matcher*) kullanılarak yapılmamaktadır. RR üzerinde seçilen noktaya ait listedeki değerler RR noktasına yakınlıklarına göre artan sıralamaya tabi tutulurlar. Sıralama sonucunda oluşan listedeki ilk değer ile RR üzerinde seçilen nokta eşlenmektedir. Bu eşleşme yöntemine ait olarak FLANN kütüphanesi kullanılabilir. FLANN kütüphanesi en yakın komşuyu tespit edebilmek için hızlı yöntemler sunan bir kütüphanedir. İçerisinde birden fazla algoritma barındırmaktadır. FLANN otomatik olarak veri setine en uygun olan yöntemi belirlemektedir. Bu yöntemler arasında K-D ağaç algoritması, rastgeleleştirilmiş K-D ağaç algoritması ve K-Means kümeleme algoritmaları bulunmaktadır (Muja et al.,2014). FLANN kütüphanesi kullanılarak aranan değere en yakın değer arama uzayı daraltılarak daha kolay bir şekilde tespit edilebilir. Fakat FLANN kütüphanesi tez çalışmamızın kapsamında yer almadığı için kullanılmamaktadır. Eşleşme ait son işlemler RR üzerinde belirlenen tüm noktalar için yapılmaktadır. Tüm noktaların eşlenmesinden sonra işlem sonlanmaktadır.





Şekil 4.2 (a) Referans resim-1 (b) Karşılaştırılacak resim-1 (Visual Size, 2015)

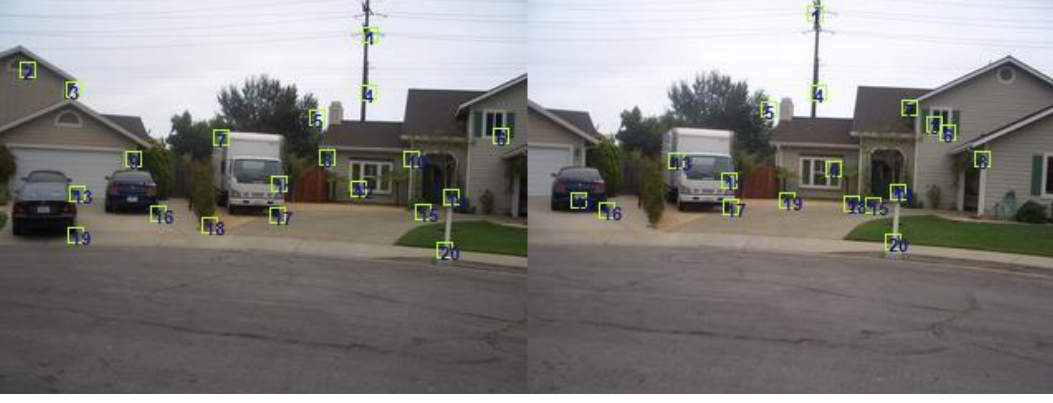


Şekil 4.3 Sürü Zekâsı Yaklaşımı ile noktaların eşlenmesi-1 (Visual Size, 2015)

Şekil 4.2’de resimlerin orjinal halleri gösterilmektedir. Şekil 4.3’te ise referans resimdeki önemli noktalar ile karşılaştırılacak resimdeki köşe noktaların eşlenmesinin sonuçları gösterilmektedir. Sarı kare içine alınan bölgeler öznelik bölgeleridir ve farklı resimlerdeki aynı numaraya sahip alanlar eşlenmiş bölgeleri belirtmektedir.

### 4.3 Aykırı noktaların elenmesi

Aykırı noktaların elenmesi ve Eşyazımlılık matrisi RANSAC algoritması tarafından gerçekleştirilmektedir. RANSAC algoritmasının başarısı literatürdeki birçok resim birleştirme işlemlerinde kanıtlanmıştır. Aykırı noktaların elenmesinden sonraki işlem olan Eşyazımlılık matrisinin hesaplanmasından sonra resimler birleştirilmektedir. Aşağıdaki şekillerde RANSAC algoritmasının yanlış eşleşen noktaları elemesinin örnekleri belirtilmiştir. Örnekteki farklı resimlerdeki aynı numaraya sahip alanlar eşlenmiş alanlardır.



Şekil 4.4 RANSAC uygulanmamış resim (Visual Size, 2015)

Şekil 4.4'de öznelikleri eşlenmiş bir resim gösterilmektedir. Bu görüntüye RANSAC uygulayınca Şekil 4.5'te belirtilen görüntü ortaya çıkmaktadır.



Şekil 4.5 RANSAC algoritması ile aykırı noktaların elenmesi sonrası (Visual Size, 2015)

Şekil 4.5'te sarı kare içine alınan noktalar RANSAC algoritması sonrası aykırı noktaların elenmesinden sonra geriye kalan noktalardır.

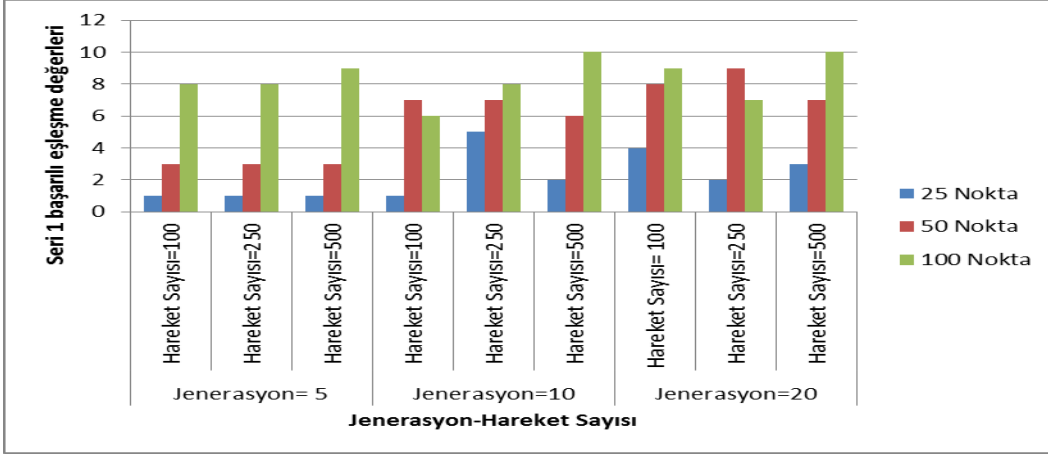
## 5. TARTIŞMA VE DENEYSEL SONUÇLAR

Bu tez çalışmasında tercih edilmesi gereken parametrelerin tespiti için biri 11 (Seri 1) diğeri 6 (Seri 2) olmak üzere toplam 17 tekil resim üzerinde birleştirme işlemi denenmiştir. Seri 1’de 10 eşleşme, Seri 2’de 5 eşleşme yapılmaktadır. Sonuç olarak tez çalışmasının esnek parametreleri kullanılarak toplam 15 eşlenmiş resim üzerinden 405 eşlenmiş sonuç resim elde edilmiştir. Seri 1’e ait sonuçların bulunduğu çizelge aşağıda gösterilmektedir.

Çizelge 5.1 Seri 1’e ait birleştirme sonuçları

Dağıtılan nokta sayısı	5 Jenerasyon						10 Jenerasyon						20 Jenerasyon					
	100		250		500		100		250		500		100		250		500	
	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket	Hareket
	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-
25	1	9	1	9	1	9	1	9	5	5	2	8	4	2	2	8	3	7
50	3	7	2	8	6	4	7	3	7	3	6	4	8	9	9	1	7	3
100	8	2	8	2	9	1	6	4	8	2	10	0	9	7	7	3	10	0

Çizelge 5.1’de Seri 1 kullanılarak yapılan 10 eşleştirme sonucundaki tez çalışmasının parametrelerine göre başarılı ya da başarısız sayısındaki değişim gösterilmektedir. Çizelgedeki “+” kolonuna ait değerler başarılı, “-“ kolonundaki değerler başarısız birleşmelerin sayısını göstermektedir. Çizelge 5.1’e ait sütun grafik aşağıda belirtilmektedir.



Şekil 5.1 Seri 1'e ait başarılı birleştirme sayıları grafiği

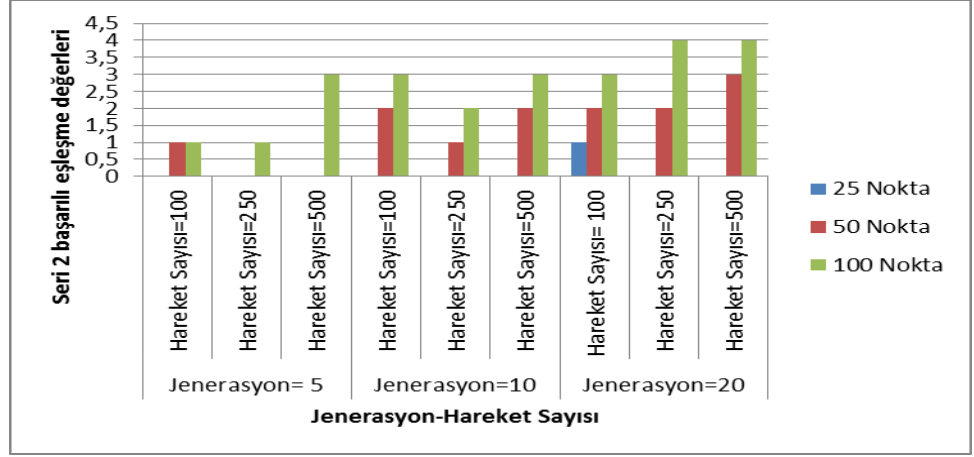
Şekil 5.1'de Seri 1'e ait başarılı birleşmelerin sayısının grafiği verilmiştir. Grafikten yola çıkarak jenerasyon sayısının artması, birleşme oranındaki başarıyı yükseltmiştir. Bunun yanında KR (Karşılaştırılacak resim) üzerindeki nokta sayısını artışı, birleşme oranını yükseltmiştir. Bu işlemler sonucunda Seri 1 için tercih edilen hareket sayısı 500 olarak belirlenmiştir.

Bu tez çalışmasında tercih edilecek parametrelerin belirlenmesinde Seri 2 kullanılmaktadır. Tek bir seferde 6 tekil resim kullanılarak ardışık 5 birleştirme yapılmıştır. Birleştirmelere ait sonuçların bulunduğu çizelge 5.2'de belirtilmektedir.

Çizelge 5.2 Seri 2'ye ait birleştirme sonuçları

Dağıtılan nokta sayısı	5 Jenerasyon						10 Jenerasyon						20 Jenerasyon					
	100 Hareket		250 Hareket		500 Hareket		100 Hareket		250 Hareket		500 Hareket		100 Hareket		250 Hareket		500 Hareket	
	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-
25	0	5	0	5	0	5	0	5	0	5	0	5	1	4	0	5	0	5
50	1	4	0	5	0	5	2	3	1	4	2	3	2	3	2	3	3	2
100	1	4	1	4	3	2	3	2	2	3	3	2	3	2	4	1	4	1

Çizelge 5.2’de Seri 2 kullanılarak yapılan 5 eşleştirme sonucundaki tez çalışmasının parametrelerine göre başarılı ya da başarısız sayısındaki değişim gösterilmektedir. Sonuçlar Seri 1 sonuçları ile benzerlik göstermektedir. Çizelge 5.2’ye ait sütun grafik aşağıda gösterilmektedir.

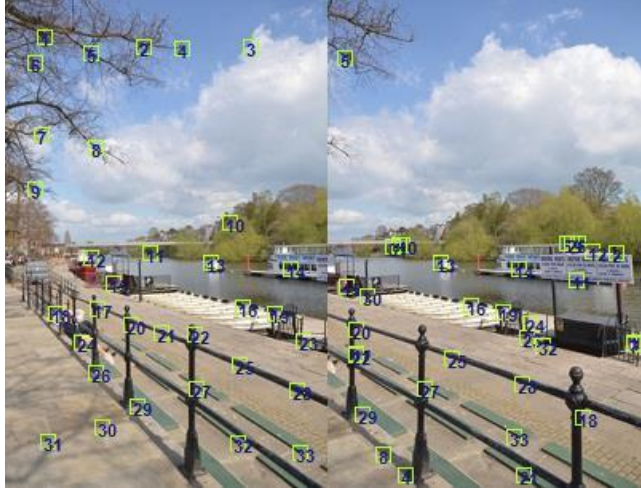


Şekil 5.2 Seri 2’ye ait başarılı birleştirme sayıları grafiği

Bu tez çalışmasının, çizelgelerin, şekillerin ve yapılan deneylerin sonucunda jenerasyon sayısının 20, KR üzerine dağıtılan nokta sayısının 100 ve hareket sayısının 500 olduğu durumlarda verimli olarak çalıştığı gözlemlenmiştir. Dağıtılan nokta sayısı parametresinin azalması karşılaştırılacak resim üzerine dağıtılacak nokta sayısının azalması anlamına gelmekte ve bu şekilde daha az karşılaştırma yapılabilmektedir. Fakat daha az karşılaştırma bize eşlemenin doğru bir şekilde yapılabileceği garantisini vermemektedir. Hareket sayısı parametresi sayesinde dağıtılan noktalar KR üzerinde daha çok gezebilmekte ve bu sayede daha fazla karşılaştırma yapılabilmektedir. Jenerasyon parametresi ise noktaların toplam hareket sayılarını yeniden kaçır kere tekrar edeceklerini gösterme ve bu sayede bir önceki jenerasyonda doğru eşleme olmaması durumunda yeni bir deneme yapılabilmektedir.



Şekil 5.3 Orjinal Resimler (a)Referans Resim-2 (b)Karşılaştırılacak resim-2 (Visual Size, 2015)



Şekil 5.4 Özniteliklerin Sürü Zekâsı ile eşlenmesi-2 (Visual Size, 2015)

Şekil 5.3'te birleştirilecek olan resimlerin orjinal halleri gösterilmektedir. Şekil 5.4'te ise tez çalışmasında verimli olarak belirlenen 20 jenerasyon, 100 nokta sayısı ve 500 hareket sayısı parametreleri girilerek elde edilen eşleşme sonucu gösterilmektedir. Yanlış eşleşmelerin elenmesi RANSAC algoritması tarafından gerçekleştirilmektedir. Şekil 5.5'te yanlış noktaların elenmesi sonrasında elde kalan eşleşmeler gösterilmektedir.





Şekil 5.5 RANSAC sonucu-2 (Visual Size, 2015)

Şekil 5.6'da kalan noktalar kullanılarak birleşmenin sonucu gösterilmektedir.



Şekil 5.6 Birleşme sonucu-2 (Visual Size, 2015)

Bu tez çalışmasında, yapılan deneyler sonucu verimli olarak kabul edilen 20 jenerasyon, 100 nokta ve 500 hareket sayısı resim veri setine uygulanmış, noktalar eşleştirilmiş ve resimler birleştirilmiştir. Bu eşleşmelere ait bazı başarılı ve başarısız sonuçlara ait görseller aşağıda gösterilmektedir.



Şekil 5.7 Orjinal Resimler (a) Referans Resim-3 (b) Karşılaştırılacak resim-3 (Visual Size, 2015)



Şekil 5.8 PSO ile noktaların eşlenmesi-3 (Visual Size, 2015)



Şekil 5.9 RANSAC algoritması sonrası-3 (Visual Size, 2015)



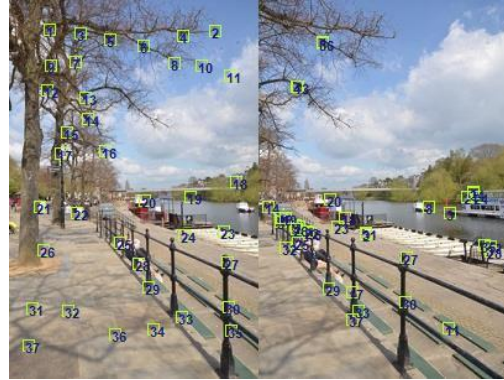
Şekil 5.10 Birleşme sonucu-3 (Visual Size, 2015)



Şekil 5.7, 5.8, 5.9 ve 5.10’da başarısız olarak eşlenen iki resme ait birleştirme süreci gösterilmektedir. RANSAC algoritmasının birleşme için gerekli olan düzgün noktaları seçemediğinden dolayı bu birleştirme işlemi başarısız olmuştur. Şekil 5.11, 5.12, 5.13 ve 5.14’te başarısız olarak birleştirilen bir başka resim grubuna ait veri gösterilmektedir.



Şekil 5.11 Orjinal Resimler (a) Referans Resim-4 (b) Karşılaştırılacak resim-4 (Visual Size, 2015)



Şekil 5.12 PSO ile noktaların eşlenmesi-4 (Visual Size, 2015)



Şekil 5.13 RANSAC algoritması sonrası-4 (Visual Size, 2015)



Şekil 5.14 Birleşme sonucu-4 (Visual Size, 2015)

Şekil 5.14 başarısız bir birleştirme sonucudur. Bunun nedeni PSO algoritmasının RANSAC algoritmasının yığın hesaplamasına yetecek kadar doğru eşleştirme üretememesinden kaynaklanmaktadır. Aşağıdaki şekillerde başarılı olarak eşleştirilen resim ikilisine yer verilmektedir.



Şekil 5.15 Orjinal Resimler (a) Referans Resim-5 (b) Karşılaştırılacak resim-5 (Visual Size, 2015)



Şekil 5.16 PSO ile noktaların eşlenmesi-5 (Visual Size, 2015)



Şekil 5.17 RANSAC algoritması sonrası-5 (Visual Size, 2015)



Şekil 5.18 Birleşme sonucu-5 (Visual Size, 2015)

Resim birleştirme işlemlerinde tez çalışmamızın başarısı bu alanda kullanılan yaygın algoritmalar olan SIFT, Normalized Cross-Correlation ve Yerel İkili Örüntü yöntemleri ile kıyaslanmıştır. Kıyaslamaların sonuçlarına dair bilgiler çizelge 5.3'te gösterilmektedir.

Çizelge 5.3 Performans Sonuçları ve Algoritmik Karmaşıklıklar

Algoritmaların Karşılaştırılması				
	Normalize edilmiş Çapraz İlişkilendirme	SIFT	PSO	Yerel İkili Örüntü
Setteki resim sayısı	47	47	47	47
Setten üretilen ikili resim sayısı	39	39	39	39
Başarılı birleşme sayısı	37	36	29	0
En iyi durumda algoritmik karmaşıklık	$\Omega(1)$	$\Omega(1)$	$\Omega(n^3)$	$\Omega(1)$
En kötü durumda algoritmik karmaşıklık	$O(n^2)$	$O(n^2)$	$O(n^4)$	$O(n^3)$

Çizelge 5.3'te performans sonuçları gösterilmektedir. Sonuçlar oluşturulurken veri seti olarak Visual Size veri seti kullanılmıştır. Resimlerin birleştirilmesi işlemi 8 GB RAM, AMD A-10-4600M APU modeli işlemcisine sahip 64 bitlik işletim sistemi içeren bir makine üzerinde gerçekleştirilmiştir.

Algoritmalara ait en iyi ve en kötü durumlardaki karmaşıklıklar çizelge 5.3'te belirtilmektedir. Algoritmik karmaşıklıklar hesaplanırken RANSAC algoritmasının en iyi ve en kötü durumdaki karmaşıklıkları hesaba katılmamaktadır. RANSAC algoritmasının en iyi durumdaki karmaşıklığı belirlenen “k” iterasyon sayısı ile  $\Omega(kn)$ , en kötü durumdaki karmaşıklığı ise  $O(n^3)$  olarak hesaplanmaktadır. Sonuç olarak algoritmalar RANSAC ile birlikte

kullanıldıklarında Normalize edilmiş çapraz ilişkilendirme yöntemi en iyi durumda  $\Omega(1) + \Omega(kn)$  toplamından  $\Omega(kn)$ , en kötü durumda ise  $O(n^2) + O(n^3) = O(n^3)$  olarak belirlenmektedir. SIFT ve RANSAC algoritmasının birlikte kullanımı sonucunda ise en iyi durumdaki karmaşıklık değeri  $\Omega(kn)$  olarak, en kötü durumdaki karmaşıklık değeri  $O(n^3)$  olarak değişmektedir. Tez çalışmamızda kullanılan bir diğer metot olan Yerel ikili örüntü operatörü ve RANSAC algoritmasının birlikte kullanımında ise en iyi durumda karmaşıklık değeri  $\Omega(kn)$ , en kötü durumdaki karmaşıklık değeri  $O(n^3)$  olarak belirlenmektedir. Tez çalışmamızın asıl yöntemini oluşturan AB-CD algoritmasının RANSAC ile birlikte kullanılması sonucu en iyi durumda karmaşıklığın  $\Omega(n^3)$  ve en kötü durumdaki karmaşıklığın  $O(n^4)$  olduğu görülmektedir.

Resimler üzerinde önemli noktaların tespiti için Harris algoritması kullanılmıştır. Harris eşik değeri 1000, nokta baskılama alanı  $15 \times 15$  olarak kabul edilmiştir. Nokta eşleştirme için kullanılan Normalize edilmiş çapraz ilişkilendirme ve SIFT özniteliklerine ait parametrik yapılar bulunmamaktadır. Yerel ikili örüntü operatörlerinden hiçbirine ait başarılı sonuçlar elde edilmemiştir. Son olarak tez çalışmasının yöntemi olan PSO algoritmasında jenerasyon sayısı 20, hareket sayısı 500 ve karşılaştırılacak resim üzerine dağıtılacak nokta sayısı 100 olarak kabul edilmiştir.

Çizelge 5.3'e bakarak tez çalışmasının verimli olarak kabul edilen parametrelerinin SIFT ve Normalize edilmiş çapraz ilişkilendirme yöntemlerine göre başarı oranının daha düşük olduğu gözlemlenmektedir. Başarı oranının düşük olmasının ilk nedeni sistemin belli parametrelere bağlı olmasından kaynaklanmaktadır. Tez çalışması ile karşılaştırılan diğer algoritmalar, resimler üzerindeki önemli noktalar üzerinden gitmekte olup, tez çalışmamız ise referans resimdeki (RR) önemli noktalar ile KR üzerindeki kenar noktalar arasında eşleştirme yapmaktadır. Bu eşleştirme esnasında kullanılan parametreler genel olarak RR üzerindeki önemli nokta ile KR'deki kenar noktalarından hangilerini kıyaslayacağını etkilediğinden dolayı tez çalışmamız her zaman diğer algoritmalar kadar iyi sonuç vermemektedir.



## 6. SONUÇ

Tez çalışmamızda panoramik görüntü oluşturma işlemlerine ait özgün bir çalışma gerçekleştirilmektedir. Referans resim üzerinde köşe noktalarının tespiti ve karşılaştırılacak resimdeki Canny kenar noktalarının tespiti çalışmamıza özgünlük katan bir yöntemdir. Kalıp eşleştirme yöntemleri ile yapılan birleştirme işlemlerindeki aranmak istenen noktayı bulma karmaşıklığı dezavantajı, çalışmamızda karşılaştırılacak resim üzerindeki Canny noktalarının bulunması ile avantaj konumuna geçmiştir. Bu sayede aranmak istenen nokta daha kolay elde edilebilmektedir. Noktaların tespitinden sonra resimler üzerinde tespit edilen tüm noktaların SIFT özniteliklerinin eşleşmeden önce sadece bir kere oluşturulması tez çalışmamızdaki bir diğer özgün yöntemdir.

Tez çalışmamızda rastgele jenerasyon oluşturma ve eşleştirme aşamalarında yaşanan hesapsal zorluklar öznitelik eşleme oranındaki başarıyı diğer algoritmalara kıyasla daha düşük olmasını sağlamaktadır.

Bu tez çalışması, ileride uygulanabilecek farklı Sürü Zekâsı algoritmaları ile geliştirilebilir ve daha iyi sonuçlar elde edilebilir. Tez çalışmasına alternatif olarak ileriki çalışmalarda resimler üzerinde kesişen bölgeler tespit edilip kesişen bölgeler üzerinden sürü zekâsı algoritması çalıştırılabilir. Bu şekilde eşleşme yapılıp elde edilen sonuçlarla bizim çalışmamızdaki sonuçlar karşılaştırılabilir.





## KAYNAKLAR DİZİNİ

- Harris, C. and Stephens, M., 1988.** A combined corner and edge detector., *Proceedings of the 4th Alvey Vision Conference. pp. 147–151.*
- Başak, S.,** Harris ve Moravec Köşe Belirleme Metodları (Erişim Tarihi: 25 Mart 2015)
- Russol, A. and Haitham, O., 2013,** Automatic Seamless Of Image Stitching , *Computer Engineering and Intelligent Systems ISSN 2222-1719*
- Brown, M. and Lowe D. G., 2003,** Recognizing Panoramas, In *Proceedings of the Ninth IEEE international Conference on Computer Vision - Volume 2* (October 13 - 16, 2003). ICCV. IEEE Computer Society.
- Hemlata, J., 2013,** Image Mosaicing using Harris, SIFT Feature Detection Algorithm, *Internation Journal of Science,Engineering and Technology research (IJSETR), ISSN:2278-7798*
- Ashwini, P., 2014,** Image Mosaicing using SIFT and Corner Detection Algorithm, *International Journal of Advanced Technology and Engineering Research (IJATER)*
- Utkarsh, S.,** SIFT: scale space, <http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoints>, (Erişim Tarihi: 20 Ocak 2015)
- Chia, W. Yeong, L. and Ch'ng, S., 2014,** The Effect of Rainfall on Feature Points Extraction and Image Stitching., *IEEE, 978-1-4799-3197-2/14*
- Müezzinoğlu, T., Çolak, F. ve Karaköse, M., 2014,** Görüntü Mozaikleme Algoritması İçin Deneysel Bir Çalışma, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*
- Ravi, M.,** Feature Detection and Matching, Teknik Rapor, (Erişim Tarihi: 15 Nisan 2015)
- Visual Size,** <http://www.visualsize.com/Mosaic3D/index.php>, (Erişim Tarihi: 21 Aralık 2014)
- Peng, K. and Hongbing, M., 2011,** An Automatic Image Mosaicing Method Based on SIFT Feature Matching, *IEEE 978-1-61284-774-0/11*
- Deepak, P.,** Feature Decription and Matching, Teknik Rapor, (Erişim Tarihi: 24 Nisan 2015)

**KAYNAKLAR DİZİNİ (devam)**

- Tekeli, E. Çetin, M. ve Erçil, A.,** Yerel İkili Örüntü (LBP) ve Önsel Şekil Bilgisi Tabanlı bir Desen Bölütleme Metodu, (Erişim Tarihi: 23 Haziran 2015)
- Nabiyev, V. ve Günay, A., 2011,** Yerel Görüntüdeki Kişinin Yaşının Bulunması, *Çankaya University Journal of Science and Engineering*
- Lu, X. ve Wang, J., 2009,** An Improved Algorithm for Image Mosaic Based on Speeded-up Robust Features
- Chen, K., 2014,** Image Stitching Algorithm Research Based on OpenCV, *Proceedings of the 33rd Chinese Control Conference*
- Mubarak, S. ve Yaser, S., 2004,** Aligning ‘Dissimilar’ Images Directly
- Yıldız, C., 2012 ,** An Implementation on Recognizing Panoramas, Teknik Rapor
- Jignesh, S., Suprava, P. And Salman, B., 2009,** Image Registration by Template Matching Using Normalized Cross-Correlation, *2009 International Conference on Advances in Computing, Control and Telecommunication Technologies*
- Xiaoxiang, L., Weigang, J. And Jianwen, X., 2009,** An Image Template Matching Using Particle Swarm Optimization, *IEEE 978-1-4244-6-4607-0/09*
- Yang, L., Xiaoyu, W., Jun. Z. And Hui, Li., 2011,** A Research of Feature-based Image Mosaic Algorithm, *IEEE 978-1-4244-9306-7/11*
- Aathreya, S., Amith, V., Namrata, S. And Chaitraya, N., 2013,** Template Matching Technique for Panoramic Image Stitching, *IEEE 978-0-7695-5101-2/13*
- Niu, J., Yang, F. And Shi, L., 2013,** Improved Method of Automatic Image Stitching Based on SURF, *IEEE 978-1-4799-0764-9*
- Manisha, D., 2014,** Image Stitching using Matlab, *IJITEE 2278-3075*
- Rupali, C. And Vrishali, C., 2014,** Image Stitching using Harris Feature Detection and Random Sampling, *International Journal of Computer Applications(0975-8887)*

**KAYNAKLAR DİZİNİ (devam)**

- Karaboğa, D., 2005**, An Idea Based on Honey Bee Swarm for Numerical Optimization, Teknik Rapor-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department
- Tamer, S. ve Karakuzu, C.**, Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri, Teknik Rapor, (Erişim Tarihi: 24 Mart 2015)
- Karaboğa, D., 2014**, Yapay Zeka Optimizasyon Algoritmaları, 3. Basım, Nobel Yayınevi, Ankara, 244s.
- Shi, Y. and Eberhart, R. C., 1998**, A modified particle swarm optimizer., *IEEE* 0-7803-4869-9/98
- Özgen, N. ve Sarıtaş, M.**, SIFT metodu ile Hedef Takibi, Teknik Rapor, (Erişim Tarihi: 8 Temmuz 2015)
- Shrivakshan, G.T. and Chandrasekar, C., 2012**, A Comparison of various Edge Detection Techniques used in Image Processing, *IJCSI* 1694-0814
- Viswanathan, D.G.**, Features From Accelerated Segment Test, Teknik Rapor, (Erişim Tarihi: 9 Haziran 2015)
- Varol, A.H., 2014**, Görüntü Mozaikleme ile Panoramik Görüntü Üretimi, Yüksek Lisans Tezi
- Traykovic, M. and Hedley, M., 1998**, Fast Corner Detection, *Image Vision and Computing* 16 (2):75-87
- Rydholm, N., 2015**, Panoramic Video Stitching, Yüksek Lisans Tezi
- Adel, E., Elmogu, M. and Elbakry, H., 2014**, Image Stitching based on Feature Extraction Techniques: A Survey, *International Journal of Computer Applications* (0975-8887)
- Nigar, S.M., Mamesh, 2012**, Image Mosaic Using FAST Corner Detection, *IJARECE* ISSN: 2278-909X
- Muja, M., Lowe, D., 2014**, Scalable Nearest Neighbour Algorithms for High Dimensional Data, *IEEE* 0162-8828



## ÖZGEÇMİŞ

**Fatih DAĞDEVİREN**

**Doğum Tarihi ve Yeri:** 09.06.1989, İzmir/Türkiye

**Uyruk:** Türkiye

**E-mail:** [fatihdagdeviren21@gmail.com](mailto:fatihdagdeviren21@gmail.com)

### **EĞİTİM**

---

**Yüksek Lisans:** Uluslararası Bilgisayar Enstitüsü, Ege Üniversitesi 2012 – 2015.

**Lisans:** Bilgisayar Mühendisliği, İzmir Ekonomi Üniversitesi, 2007 – 2012.

### **ARAŞTIRMA ALANLARI**

---

1. Görüntü işleme
2. Yazılım mühendisliği

### **İŞ TECRÜBESİ**

---

<b>07.01.2013</b>	–	<b>2014</b>	UNIPA	Junior Yazılım Geliştiricisi
(Şubat)				
<b>01.03.2014-16.03.2015</b>			UNIPA	Takım Lideri
<b>06.07.2015- D.E</b>			UNIPA	Kıdemli Yazılım Uzmanı-Yarı Zamanlı



## **EKLER**

Ek 1 İdeal Parametreler Kullanılarak Üzerinde Çalışılan Görüntüler ve İşlenmiş Halleri

## Ek 1 İdeal Parametreler Kullanılarak Üzerinde Çalışılan Görüntüler ve İşlenmiş Halleri



(a)



(b)



(c)



(d)



(e)

Şekil Ek 1 –Seri 2 (a) Referans 1 (b) Karşılaştırılacak 1 (c) PSO eşleme 1 (d) RANSAC 1 (e) Birleştirme Sonucu 1





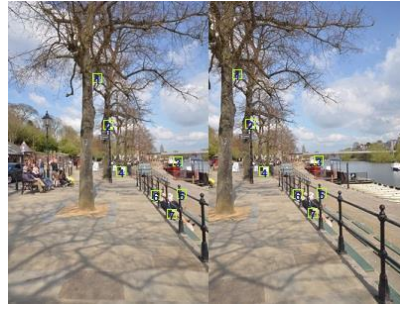
(a)



(b)



(c)



(d)



(e)

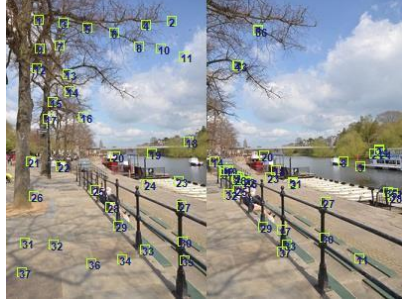
Şekil Ek 2 –Seri 2 (a) Referans 2 (b) Karşılaştırılacak 2 (c) PSO eşleme 2 (d) RANSAC 2 (e) Birleştirme Sonucu 2



(a)



(b)



(c)



(d)



(e)

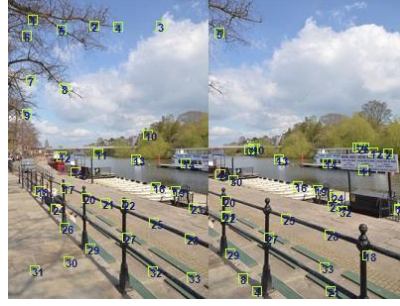
Şekil Ek 3 –Seri 2 (a) Referans 3 (b) Karşılaştırılacak 3 (c) PSO eşleme 3 (d) RANSAC 3 (e)  
Birleştirme Sonucu 3



(a)



(b)



(c)



(d)



(e)

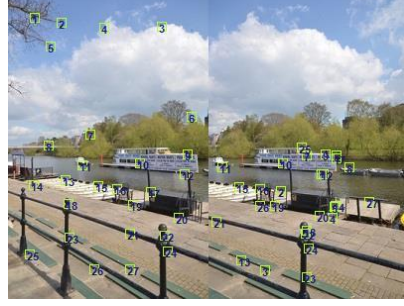
Şekil Ek 4 –Seri 2 (a) Referans 4 (b) Karşılaştırılacak 4 (c) PSO eşleme 4 (d) RANSAC 4 (e) Birleştirme Sonucu 4



(a)



(b)



(c)



(d)



(e)

Şekil Ek 5 –Seri 2 (a) Referans 5 (b) Karşılaştırılacak 5 (c) PSO eşleme 5 (d) RANSAC 5 (e)  
Birleştirme Sonucu 5





(a)



(b)



(c)



(d)



(e)

Şekil Ek 6 –Seri 1 (a) Referans 6 (b) Karşılaştırılacak 6 (c) PSO eşleme 6 (d) RANSAC 6 (e) Birleştirme Sonucu 6



(a)



(b)



(c)



(d)

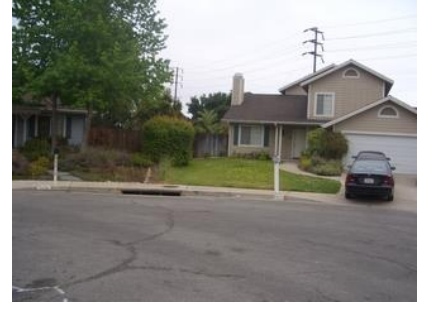


(e)

Şekil Ek 7 –Seri 1 (a) Referans 7 (b) Karşılaştırılacak 7 (c) PSO eşleme 7 (d) RANSAC 7 (e)  
Birleştirme Sonucu 7



(a)



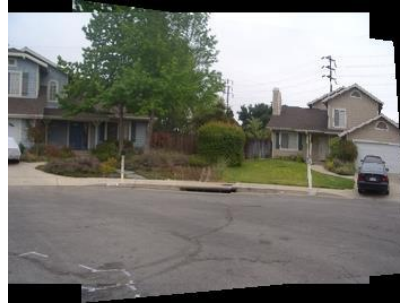
(b)



(c)



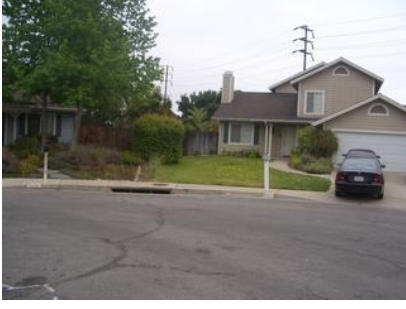
(d)



(e)

Şekil Ek 8 –Seri 1 (a) Referans 8 (b) Karşılaştırılacak 8 (c) PSO eşleme 8 (d) RANSAC 8 (e) Birleştirme Sonucu 8





(a)



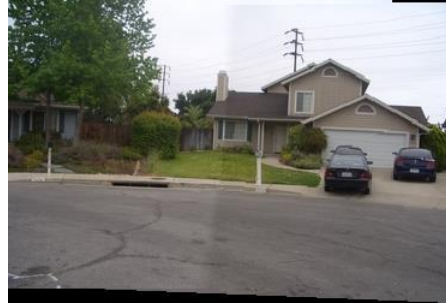
(b)



(c)



(d)



(e)

Şekil Ek 9 –Seri 1 (a) Referans 9 (b) Karşılaştırılacak 9 (c) PSO eşleme 9 (d) RANSAC 9 (e)  
Birleştirme Sonucu 9





(a)



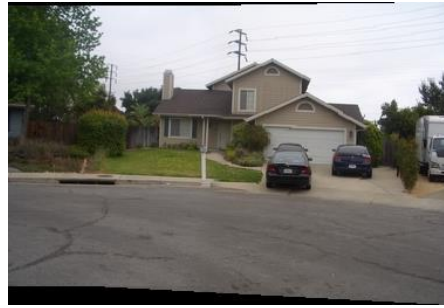
(b)



(c)



(d)



(e)

Şekil Ek 10 –Seri 1 (a) Referans 10 (b) Karşılaştırılacak 10 (c) PSO eşleme 10 (d) RANSAC 10  
(e) Birleştirme Sonucu 10



(a)



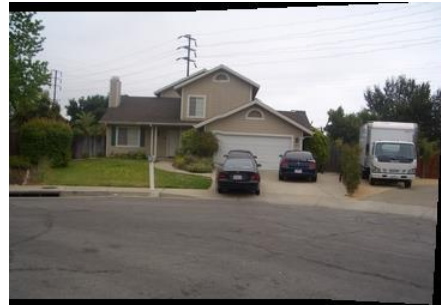
(b)



(c)



(d)



(e)

Şekil Ek 11 –Seri 1 (a) Referans 11 (b) Karşılaştırılacak 11 (c) PSO eşleme 11 (d) RANSAC 11  
(e) Birleştirme Sonucu 11



(a)



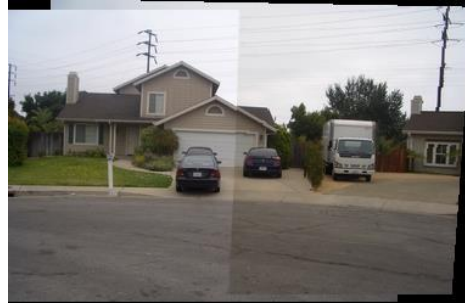
(b)



(c)



(d)



(e)

Şekil Ek 12 –Seri 1 (a) Referans 12 (b) Karşılaştırılacak 12 (c) PSO eşleme 12 (d) RANSAC 12  
(e) Birleştirme Sonucu 12



(a)



(b)



(c)



(d)



(e)

Şekil Ek 13 –Seri 1 (a) Referans 13 (b) Karşılaştırılacak 13 (c) PSO eşleme 13 (d) RANSAC 13  
(e) Birleştirme Sonucu 13





(a)



(b)



(c)



(d)



(e)

Şekil Ek 14 –Seri 1 (a) Referans 14 (b) Karşılaştırılacak 14 (c) PSO eşleme 14 (d) RANSAC 14  
(e) Birleştirme Sonucu 14



(a)



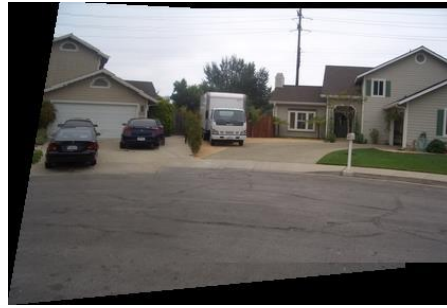
(b)



(c)



(d)



(e)

Şekil Ek 15 –Seri 1 (a) Referans 15 (b) Karşılaştırılacak 15 (c) PSO eşleme 15 (d) RANSAC 15  
(e) Birleştirme Sonucu 15



(a)



(b)



(c)



(d)

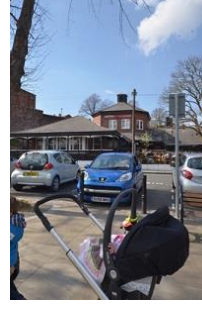


(e)

Şekil Ek 16 –Seri 3 (a) Referans 16 (b) Karşılaştırılacak 16 (c) PSO eşleme 16 (d) RANSAC 16  
(e) Birleştirme Sonucu 16



(a)



(b)



(c)



(d)



(e)

Şekil Ek 17 –Seri 3 (a) Referans 17 (b) Karşılaştırılacak 17 (c) PSO eşleme 17 (d) RANSAC 17  
(e) Birleştirme Sonucu 17





(a)



(b)



(c)



(d)



(e)

Şekil Ek 18 –Seri 3 (a) Referans 1 (b) Karşılaştırılacak 18 (c) PSO eşleme 18 (d) RANSAC 18 (e) Birleştirme Sonucu 18



(a)



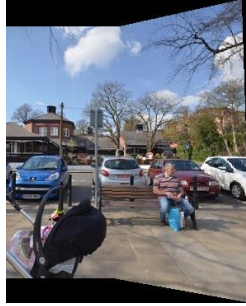
(b)



(c)



(d)



(e)

Şekil Ek 19 –Seri 3 (a) Referans 19 (b) Karşılaştırılacak 19 (c) PSO eşleme 19 (d) RANSAC 19  
(e) Birleştirme Sonucu 19



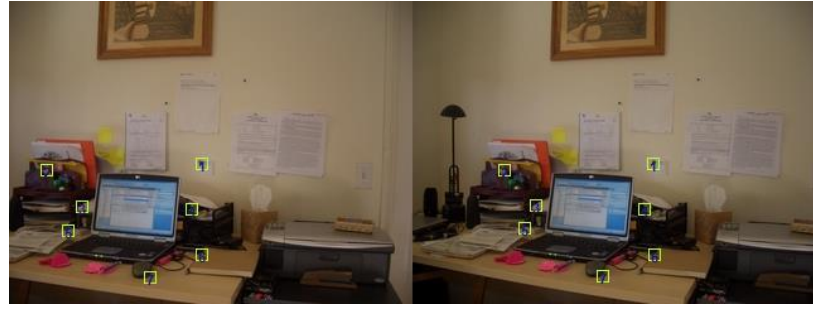
(a)



(b)



(c)



(d)



(e)

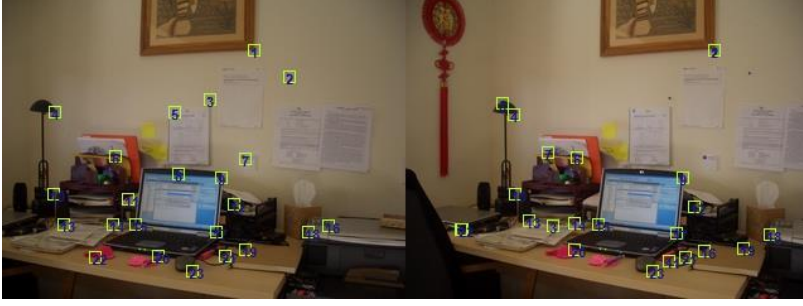
Şekil Ek 20 –Seri 4 (a) Referans 20 (b) Karşılaştırılacak 20 (c) PSO eşleme 20 (d) RANSAC 20  
(e) Birleştirme Sonucu 20



(a)



(b)



(c)



(d)



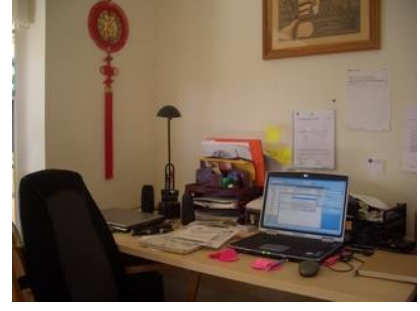
(e)

Şekil Ek 21 –Seri 4 (a) Referans 21 (b) Karşılaştırılacak 21 (c) PSO eşleme 21 (d) RANSAC 21  
(e) Birleştirme Sonucu 21

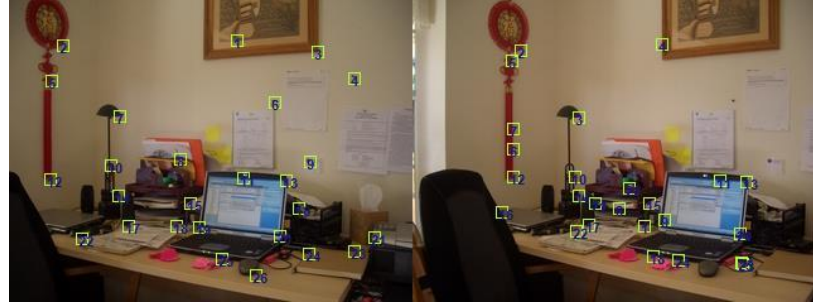




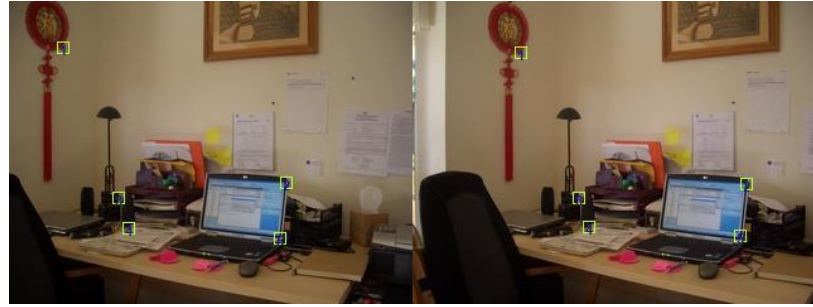
(a)



(b)



(c)

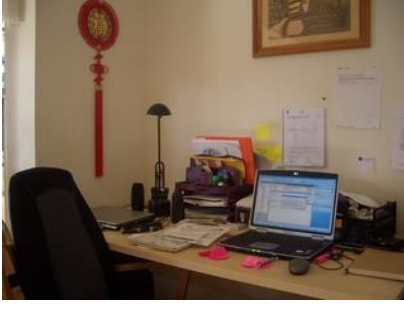


(d)



(e)

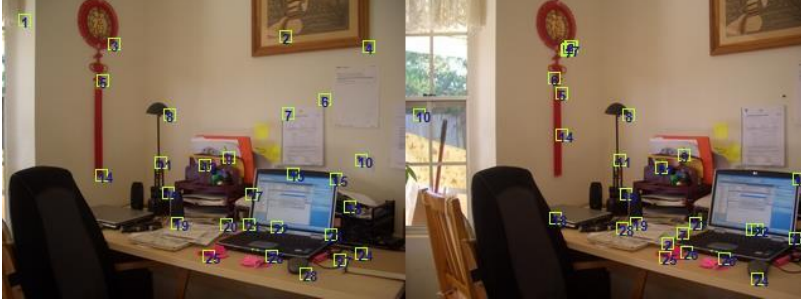
Şekil Ek 22 –Seri 4 (a) Referans 22 (b) Karşılaştırılacak 22 (c) PSO eşleme 22 (d) RANSAC 22  
(e) Birleştirme Sonucu 22



(a)



(b)



(c)



(d)



(e)

Şekil Ek 23 –Seri 4 (a) Referans 23 (b) Karşılaştırılacak 23 (c) PSO eşleme 23 (d) RANSAC 23  
(e) Birleştirme Sonucu 23



(a)



(b)



(c)



(d)



(e)

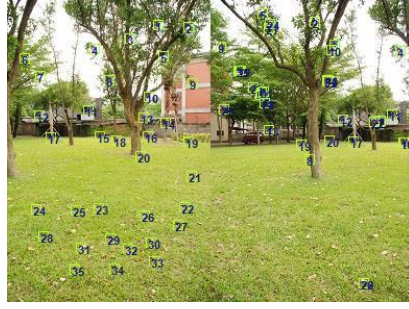
Şekil Ek 24 –Seri 5 (a) Referans 24 (b) Karşılaştırılacak 24 (c) PSO eşleme 24 (d) RANSAC 24  
(e) Birleştirme Sonucu 24



(a)



(b)



(c)



(d)



(e)

Şekil Ek 25 –Seri 5 (a) Referans 25 (b) Karşılaştırılacak 25 (c) PSO eşleme 25 (d) RANSAC 25  
(e) Birleştirme Sonucu 25

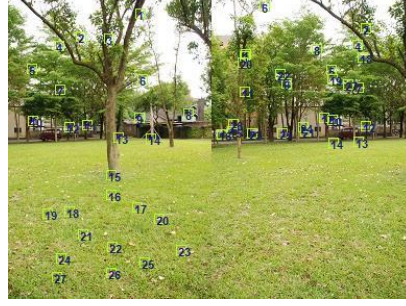




(a)



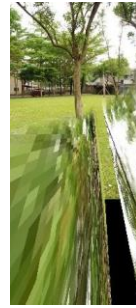
(b)



(c)



(d)



(e)

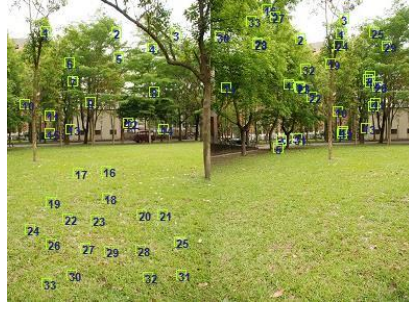
Şekil Ek 26 –Seri 5 (a) Referans 26 (b) Karşılaştırılacak 26 (c) PSO eşleme 26 (d) RANSAC 26  
(e) Birleştirme Sonucu 26



(a)



(b)



(c)



(d)



(e)

Şekil Ek 27 –Seri 5 (a) Referans 27 (b) Karşılaştırılacak 27 (c) PSO eşleme 27 (d) RANSAC 27  
(e) Birleştirme Sonucu 27



(a)



(b)



(c)



(d)



(e)

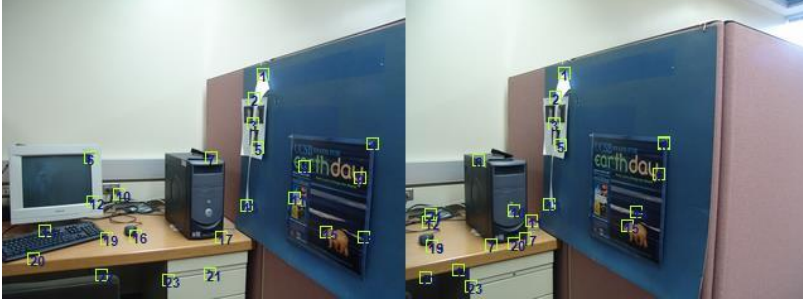
Şekil Ek 28 –Seri 6 (a) Referans 28 (b) Karşılaştırılacak 28 (c) PSO eşleme 28 (d) RANSAC 28  
(e) Birleştirme Sonucu 28



(a)



(b)



(c)



(d)



(e)

Şekil Ek 29 –Seri 6 (a) Referans 29 (b) Karşılaştırılacak 29 (c) PSO eşleme 29 (d) RANSAC 29  
(e) Birleştirme Sonucu 29





(a)



(b)



(c)



(d)



(e)

Şekil Ek 30 –Seri 6 (a) Referans 30 (b) Karşılaştırılacak 30 (c) PSO eşleme 30 (d) RANSAC 30  
(e) Birleştirme Sonucu 30



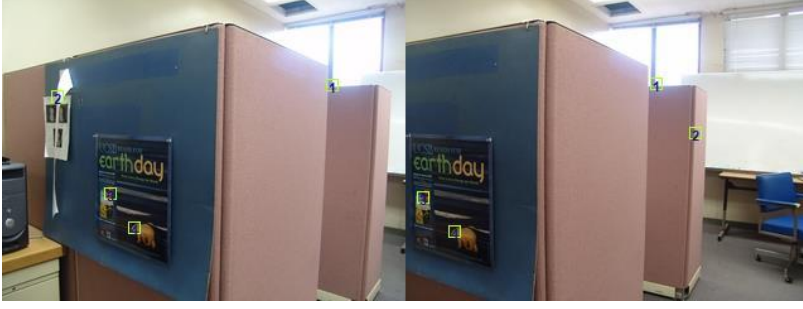
(a)



(b)



(c)



(d)



(e)

Şekil Ek 31 –Seri 6 (a) Referans 31 (b) Karşılaştırılacak 31 (c) PSO eşleme 31 (d) RANSAC 31  
(e) Birleştirme Sonucu 31



(a)



(b)



(c)



(d)



(e)

Şekil Ek 32 –Seri 7 (a) Referans 32 (b) Karşılaştırılacak 32 (c) PSO eşleme 32 (d) RANSAC 32  
(e) Birleştirme Sonucu 32



(a)



(b)



(c)



(d)



(e)

Şekil Ek 33 –Seri 7 (a) Referans 33 (b) Karşılaştırılacak 33 (c) PSO eşleme 33 (d) RANSAC 33  
(e) Birleştirme Sonucu 33





(a)



(b)



(c)



(d)



(e)

Şekil Ek 34 –Seri 7 (a) Referans 34 (b) Karşılaştırılacak 34 (c) PSO eşleme 34 (d) RANSAC 34  
(e) Birleştirme Sonucu 34



(a)



(b)



(c)



(d)



(e)

Şekil Ek 35 –Seri 7 (a) Referans 35 (b) Karşılaştırılacak 35 (c) PSO eşleme 35 (d) RANSAC 35  
(e) Birleştirme Sonucu 35



(a)



(b)



(c)



(d)



(e)

Şekil Ek 36 –Seri 8 (a) Referans 36 (b) Karşılaştırılacak 36 (c) PSO eşleme 36 (d) RANSAC 36 (e) Birleştirme Sonucu 36





(a)



(b)



(c)



(d)



(e)

Şekil Ek 37 –Seri 8 (a) Referans 37 (b) Karşılaştırılacak 37 (c) PSO eşleme 37 (d) RANSAC 37  
(e) Birleştirme Sonucu 37



(a)



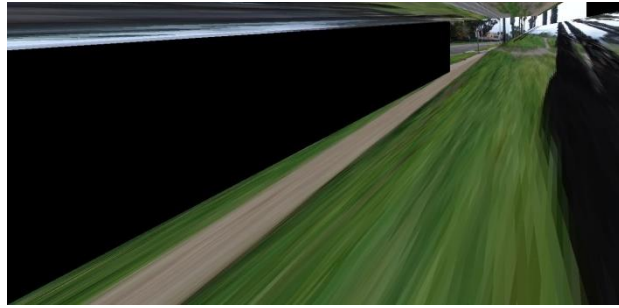
(b)



(c)



(d)



(e)

Şekil Ek 38 –Seri 8 (a) Referans 38 (b) Karşılaştırılacak 38 (c) PSO eşleme 38 (d) RANSAC 38  
(e) Birleştirme Sonucu 38



(a)



(b)



(c)



(d)



(e)

Şekil Ek 39 –Seri 8 (a) Referans 39 (b) Karşılaştırılacak 39 (c) PSO eşleme 39 (d) RANSAC 39  
(e) Birleştirme Sonucu 39

