

**OTOMATİK MOZAIKLEME İLE GENİŞ
YÜZEY ALANLARINA AİT
GÖRÜNTÜLERİN MODELLENMESİ VE
DSP TABANLI GERÇEKLENMESİ**

Mehmet EFENDİOĞLU

**OTOMATİK MOZAIKLEME İLE GENİŞ
YÜZEY ALANLARINA AİT
GÖRÜNTÜLERİN MODELLENMESİ VE
DSP TABANLI GERÇEKLENMESİ**

Mehmet EFENDİOĞLU



T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

OTOMATİK MOZAIKLEME İLE GENİŞ YÜZEY ALANLARINA AİT
GÖRÜNTÜLERİN MODELLENMESİ VE DSP TABANLI GERÇEKLENMESİ

Mehmet EFENDİOĞLU

Yrd. Doç. Dr. Ersen YILMAZ
(Danışman)

YÜKSE LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA-2013

Her Hakkı Saklıdır

TEZ ONAYI

MEHMET EFENDİOĞLU tarafından hazırlanan “OTOMATİK MOZAIKLEME İLE GENİŞ YÜZEY ALANLARINA AİT GÖRÜNTÜLERİN MODELLENMESİ VE DSP TABANLI GERÇEKLENMESİ” adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Yrd. Doç. Dr. Ersen YILMAZ

Başkan: Yrd. Doç. Dr. Ersen YILMAZ İmza
Uludağ Üniversitesi, Mühendislik-Mimarlık
Fakültesi, Elektronik Mühendisliği Anabilim Dalı

Üye: Prof.Dr. Erdoğan DİLAVEROĞLU İmza
Uludağ Üniversitesi, Mühendislik-Mimarlık
Fakültesi, Elektronik Mühendisliği Anabilim Dalı

Üye: Yrd. Doç. Dr. Fatih KARPAT İmza
Uludağ Üniversitesi, Mühendislik-Mimarlık
Fakültesi, Makina Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım / ONAY

Prof. Dr. Kadri ARSLAN
Enstitü Müdürü
... / ... / 2013

U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

04/ 01 /2013

İmza

Mehmet EFENDİOĞLU

ÖZET

Yüksek Lisans Tezi

OTOMATİK MOZAIKLEME İLE GENİŞ YÜZEY ALANLARINA AİT
GÖRÜNTÜLERİN MODELLENMESİ VE DSP TABANLI GERÇEKLENMESİ

Mehmet EFENDİOĞLU

Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Ersen YILMAZ

Bu tez çalışmasında otomatik mozaikleme kullanılarak geniş görüş açısına sahip görüntülerin oluşturulması üzerine yoğunlaşmıştır. Literatürdeki özellik tabanlı birleştirme algoritmaları incelenmiştir. Pratik uygulamalarda sıklıkla tercih edilen algoritmaların yazılım ortamında başarımlarını karşılaştırmaları video, resim dosyası ve gerçek zamanlı veri akışı üzerinden geniş açılı görüntüler oluşturularak gerçekleştirilmiştir.

Anahtar Kelimeler: Panoramik Görüntüleme, Arazi Görüntüleme, Görüntü Dikme, Görüntü Bütünleştirme, Öznitelik Çıkarma, SIFT, SURF, FAST, ORB.

2013, vii + 54 sayfa.

ABSTRACT

Master Thesis

**MODELLING IMAGES OF LARGE SURFACE AREAS BY USING AUTOMATED
MOSAICING AND A DSP BASED IMPLEMENTATION**

Mehmet EFENDİOĞLU

Uludag University
Graduate School of Natural and Applied Sciences
Department of Electronic Engineering

Supervisor: Asst. Prof. Dr. Ersen YILMAZ

In this thesis, we focused on construction of a panoramic image by using automated mosaicing. Feature based registration algorithms in the literature are examined. The performance comparisons in software environment of the algorithms which are oftenly preferred to use in practical applications are realized by generating panoramic images from videos, image files and real time data streams.

Key words: Panoramic Imaging, Landscape Rendering, Image Stitching, Image Blending, Feature Extraction, SIFT, SURF, FAST, ORB.

2013, vii + 54 pages.

TEŞEKKÜR

Çalışmalarında sağladığı katkılar ve olumlu yönlendirmeleri için danışman hocam Yrd. Doç. Dr. Ersen Yılmaz'a, Araş. Gör. Çağlar KILIKÇIER'e ve diğer değerli hocalarıma teşekkürlerimi sunarım.

Başta ailem olmak üzere, tüm çalışma arkadaşlarım ve çalıştığım şirketim YÜKSEL SAVUNMA SİSTEMLERİ'ne bana her anlamda destek verdikleri için minnettarım.

Çalışmalarına yaptıkları önemli katkılar ve ayırdıkları zaman için değerli arkadaşlarım Gürhan YILMAZ, Zeynep BAŞARAN, Salih GÜLER, Serkan KÜRÜZ ve Kemal Özberk USTA'ya teşekkür ediyorum.

Mehmet EFENDİOĞLU
04/01/2013

İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vi
1. GİRİŞ.....	1
1.1. Geniş Yüzey Alanlarına Ait Görüntülerin Oluşturulması.....	1
1.2. Uygulamadaki Amaç ve Temel Sistem Mimarisi.....	2
1.3. Çalışmaların Akışı.....	4
2. KAYNAK ARAŞTIRMASI.....	5
2.1. Öznitelik Noktaları.....	5
2.1.1. SIFT Öznitelik Noktaları.....	7
2.1.2. SURF Öznitelik Noktaları.....	13
2.1.3. FAST Öznitelik Noktaları.....	18
2.1.4. ORB Öznitelik Noktaları.....	22
2.2. İki Boyutlu Hareket Modelleri.....	26
2.3. Dönüşüm Matrisi.....	27
3. MATERYAL VE YÖNTEM.....	32
3.1. Uygulamada Kullanılan Veri Setleri ve Uygulama Ortamı.....	32
3.2. Mozaik Görüntülerin Birleştirilmesi.....	33
3.3. Video Dosyası Üzerinde Otomatik Mozaikleme.....	36
3.3. Dikişsiz Bütünleştirme İşlemi.....	36
3.3.1. Laplaciyan Piramitleri ile Görüntü Bütünleştirme.....	37
3.3.2. Poisson Tabanlı Görüntü Bütünleştirme.....	37
4. BULGULAR VE TARTIŞMA.....	39
5. SONUÇ.....	50
KAYNAKLAR.....	52
ÖZGEÇMİŞ.....	54

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

*	Konvolüsyon toplamı
σ^2	Gauss dağılımının varyansı
Tr()	Matris iz işlemi
Det()	Matris determinant işlemi
t	Koordinatsal öteleme miktarı
θ	Açısal dönme değeri
T	Koordinat dönüşüm matrisi
Mag	Genlik
Ang	Açı

Açıklama

Kısaltmalar

DSP	Sayısal İşaret İşleme
SIFT	Ölçekten Bağımsız Öznitelik Dönüşümü
GOM	Fark Yönlendirmeli Modifikasyon
SURF	Hızlandırılmış Gürbüz Öznitelikler
FAST	Hızlandırılmış Bölüt Test Öznitelikleri
BRIEF	İkili Gürbüz ve Bağımsız Temel Öznitelikler
ORB	Yönlendirilmiş FAST ve Döndürülmüş BRIEF
DoG	Gauss Farkları
SVD	Tekil Değer Ayrıştırma
RANSAC	Rasgele Örnek Onaylama
UAV	İnsansız Hava Aracı
SSD	Karesel Farkların Toplamı
RGB	Kırmızı, Yeşil, Mavi Renk Uzayı
FPS	Saniyedeki Sahne Sayısı
CPU	Merkezi İşlem Birimi
PCI	Çevresel Eklenti Arabağlantısı
GPU	Grafik İşlem Birimi

Açıklama

ŞEKİLLER DİZİNİ

Sayfa

Şekil 1.1. İç mekandan sabit görüş alanı ve yaklaşma ile alınan görüntüler	1
Şekil 1.2. İç mekanın görüntü mozaikleme ile birleştirilmiş görüntüsü	2
Şekil 1.3. Temel uygulama mimarisi	4
Şekil 2.1. İnsansız hava aracından farklı zamanlarda alınan görüntüler üzerindeki öznitelik noktaları (GE© görüntü işleme uygulamaları)	6
Şekil 2.2. Farklı ölçek ve oktavlarda Gauss piramitlerini oluşturan görüntüler	9
Şekil 2.3. Gauss piramitlerinin ve Gauss piramitlerinin ardışıl oktav ve ölçeklerde farklarının oluşturulması (Lowe, 2004)	9
Şekil 2.4. Sabit oktavda farklı ölçekler arasındaki extremum değere sahip anahtar nokta (Lowe, 2004)	10
Şekil 2.5. Öznitelik noktası etrafındaki 16x16 piksellik penceredeki fark vektörleri ve bu vektörlerin her 4x4 pencerede 8 yönlü olacak şekilde hesaplanması (Lowe, 2004)..	13
Şekil 2.6. İntegral görüntü üzerindeki ABCD alanının hesaplanması (Evans, 2009).....	14
Şekil 2.7. Gauss türevleri hesaplamak için kullanılan kutu filtreler (Bay, 2008).....	15
Şekil 2.8. SIFT ve SURF ölçek uzayları (Evans, 2009)	16
Şekil 2.9. x ve y yönünde fark vektörlerini veren Haar dalgacıkları (Bay, 2008).....	17
Şekil 2.10. Tanımlayıcı vektör bileşenleri (Evans, 2009).....	18
Şekil 2.11. Hedef piksel etrafındaki 16 pikselden oluşan ark (Drummond, 2012).....	19
Şekil 2.12. Hedef pikselin testinde istenilen şartı sağlayan ve sağlamayan noktalar (Drummond, 2012).....	20
Şekil 2.13. Öznitelik noktasının etrafındaki 8x8 boyutunda ızgara ve ızgara üzerindeki nicemlenmiş değerler (Drummond, 2012).....	21
Şekil 2.14. Nicemlenmiş değerlerin ikili sayı dizilerine atanması (Drummond, 2012) .	21
Şekil 2.15. Rotasyonel nicemlenmiş yoğunluk alanlarında kullanılan ızgara örüntüsü (Drummond, 2012).....	22
Şekil 2.16. Öznitelik noktasının yönlenme vektörü.....	24
Şekil 2.17. BRIEF tanımlayıcı vektörü için gereken rasgele eşleşme örüntüsü (Rublee, 2011)	25
Şekil 2.18. Tüm eşleşen öznitelik noktaları	31

Şekil 2.19. Doğru eşleşmeye sahip öznitelik noktaları	31
Şekil 3.1. Matrox Morphis sahne yakalama kartı	32
Şekil 3.2. Ardışık iki sahnedeki SIFT noktaları.....	33
Şekil 3.3. Ardışık iki sahnedeki SURF noktaları.....	34
Şekil 3.4. Ardışık iki sahnedeki FAST noktaları	34
Şekil 3.5. Ardışık iki sahnedeki ORB noktaları.....	34
Şekil 3.6. Dönüşüm matrisi yardımı ile birleştirilmiş görüntü	35
Şekil 3.7. Video üzerinden geniş açılı görüntü oluşturma akış şeması.....	36
Şekil 3.8. Poisson tabanlı bütünleştirme yöntemi uygulanmış görüntü.....	38
Şekil 4.1. SURF kullanılarak oluşturulmuş geniş açılı görüntü.....	39
Şekil 4.2. Çalışılan dört farklı yöntem için uygulamada tespit edilen toplam öznitelik nokta sayısı.....	40
Şekil 4.3. Çalışılan dört farklı yöntem için uygulamadaki doğru eşleşme sayısı	41
Şekil 4.4. Çalışılan dört farklı yöntem için her adımdaki toplam işlem süresi.....	41
Şekil 4.5. Poisson tabanlı bütünleştirilmiş görüntü	42
Şekil 4.6. Hava aracından alınan ilk görüntü.....	42
Şekil 4.7. Hava aracından alınan ikinci görüntü	43
Şekil 4.8. SURF yardımı ile hesaplanan hareket vektörleri.....	43
Şekil 4.9. Örnek UAV videosundan oluşturulan geniş açılı görüntü.....	44
Şekil 4.10. UAV videosundaki çalışmada dört farklı yöntemin tespit ettiği toplam öznitelik noktaları.....	45
Şekil 4.11. UAV videosundaki çalışmada dört farklı yöntemin doğru eşleşme performansı	45
Şekil 4.12. UAV videosunda uygulanan dört farklı yöntemin zaman performansı.....	46
Şekil 4.13. Rotasyonel görüntü alan UAV videosunda ilk ve son sahneler	47
Şekil 4.14. Video üzerinden birleştirilmiş görüntü.....	47
Şekil 4.15. Rotasyonel değişimin olduğu video üzerinden birleştirilmiş görüntüde dört yöntemin tespit ettikleri toplam öznitelik noktası sayısı.....	48
Şekil 4.16. Rotasyonel değişimin olduğu video üzerinden birleştirilmiş görüntüde dört yöntemin doğru eşleşme performansı	48
Şekil 4.17. Rotasyonel değişimin olduğu video üzerinden birleştirilmiş görüntüde dört yöntemin zaman performansı.....	49

1. GİRİŞ

1.1. Geniş Yüzey Alanlarına Ait Görüntülerin Oluşturulması

Günümüz sayısal görüntüleme sistemlerinin temel gereksinimlerinden bir tanesi, görüntülenecek sahneden olabildiğince fazla bilgi ve detayı elde etmektir. Buna karşın gelecek olan bilgi bant genişliği, çözünürlük, görüş alanı gibi bir takım fiziksel kriterler dahilinde daima sınırlanmaktadır ve her sistemde bu sınırlar bazı takaslar yaratmaktadır. Bu sınır ve takaslar görüntülenecek alan genişledikçe daha dikkat çekici bir şekilde sistem mimarilerini etkilemektedir.

Geniş yüzey alanlarının görüntülemek isteyen sistemlerdeki en göze çarpıcı takas yaklaşma ve görüş alanı arasındaki orantıdır. Bir alanda daha fazla detayı görmek için optik olarak yaklaşma yapıldığında görüntüleme sisteminin görüş alanı yatay ve dikeyde açısal olarak azalır. Tersisi durumda ise geniş alanı görüntülememize rağmen, elde edebileceğimiz detay miktarı sınırlı kalmaktadır. Bu takas göz önüne alındığında sabit bir yaklaşma - dolayısıyla sabit bir görüş alanında - görüntüleme sensöründen alınacak bilgiyi bütünleştirmeye yönelik en elverişli yöntem mozaik görüntülerdir.



Şekil 1.1. İç mekandan sabit görüş alanı ve yaklaşma ile alınan görüntüler

Şekil 1.1'deki görüntü, kapalı bir iç mekandan alınan 128 adet görüntüden sadece 8 tanesini göstermektedir. Görüldüğü gibi, bir iç mekandan sabit yaklaşma ve görüş alanı ile alınan bilgiler her sahnede sınırlıdır. Bu sahnelerin (Halcon ortamında) uygun

biçimde birleştirilmesi ile oluşan görüntüde bilgi daha bütünseldir ve her sahnenin detayını ayrı ayrı içermektedir. Görüntü mozaikleme ile birleştirilmiş görüntü şekil 1.2’de gösterilmiştir.



Şekil 1.2. İç mekanın görüntü mozaikleme ile birleştirilmiş görüntüsü (MVTec Halcon)

Mozaik görüntüler bir görüntüleme sensöründen istenilen bir alan üzerinden belirli düzende görüntülerin alınarak, bu görüntülerin birbirleriyle uyumlu bir şekilde geometrik olarak birleştirilmesi ile oluşturulan görüntülerdir. Mozaik görüntüleri oluşturan veriler pek çok farklı kaynak ve biçimde sisteme gelebilir. İster gerçek zamanlı veri akışı üzerinden, isterse gerçek zamanlı olmayan bir veri kümesi üzerinden olsun mozaik görüntülerin oluşturulmasındaki en temel yöntem birleştirilecek görüntüler arasındaki dönüşüm matrisinin hesaplanmasıdır. Dönüşüm matrisi her birleştirme işleminde görüntüler arasındaki geometrik ilişkiyi veren en önemli parametrelerdir ve hesaplama hızı sistem kalitesini doğrudan etkilemektedir.

1.2. Uygulamadaki Amaç ve Temel Sistem Mimarisi

Bu tez çalışmasındaki asıl amaç otomatik görüntü mozaikleme uygulamalarını farklı veri kaynakları ve kullanılacak algoritmalar bazında uygulamaya yönelik sonuçların karşılaştırılmalarıdır. Bunun için ilk olarak otomatik mozaikleme uygulamalarındaki literatürde yer alan yöntemler incelenmiştir. Otomatik mozaiklemede kullanılan yöntemler genel olarak öznitelik tabanlı yöntemler ve öznitelik tabanlı olmayan yöntemler olacak şekilde ikiye ayrılırlar. Özellikle öznitelik tabanlı yöntemlerin

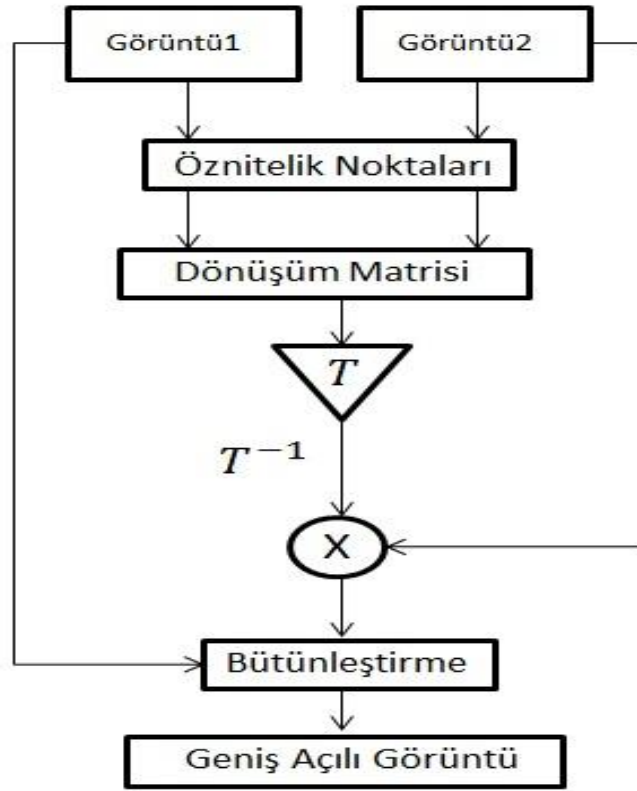
incelendiđi bu tezde görüntüler üzerindeki öznitelik noktalarının tespiti, bu noktaların eşleştirilmeleri için kullanılan algoritmalarından uygulamalara elverişli olanlar incelenmiş ve birbirlerine olan üstün yönleri incelenmiştir.

Bir sonraki adımda görüntüler arası dönüşüm matrisinin hesaplanması yer almaktadır. Dönüşüm matrisi, eklenecek görüntüyü geometrik olarak ana görüntüye uygun bir biçime dönüştürecek hareket modelinin parametrelerini içermektedir. Bu matris sayesinde birleştirme işlemi görüntülerde bozulma olmadan, ana sahneye uygun bir biçimde gerçekleşmektedir.

Dikilen görüntüler arasındaki geçiş bölgesinde, izdüşüm farklılığı, kontrast değişimi gibi sebeplerden dolayı dikme yerlerinde lineer olmayan bozulmalar meydana gelmektedir. Bu problemin çözümünde Laplaciye piramitleri ve Poisson tabanlı bütünleştirme algoritmaları kullanılmaktadır. Bu sayede birleşme bölgelerindeki geçiş daha uygun bir biçimde görüntüye yansımaktadır.

Veri kaynakları olarak çalışmalarda önce resim ve video kullanılmış, daha sonra sahne yakalama donanımı ile alınan gerçek zamanlı video verisi kullanılmış ve gerçek zamanlı uygulamada karşılaşılan problemler incelenmiş ve uygun bir mimari sunulmuştur.

Uygulama adımlarındaki sistem mimarisi Şekil 1.3'te gösterilmiştir. Görüldüğü gibi ilk adım olarak 2 görüntü verisi üzerinde öznitelikleri çıkartma işlemi yapılmaktadır. Daha sonraki adımda bu öznitelik noktaları yardımı ile görüntüler arasındaki geometrik dönüşüm matrisi T hesaplanarak, görüntü 2 bu dönüşüme uygun bir biçimde değiştirilir. Ana resim görüntü 1 üzerine eklendikten sonra bütünleştirme işlemi ile birleştirme bölgesindeki bozulma giderilir ve geniş açılı görüntü elde edilmiş olur.



Şekil 1.3. Temel uygulama mimarisi

1.3. Çalışmaların Akışı

Tezin ilerleyen bölümleri şu şekilde organize edilmiştir: İkinci bölümde otomatik mozaikleme ile ilgili literatürde yer alan çalışmalar incelenmiştir. Özellikle öznitelik tabanlı yöntemler üzerinde yoğunlaşmış olup, bu yöntemlerden SIFT, SURF, FAST ve ORB yöntemleri ile görüntü üzerindeki öznitelik noktalarının tespiti ve tanımlanması üzerinde durulmuştur. Daha sonra bu noktaların eşleştirilmeleri ve görüntüler arası hareket modeli incelenmiş, dönüşüm matrisinin hesaplanmasından bahsedilmiştir. Üçüncü bölümde mozaik görüntülerin oluşturulması işlemi yer almaktadır. Birleştirme esnasında görüntü geçişini düzenleme işlemi olarak “bütünleştirme işlemleri” yine bu başlık altında incelenmiştir. Dördüncü bölümde araştırmada uyguladığımız yöntemlere ait elde ettiğimiz bulgular ve yaptığımız benzetimlerin detayları yer almaktadır. Beşinci bölümde de yaptığımız deneyler ve benzetimler üzerinde ulaştığımız sonuçlardan ve değerlendirmelerden bahsedilmektedir.

2. KAYNAK ARAŞTIRMASI

Otomatik mozaikleme konusunda kullanılan yöntemler literatürde öznitelik tabanlı ve özniteliksiz olmak üzere ikiye ayrılmaktadır.

Özniteliksiz mozaikleme yöntemleri görüntüler üzerinde daha bütünsel analizler yapar. Ramachandran ve Chellappa' nın (2006) çalışmasında olduğu gibi görüntüde hareket modeli, faz korelasyonu kullanılarak hesaplanabilir. Diğer yandan Irani ve Anandand (1999) çalışmalarında giriş görüntüsü ve uygun modelden oluşturduğu Laplaciens piramidlerini kullanan bir algoritma oluşturmuş, SSD fonksiyonunu minimize ederek iki görüntü arasındaki dönüşümü hesaplamıştır. Tüm bu çalışmalarla birlikte özniteliksiz mozaikleme çalışmaları görüntüdeki her piksel üzerinde işlem yaptığından hesaplama maliyeti yüksek yöntemlerdir. Ayrıca görüntüdeki kontrast değişimine ve gürültüye, öznitelik tabanlı yöntemlerden daha hassas olduklarından uygulamalarda daha az tercih edilirler.

Öznitelik tabanlı yöntemlerde ise uygulanan ilk adım, görüntü üzerinde pek çok zorlayıcı şarta ve değişime (kontrast değişimi, ölçek farkı, rotasyonel değişimler vb.) direnç gösteren ve gürbüz sayılabilecek noktaların tespitidir. Görüntü dikme uygulamalarının doğası gereği, bir görüntüde tespit edilen öznitelik noktaların bu görüntüye ait belirli bir kısım içeren diğer görüntülerde de – kesişim bölgesinde – aynı şekilde tespit edilebilmesi beklenir. Bu sayede tespit edilen noktalar eşleştirilerek iki görüntü arasındaki geometrik ilişki matematiksel olarak modellenabilir. Bu modelleme işlemi genel olarak görüntü hareketini ve üç boyutlu uzaya ait koordinatların iki boyuttaki izdüşümünü hesaplamak için matematiksel olarak yardımcı olur.

2.1. Öznitelik Noktaları

Her pikseli işleyen yöntemlerin getirdiği hesaplama yükü, öznitelik noktalarının kullanıldığı yöntemlerin tercih edilmesi ile aşılabilmektedir. Bir görüntüde bulunan ideal bir öznitelik noktasının en önemli özelliği global bir benzerliğe sahip olmasıdır, dolayısıyla aynı nokta pek çok farklı şartlar altında tekrar tespit edilerek gerekli

eşleştirme yapılabilmelidir. Şekil 2.1’de görüldüğü gibi, bir hava aracından alınan görüntüde öznitelik noktaları gösterilmektedir. Yaklaşık 5 saniye sonra alınan ikinci görüntüde de fiziksel olarak aynı koordinatlara denk gelen öznitelik noktaları tekrar tespit edilmiştir. Hareket modeli için yeterli sayıda noktanın şekildeki gibi başarılı şekilde tespiti istenen bir durumdur ancak pek çok farklı otomatik mozaikleme uygulamasında bu konuda farklı ihtiyaçlar doğmaktadır. Bazı uygulamalar mümkün oldukça çok eşleşen nokta tespitine ihtiyaç duyarken, bazı uygulamalarda ise öznitelik noktalarının tespit ve eşleşme hızı daha önemlidir. Bu nedenle literatürde artı ve eksi yönleri olan pek çok öznitelik noktası tespit yöntemi yer almaktadır. Ayrıca görüntü eşleştirme, hedef tanıma ve takibi, stereo görüntü birleştirme gibi uygulamaların da yaygınlaşmasıyla öznitelik noktalarının tespiti daha popüler bir araştırma konusu haline gelmiştir.



Şekil 2.1. İnsansız hava aracından farklı zamanlarda alınan görüntüler üzerindeki öznitelik noktaları (GE© görüntü işleme uygulamaları)

Görüntüde hareket kestirimi konusunda en önemli bilgiyi veren değerlerden bir tanesi, görüntünün Hessian matrisinin özdeğerleridir. Bu nedenle pek çok yüksek doğruluklu öznitelik tespiti algoritması Hessian matrisi üzerinden hesaplama yapmaktadır. Harris ve Stephens (1988), Gauss ağırlıklandırılmış fonksiyonları kullanarak filtreleme ve matris işlemleri ile Hessian matrisini hesaplamışlardır. Harris öznitelik noktaları pek çok uygulamada hızlı sonuç verdiği için tercih edilen bir uygulamadır, ancak bu

noktalar ölçeğe karşı oldukça duyarlıdır ve pek çok gerçek problemde tatmin edici sonuç verememektedir. Farklı bir yaklaşım olarak Shi ve Tomasi (1994) en küçük özdeğerler yöntemini kullanarak öznitelik noktalarını, kendi tabirleri ile “takip için uygun noktalar” tespit etmişlerdir. Hızlı bir şekilde öznitelik tespit eden bu yöntem hızlı parlaklık değişimine dirençli değildir. Noktaların eşleştirilmeleri için çoğunlukla optik akışın tercih edilmeside bu sorunun dahada ciddileşmesine sebep olmaktadır. Daha gelişmiş bir yöntem olan SIFT algoritmasında Lowe (1999) öznitelik noktalarını görüntünün farklı ölçek ve oktavlarda Gauss piramitlerinin farkı üzerinde hesaplamış ve Hessian matrisini kenarlar üzerindeki noktaların eliminasyonu için kullanmıştır. Kendi tanımlayıcı vektörü oluşturma algoritmasında öneren SIFT, işlem yükü oldukça ağır olsada pek çok şarta dayanıklı öznitelik noktaları sağlayan bir algoritmadır. SIFT üzerindeki hız problemini Bay ve arkadaşları (2006) SURF algoritmasında Hessian matrisinin determinantını, integral görüntüler üzerinde farklı kutu filtreler kullanıp hesaplayarak gidermeye çalışmıştır.

Her ne kadar SIFT ve SURF pek çok şarta karşı dayanıklı olsalar da, gerçek zamanlı uygulamalarda yöntemlerin sonuç hızı, doğruluk derecelerinden çok daha önemlidir. Dolayısıyla öznitelik noktalarının tespit ve eşleştirmelerini daha hızlı yapabilmek için de çeşitli araştırmalar mevcuttur. Rosten ve Drummond (2006) ise FAST adını verdikleri yöntemde, görüntüde lokal bölgelerdeki maksimum değere sahip noktaları hızlandırılmış segment testi tabanlı bir yöntem ile tespit edip belirlemiştir. Tespit ettiği noktalar SIFT kadar tutarlı olmasada oldukça hızlı işlem yapma kapasitesi FAST’ın en önemli artısıdır. Ruble ve arkadaşları (2011) FAST ve BRIEF (Calonder ve arkadaşları, 2010) yöntemlerinden faydalanarak öznitelik tespiti için hızlı ve rotasyona dayanıklı bir yöntem olan ORB yöntemini geliştirmişlerdir.

2.1.1. SIFT Öznitelik Noktaları

Görüntü mozaiklerinin otomatik olarak birleştirilmesinde SIFT noktalarının eşleştirilmeleri, fiziksel koşullara karşı uygun olduklarından sıklıkla kullanılan bir metottur. Referans görüntü ve birleştirilecek görüntü arasındaki SIFT noktaları tespit edilip eşleştirildikten sonra – gelecek görüntüler için parlaklık, ölçek ve dönme (

rotasyon) açısından bir sınırlama getirilmediyse - görüntü çiftleri arasındaki benzerlik hesaplanmalıdır.

SIFT pratik uygulamada görüntü üzerinde oldukça yüksek ayırt ediciliğe sahip çözünürlükte öznelik noktaları vermektedir. Temel olarak gri seviye görüntüler üzerindeki öznelik noktalarının tespiti üzerine geliştirilmiş bir çalışmadır. Anahtar noktaların görüntü üzerindeki koordinatlarının tespit edilmesi, sağlam olmayan noktaların (kenar üzerindeki noktalar gibi) kümeden çıkartılması ve bu noktalara tanımlayıcı vektörler atanması şeklinde 3 adımdan oluşur.

SIFT metodunda anahtar noktaların tespit edilmesi, gri seviye görüntünün farklı ölçek ve oktavlardaki Gauss piramidi farklarından oluşturulan ölçek uzayı üzerindeki ekstremum noktaların belirlenmesi esasına dayanır (Burt ve Adelson 1983) (Crowley ve Stern 1984).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.1)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.2)$$

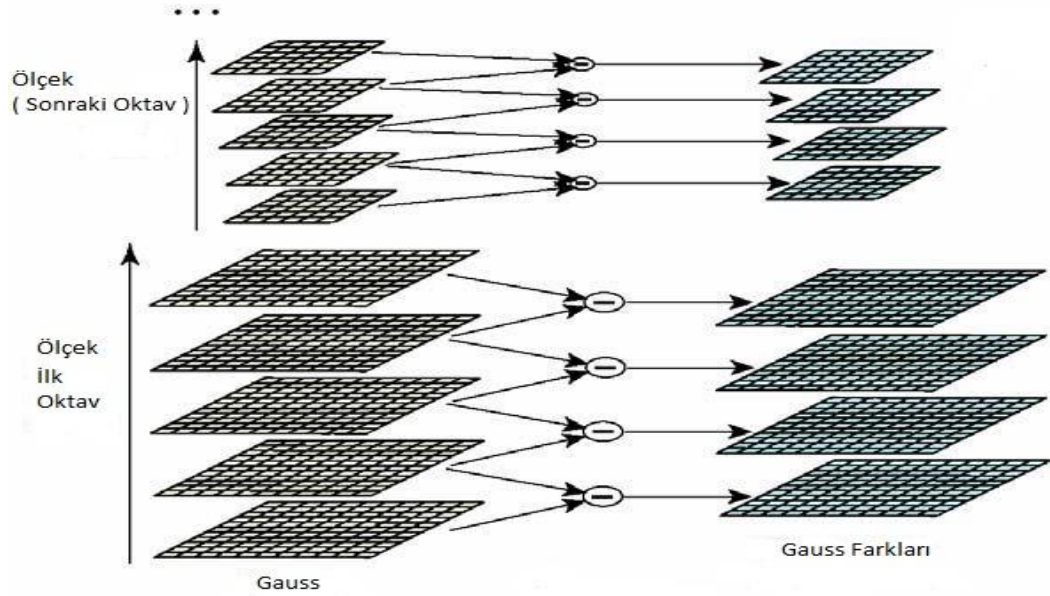
$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (2.3) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

Bir görüntünün Gauss piramidi denklem 2.1 de görüldüğü üzere, önce görüntüyü 2.2 şeklinde tanımlı olan Gauss filtre ile keskin geçişlerini yumuşatarak daha sonrada görüntüyü istenilen oranda alt-örnekleyerek oluşturulur. Örnek olarak farklı ölçek ve oktavlarda Gauss piramidi oluşturulmuş görüntüler dizisi Şekil 2.2'de gösterilmiştir.



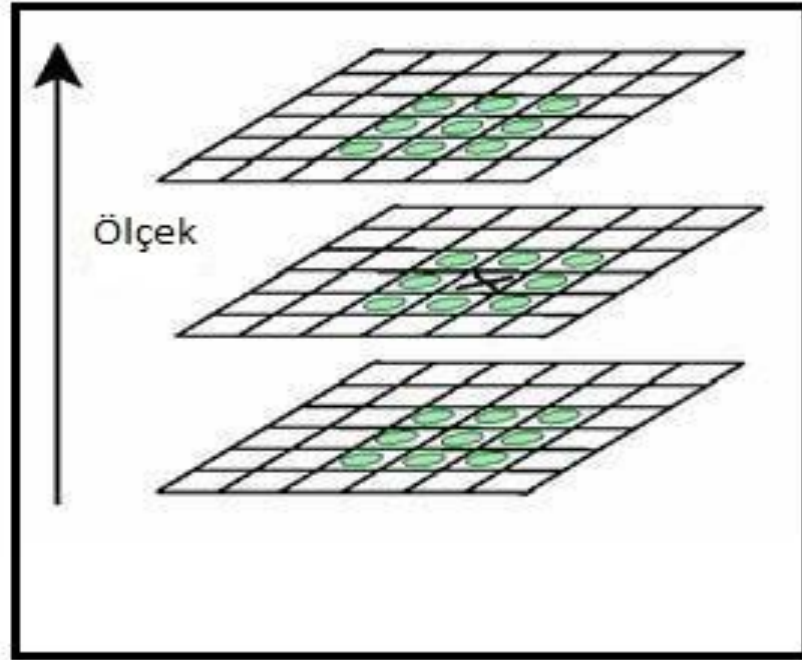
Şekil 2.2. Farklı ölçek ve oktavlarda Gauss piramitlerini oluşturan görüntüler

Gauss piramitlerinin farkları da aynı oktavdaki, komşu seviyelerdeki görüntülerin birbirlerinden çıkartılmaları sonucu oluşturulurlar (2.3). Bu işlemler Şekil 2.3’de gösterilmiştir.



Şekil 2.3. Gauss piramitlerinin ve Gauss piramitlerinin ardışıl oktav ve ölçeklerde farklarının oluşturulması (Lowe, 2004)

Anahtar noktalar Gauss farkları uzayındaki koordinatlarda hem de piramitteki ölçek seviyesindeki ekstremum noktalar sayesinde tespit edilirler. Hedef ölçekteki ekstremum noktanın tespiti için; seçilen nokta, komşuluğundaki 8 piksel ile karşılaştırılır. Daha sonra alt ve üst ölçeklerdeki aynı koordinatlara karşılık gelen piksellerle karşılaştırılır. Piksellerin sabit oktavda karşılaştırma işlemi Şekil 2.4'te gösterilmiştir. Eğer piksel değeri tüm bu noktaların değerlerinden büyükse ya da küçükse öznitelik noktası olarak belirlenir. Görüntüde herhangi bir ölçekte, birbirine yakın birden fazla aday anahtar nokta yer alabilir. Gauss filtrelemenin amaçlarından bir tanesi de yüksek bileşenlerini filtreleyerek birbirlerine yakın olan noktaları, aday nokta kümesinden çıkartmaktır.



Şekil 2.4. Sabit oktavda farklı ölçekler arasındaki ekstremum değere sahip anahtar nokta (Lowe, 2004)

Bir sonraki adımda düşük kontrasta sahip noktaların, anahtar noktalar kümesinden eleme işlemi yer almaktadır. Düşük kontrast noktalarını, parlaklık değişimi gösteren farklı görüntülerde daha farklı değer aralığına sahip olduklarından daha az durağandır. Brown ve Lowe (2002) ölçek uzayı fonksiyonunun Taylor serisi açılımı

yardımı ile bu fonksiyon üzerindeki extremum noktaları hesaplamış, belirli eşik değerinin altındaki noktaları elemişlerdir. Dolayısıyla bu noktaların çıkartılması eşleştirme aşamasında geriye daha sağlam anahtar noktaların kalmasını sağlayacaktır.

Gauss farklarının kullanılması, kenar noktadaki üzerindeki anahtar noktaların tespitini arttırmaktadır. Dolayısıyla bu noktalar üzerindeki en ufak bir gürültü, anahtar noktanın değerini ciddi bir biçimde etkilemektedir. Bundan dolayı kenar noktalar üzerindeki anahtar noktalar da, sağlam eşleştirme için anahtar noktalar kümesinden çıkartılabilir.

Lowe (1999, 2004) bu noktaların elenmesi için Hessian matrisinin özdeğerleri arasındaki oranı kullanmıştır.

$$H(I(x, y)) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial y \partial x} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.4)$$

$$\text{Tr}(H) = D_{xx} + D_{yy} \quad (2.5)$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 \quad (2.6)$$

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(r+1)^2}{r} \quad (2.7)$$

Lowe'nin 2.7 denkleminde r ($r \geq 1$) büyük ve küçük özdeğerler arasındaki üst limittir ve r değeri arttıkça takip edilen kenarın uzunluğunda artacaktır. Bu durumda r değeri istenilen hassasiyete getirilerek kenar eleme işlemi yapılabilir.

Her anahtar noktanın kendine ait bir veya birden fazla yönlenme vektörü vardır. Bu vektör değerleri sayesinde noktalar, farklı görüntülerde farklı rotasyonlara sahip olsalar bile tespit edilmeleri sağlanabilecektir. Bu yönlenme vektörlerinin hesaplanması için öncelikle Gauss filtreden geçmiş görüntülerin fark vektörleri çıkartılmalıdır.

Görüntüdeki bir (x, y) koordinatına ait fark vektörünün genlik ve açı değeri sırasıyla 2.8 ve 2.9 da verilmiştir.

$$\text{Mag}(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.8)$$

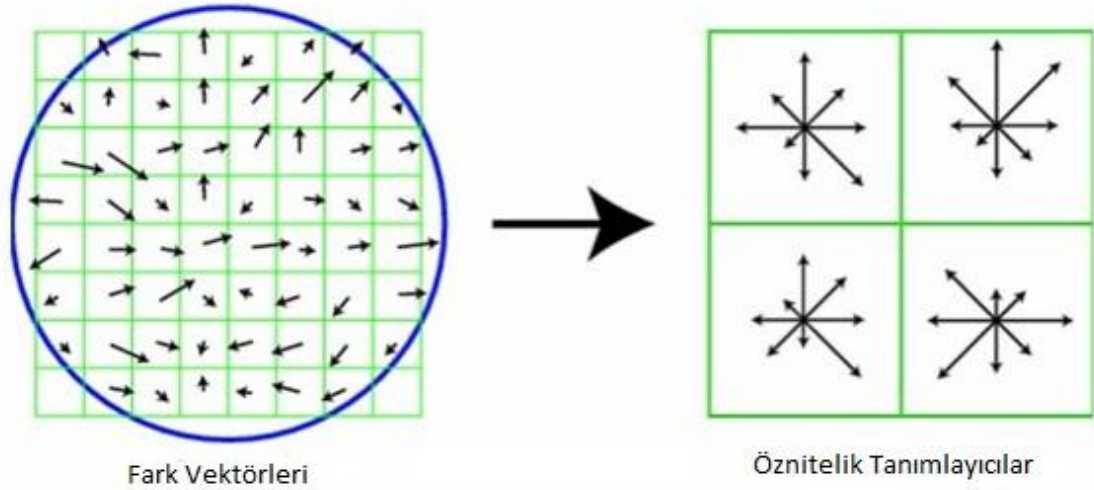
$$\text{Ang}(x, y) = \text{atan}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \quad (2.9)$$

Yönlenmede doğruluğu arttırmak için her açı vektörü 10 ar derecelik dilimler halinde 36 parçada toplanırlar. 36 parçalı açı bölgesine yerleştirilmeden önce tüm değerler, fark vektörlerinin genlikleri ile ağırlıklandırılırlar. Daha sonra (σ değeri 1.5 ile çarpılarak) Gauss pencereden geçirilerek açı bölgesine yerleştirilirler. Bu nokta SIFT yönteminin uygulamasını çeşitlendiren noktalardan biridir. Örnek vermek gerekirse SIFT'in uygulama performansını arttırmak için geliştirilmiş olan GOM – SIFT Yi ve arkadaşları (2008) tarafından geliştirilmiştir. SIFT' ten farklı olarak anahtar nokta yönlenmesini 0 – 180 derece arasında sınırlayarak yönlenme kümesini daraltır ve eşleşme esnasında yöntemin daha hızlı çalışmasını sağlar.

Belirlenen ölçekte, fark vektörleri dağılımı hesaplandıktan sonra baskın vektör, yönlenme vektörü olarak seçilebilir. Fark vektörleri dağılımındaki tepe noktaların değerlerine, tepe noktasının maksimum değerinin %80 ine kadar olan değere sahip diğer vektörlerde yönlenme vektörü olarak kaydedilebilirler.

Son adımda tespit edilen her noktaya karşılık gelen tanımlayıcı vektör hesaplanır. Bu vektörün hesaplanmasında bir önceki adımda elde edilen fark vektörlerinin öznitelik noktası etrafında Şekil 2.5'te gösterildiği gibi 16x16 piksellik pencere içerisindeki dağılımına bakılır. Bu dağılım daha sonra 4x4 boyutunda pencereye indirgenerek yönlenme vektörleri hesaplanır. Her hücrede, önceki adımdaki gibi 8 bölümlük fark vektörü histogramı hesaplanarak vektörler atanır. Dolayısıyla her öznitelik noktasına eşlik eden 128 boyutlu bir dizi hesaplanmış olur. Bu dizi anahtar noktanın tanımlayıcı

dizisidir ve diğer anahtar noktalardan istatistiksel olarak ayrılmasını sağlayan değerler bu dizi içerisinde yer almaktadır. Dolayısıyla daha sonra bu değerler dizisi, diğer noktalara ait dizilerle Euclid mesafesi kullanılarak yakınlıkları hesaplanabilir.



Şekil 2.5. Öznitelik noktası etrafındaki 16x16 piksellik penceredeki fark vektörleri ve bu vektörlerin her 4x4 pencerede 8 yönlü olacak şekilde hesaplanması (Lowe, 2004)

2.1.2. SURF Öznitelik Noktaları

SURF noktaları, pek çok uygulamada ölçeksel ve rotasyonel dönüşümlere dayanıklı oldukları bilinmektedir. Ancak SURF noktalarının kullanılmasındaki asıl amaç, SIFT noktalarının tespitindeki işlem yükünün getirdiği dezavantajı gidermektir. SURF noktalarının tespiti temel algoritmik adımlar olarak SIFT ile benzer bir yapı göstermektedir. Hessian matrisi üzerinden anahtar nokta tespiti yapılmakta ancak matrisin hesaplanmasını Haar dalgacıkları kullanarak büyük bir yaklaşıklıkla daha hızlı bir biçimde yapar. Haar dalgacıklarını hızlı bir şekilde uygulamak içinde integral görüntülerden yararlanır.

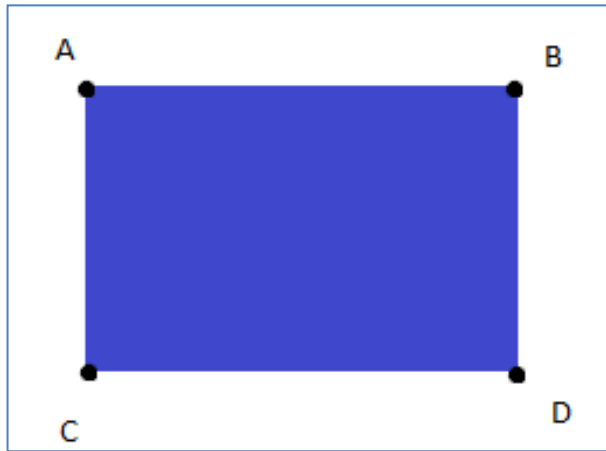
İntegral görüntüler görüntü işleme uygulamalarında sıklıkla kullanılan bir veri tipidir. (Viola ve Jones 2003) algoritması gibi Haar özelliklerinin incelenmesi gerektiği

durumlarda dikdörtgen alanların hızlı bir şekilde hesaplanmasına önemli bir katkı sağlar. SURF algoritmasında dalgacık analizi için integral görüntülerden faydalanılmaktadır. Örnek bir I giriş görüntüsünün (x,y) koordinatındaki integralini hesaplamak için 2.10 formülü kullanılabilir.

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (2.10)$$

Şekil 2.6'da gösterildiği gibi bir görüntü içerisindeki A,B,C ve D köşelerine sahip dikdörtgensel bölgenin alanı integral görüntü üzerinden köşe değerlerini kullanarak formül 2.11 yardımı ile hesaplanabilir.

$$\sum [ABCD] = A + D - (B + C) \quad (2.11)$$



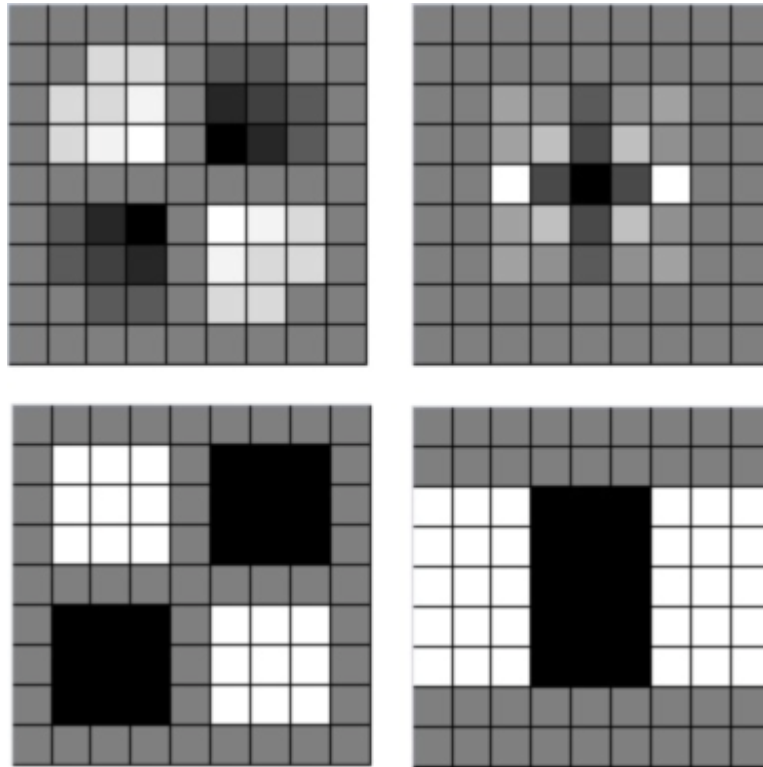
Şekil 2.6. İntegral görüntü üzerindeki ABCD alanının hesaplanması (Evans, 2009)

İntegral görüntü bir kere hesaplandıktan sonra, istenilen alanın yoğunluğu, tüm piksel değerlerini tek tek toplamak yerine formül 2.11 deki gibi basit bir toplama işlemiyle hızlı bir şekilde yapılmaktadır. Görüntünün integralini almak görüntü boyutuna bağlı olsa da, integral görüntü üzerinden bir alanın yoğunluğunu hesaplamak görüntü boyutundan bağımsızdır.

Bölümün başında bahsettiğimiz gibi SURF, Hessian matrisindeki Gauss fonksiyonunun türevlerini Gauss farkları yerine kutu filtreler kullanarak yaklaşık olarak hesaplar. Şekil 2.7’de gösterilen bu filtrelerde; 2. satırdaki filtreler D_{xx} , D_{yy} ve D_{xy} türevlerini hesaplamak için kullanılmaktadır. D_{xy} türevi için siyah pikseller +1 beyaz pikseller -1 ile ağırlıklandırılmıştır. Gri piksellerde ağırlıklandırma yoktur. D_{xx} ve D_{yy} türevi için siyah pikseller +2, beyaz pikseller -1 ile ağırlıklandırılmıştır. Birinci satırdaki filtreler de xy ve x yönündeki 2. derece türevleri hesaplamak için kullanılmaktadırlar. Bu filtreler sayesinde Hessian matrisinin determinantı formül 2.12 ile büyük bir yaklaşıklıkla hesaplanabilir.

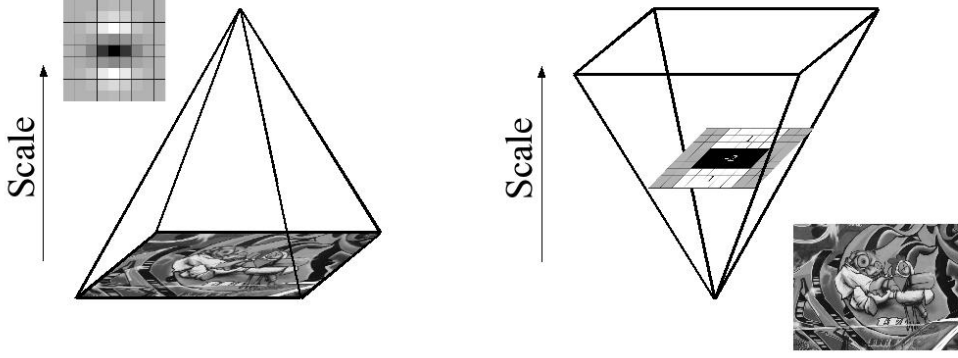
$$\text{Det}(H) \cong D_{xx}D_{yy} - (wD_{xy})^2 \quad (2.12)$$

2.12 formülünde w , Gauss filtreleri arasındaki enerji dönüşümü kullanılarak hesaplanır ve Bay ve arkadaşları (2008) tarafından 0.9 olarak önerilmiştir.



Şekil 2.7. Gauss türevlerini hesaplamak için kullanılan kutu filtreler (Bay, 2008)

SURF ölçek uzayını hesaplamak içinde sabit boyuttaki kutu filtrelerden faydalanır. SIFT metodunda kullanılan ardışık Gauss filtreleme ve alt örnekleme işlemine alternatif bir yol izler. Görüntüyü alt örnekleme yerine filtreleri üst ölçeğe taşıyarak işlem yükünü azaltır. Bu uygulama Şekil 2.8’de gösterilmiştir.



Şekil 2.8. SIFT ve SURF ölçek uzayları (Evans, 2009)

Şekil 2.7’de gösterilen filtreler, $\sigma = 1,2$ değerine sahip bir Gauss filtreye karşılık gelmektedirler. Filtrelerin boyutunda her oktavla birlikte yaklaşık 2 katına çıkar ve her oktavda sabit bir ölçek sayısına bölünerek değerlendirilir. Örneğin ilk oktavda 9x9 boyutunda olan kutu filtre üst oktaflara 15x15, 21x21 boyutlarında ilerleyerek devam eder.

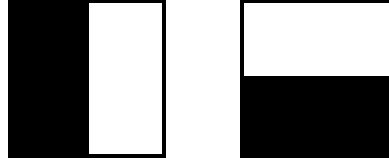
Öznelik noktalarının seçimi SIFT metodu ile benzerlik gösterir. Her nokta 3x3 pencere içerisindeki alt, üst ve mevcut oktaflardaki 26 pikselle karşılaştırılır. Extremum değere sahip olanlar ve eşik değerinden geçen noktalar elenerek seçilir. Noktaların yerlerinin bulunması için ölçek ve uzayda aradeğer bulma işlemi gerçekleştirilir. SURF’ te Brown ve Lowe’ nin (2002) uyguladığı metot standart olarak uygulanmıştır. P(x,y) noktasında ve σ ($p = (x, y, \sigma)$) ölçeğindeki Hessian matrisinin determinant Taylor serileri yardımı ile kuadrik biçimde 2.13 formülü ile ifade edilebilir.

$$H(p) = H + \frac{\partial H^T}{\partial p} + \frac{1}{2} p^T \frac{\partial^2 H}{\partial p^2} p \quad (2.13)$$

$H(p)$ nin türevini sıfıra eşitleyerek $\hat{p} = (x, y, \sigma)$ extrem noktanın konumu bulunabilir.

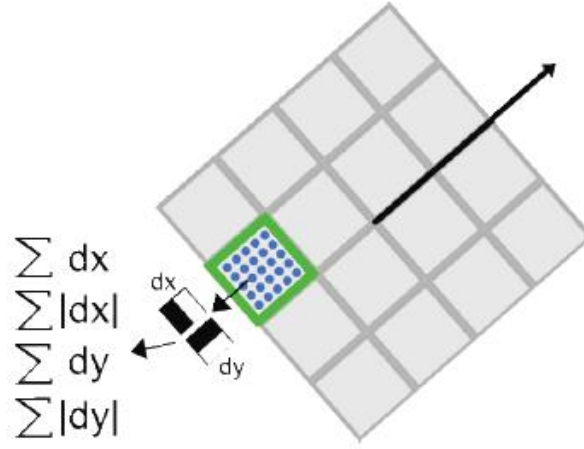
$$\hat{p} = \frac{\partial^2 H^{-1}}{\partial p^2} \frac{\partial H}{\partial p} \quad (2.14)$$

SURF metodu için tanımlayıcı vektör, öznitelik noktasının komşuları üzerindeki yoğunluk dağılımının incelenmesi ile hesaplanır ve x-y yönünde fark vektörlerinin hesaplanması için yine integral görüntü üzerindeki Haar dalgacıklarından faydalanılır. Şekil 2.9'da x ve y yönünde fark vektörlerinin hesaplanmasında kullanılan Haar dalgacıklarını göstermektedir. Siyah bölgeler +1, beyaz bölgeler -1 ile ağırlıklandırılmıştır.



Şekil 2.9. x ve y yönünde fark vektörlerini veren Haar dalgacıkları (Bay, 2008)

Öznitelik noktalarının yönelim değerleride Haar dalgacıkları sayesinde hesaplanır ve bu hesaplama sonrasında her öznitelik noktasına ait 64 boyutlu bir dizi oluşturulur. Öznitelik noktalarına rotasyondan bağımsız olması için, öznitelik dizilerinin her rotasyon değerine kendi vektörüne en yakın değerleri vermeleri gerekmektedir. $P(x,y)$ noktasında ve 's' ölçeğinde tespit edilen bir öznitelik noktasının tanımlayıcı dizisini hesaplamak için 20s boyutunda kare bölge oluşturulur ve bu kare 4x4 boyutunda alt bölgelere ayrıştırılır. Bu ayrıştırma Şekil 2.10'da gösterilmektedir.



Şekil 2.10. Tanımlayıcı vektör bileşenleri (Evans, 2009)

Her alt bölge 4 vektör bileşeninin eşlik ettiği bir alan olarak düşünülebilir. Bu alt bölgelerde 2×2 boyutundaki Haar dalgacıklarının cevapları 5×5 'lik bölgelere yerleştirilerek hesaplanır. x ve y bileşenlerinin dalgacık cevaplarına dx ve dy diyecek olursak 25 noktadan oluşan bölgenin cevabı 2.15 ile hesaplanabilir.

$$v_{\text{sub}} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy| \right] \quad (2.15)$$

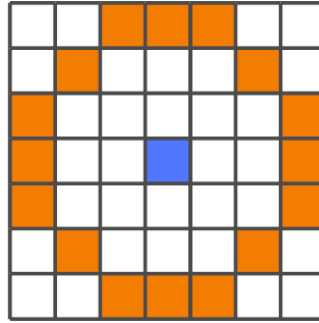
2.1.3. FAST Öznitelik Noktaları

Modern öznitelik tespit noktaları pek çok farklı uygulamada çok iyi sonuçlar verselerde, işlem maliyeti sorunları hep gündemde kalmıştır. Pek çok farklı zorlayıcı koşula karşı hesapsal olarak önlem almaları bir yandan ideal öznitelik tespitine bir adım yaklaşırsa da, pek çok uygulamada sadece öznitelik noktalarının hızlıca eşleşmeleri tatmin edici sonuçlar vermektedir. Rosten ve Drummond (2006) bu konuda yaptıkları çalışmada görüntüde oldukça hızlı öznitelik noktası tespit eden bir yöntem olarak FAST metodunu geliştirip önermişlerdir.

Piksel tabanlı bir test uygulamasına dayanan FAST yönteminde hedef pikselin bir köşeye ait olup olmadığı önemlidir. Uygulamada hedef pikselin etrafına (yarıçapı 3

piksel olan Bresenham çemberi şeklinde) 16 pikselden oluşan ark, Şekil 2.11’de görüldüğü gibi yerleştirilir. Daha sonra bu 16 pikselden oluşan arkta, sürekli 12 pikselin hedef noktanın bir eşik değeri ekleniş halinden daha büyük ya da aynı eşik değeri çıkartıldığında daha küçük olup olmadığına bakılır.

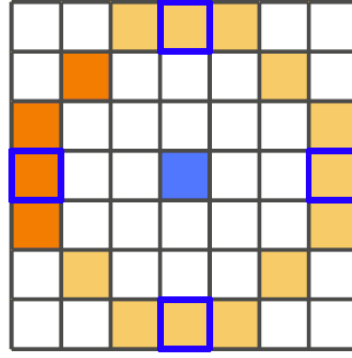
$$\forall k \in [1, 16]; \quad x_t = \begin{cases} x(i,j), & x(i,j) - t < arc(k) \\ x(i,j), & x(i,j) + t \geq arc(k) \end{cases} \quad (2.16)$$



Şekil 2.11. Hedef piksel etrafındaki 16 pikselden oluşan ark (Drummond, 2012)

2.16 formülünde de görüldüğü gibi arkın ortasında yer alacak olan öznitelik noktası belirlenen t eşik değerine bağlıdır ve t eşik değeri tespit edilecek öznitelik nokta sayısı ile ters orantılıdır. Bundan dolayı ihtiyaç duyulan hız ve nokta miktarına göre uygun bir değer belirlenmelidir.

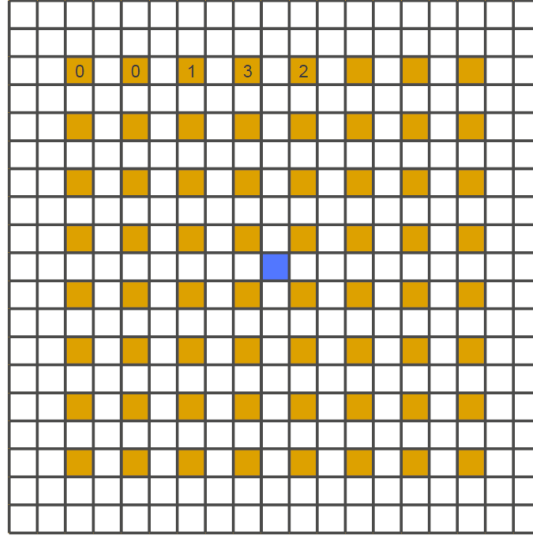
Görüntüde kenar nokta sayısı, test edilecek nokta sayısından daima az olacağından her hedef noktayı tam teste tabi tutmak işlem süresini arttıracaktır. Test adımını azaltmak için şöyle bir yaklaşım geliştirilmiştir; eğer hedef nokta bir öznitelik noktası ise 12 piksellik arkta 3 nokta test koşulunu sağlamalıdır. Şekil 2.12’de görüldüğü gibi arkta test edilen dört noktadan üçü şartı sağlarken, bir nokta kenar dışında (veya girişinde) kaldığından istenilen şartı sağlamamıştır.



Şekil 2.12. Hedef pikselin testinde istenilen şartı sağlayan ve sağlamayan noktalar (Drummond, 2012)

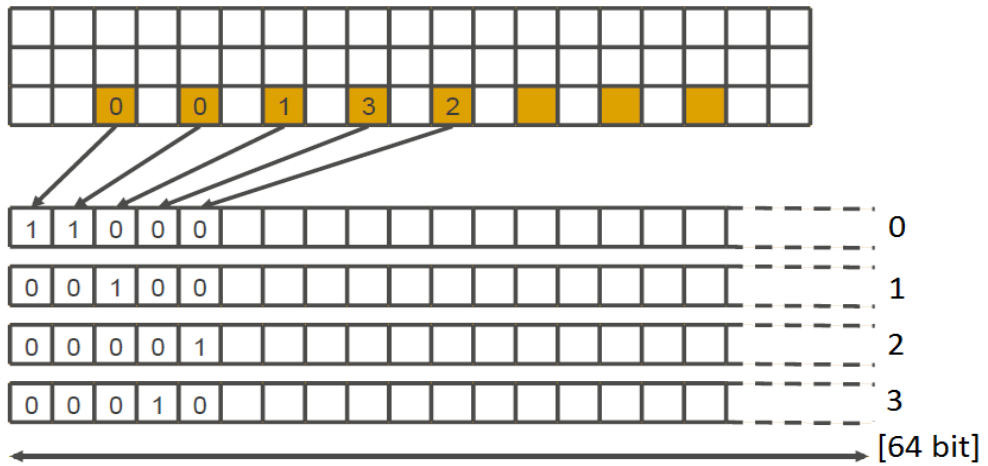
Eğer hedef nokta ilk testte başarılı olamazsa iterasyon ilerlemeden nokta elenir. İlk testin başarılı olması durumunda test, arktaki diğer noktalar içinde sürdürülür. Eğer noktalar simetrik çiftler olarak teste tabi tutulursa tam test için 8 iterasyon gerekecektir. Rosten'in önerdiği yöntemde karar ağacını oluştururken 12 yerine 9 pikselde olumlu test noktası beklemenin yeterli olduğunu gözlemlemiştir. FAST bu noktada olumlu test sonuç beklediği piksel dizisi açısından FAST – 9 ve FAST – 12 arasında versiyonlanır.

FAST yönteminde öznelik noktalarının tanımlayıcı vektörlerine ilişkin diğer pek çok yöntemin aksine standart bir uygulama yoktur. Bununla birlikte önerilen pek çok yöntem öznelik noktalarının kullanılacağı uygulamaya göre çeşitlenirler. Bu yöntemlerden bir tanesi Taylor ve Drummond'un (2009) önerdiği yöntem olan nicemlenmiş yoğunluk alanlarıdır. Bu yöntem temel olarak öznelik nokta etrafında oluşturulacak ızgara örüntü üzerinden ikili öznelik vektörü elde etme yöntemidir. Şekil 2.13'te görüldüğü gibi hedef nokta etrafından 8x8 boyutunda ızgara aralarında birer piksel boşluk olacak şekilde çıkartılır. Bu ızgaradaki değerleri direkt saklamak yerine önce değerlerin ortalama ve varyansı hesaplanarak tüm değerler dört adet değere nicemlenirler.



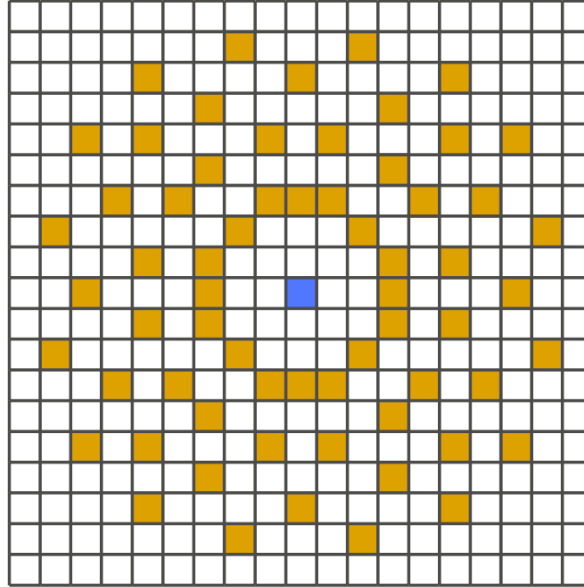
Şekil 2.13. Öznitelik noktasının etrafındaki 8x8 boyutunda ızgara ve ızgara üzerindeki nicemlenmiş değerler (Drummond, 2012)

Her dört değer için 8x8 değerden gelecek olan değerler, nicemeleme seviyesine göre ayrı dizilerde tutulurlar. Bu işlem sonucunda Şekil 2.14'te görüldüğü gibi her nicemeleme seviyesi için 64 bit ikili sayı dizisi elde edilecektir. İkili sayı dizilerini, diğer veri tiplerindeki dizilere göre kıyaslamak çok daha hızlı olacağından bu yöntem sıklıkla tercih edilebilmektedir. Diziler arası istatistiksel mesafe (Euclid ya da Mahalanobis) yerine diziler arası Hamming mesafesi yada daha basit bir biçimde XOR işlemi ile hesaplanabilir.



Şekil 2.14. Nicemlenmiş değerlerin ikili sayı dizilerine atanması (Drummond, 2012)

Bu yöntem son derece elverişli olsada rotasyonel deęişimlere karşı son derece dayanıksızdır. Rotasyonel deęişimde dağılım simetrisi bozulacağından, sayı dizilerinin yerleri de deęişecek, en iyi ihtimalle ötelenecektir. Drummond ve arkadaşları (2009) bu duruma çözüm getirmek için rotasyonel nicemlenmiş yoğunluk alanlarını kullanmıştır. Rotasyonel nicemlenmiş yoğunluk alanlarında, ızgara örüntüsü dairesel olarak uygulanmıştır ve hassasiyet için nicemeleme deęeri arttırılmıştır. Dairesel ızgara örüntüsü Şekil 2.15'te görülmektedir. Sahneler arasında rotasyonel deęişim söz konusu olduğunda tanımlayıcı vektör dizisindeki deęerler ötelenmektedir. 32 bit ötelenme, açısız olarak π kadar deęişime karşılık gelmektedir.



Şekil 2.15. Rotasyonel nicemlenmiş yoğunluk alanlarında kullanılan ızgara örüntüsü (Drummond, 2012)

2.1.4. ORB Öznitelik Noktaları

Önceki yöntemlerde de görüldüğü gibi öznitelik tespit ve tanımlama metodları arasında her zaman sağlamlık ve hız kriterleri arasında takas vardır. SIFT ve SURF gibi metodlar çok parametrik bir biçimde çok fazla sağlam nokta tespit etmelerine karşın işlem yükü çok fazladır ve bu sebepten dolayı işlemlerin gerçek zamanlı uygulanabilmesi için daha

güçlü donanıma ihtiyaç duyarlar. FAST ise aksine çok hızlı öznitelik noktası tespit ve tanımlama yapabilmesine rağmen öznitelik noktalarının kalitesi daha düşüktür; dolayısıyla ardışıl hedeflerde tespit edilebilen nokta sayısı çok daha azdır. Bu noktada araştırmalar daha optimum sonuç verebilecek tespit yöntemlerine yönelmiştir. ORB yöntemi Rublee ve arkadaşları (2011) tarafından FAST ve BRIEF yöntemlerinin hız ve sağlamlıklarının avantajını kullanmak için tasarlanmış bir metottur.

ORB öznitelik noktası tespitini FAST yöntemi (tercihen FAST-9) üzerinden yapmaktadır. SIFT ve SURF'ye benzer biçimde (ancak Hessian matrisini kullanmayarak) kenar noktalar üzerinde yoğunlaşan öznitelik noktalarını elemek için Harris kenar tespiti yöntemindeki kenar kriteri (2.18) formülünü kullanır.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial y \partial x} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (2.17)$$

$$M_c = \det(M) - k(\text{trace}(M))^2 \quad (2.18)$$

2.18 sayesinde tespit edilen noktalar üzerinde ilk önce kenar FAST eşik değeri yeterince düşürülerek istenilen sayının üstünde nokta elde edilir. Daha sonra Harris kenar kriterine göre sıralanan noktalardan en iyi sonuç verenlerden ihtiyaç duyulan kadar nokta bu küme içerisinde seçilir.

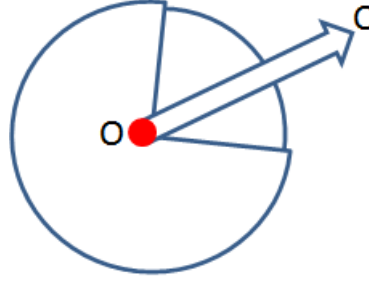
FAST noktaları, tespit edildikleri alanlar içerisinde ekstremum noktalar olduklarından yönlenme atamak için uygun noktalardır. Rosin(1999) bu noktaları yoğunluk merkezleri olarak tanımlamıştır ve bu alanların momentlerini hesaplamak için 2.19 formülünü kullanmıştır.

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (2.19)$$

Momentler yardımı ile merkez 2.20 ile hesaplanabilir.

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2.20)$$

Şekil 2.16’da görüldüğü gibi O merkezinden C koordinatına yönelen \overrightarrow{OC} vektörünün açısal değeri 2.21 yardımı ile hesaplanabilir.



Şekil 2.16. Öznitelik noktasının yönelme vektörü

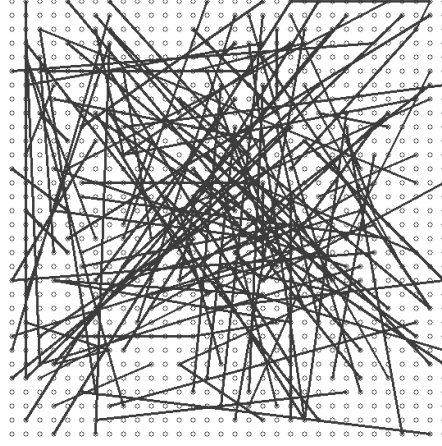
$$Ang(\overrightarrow{OC}) = \theta = atan2(m_{01}, m_{10}) \quad (2.21)$$

ORB metodu daha öncede bahsettiğimiz gibi tanımlayıcı için BRIEF’i temel almaktadır. BRIEF yönteminde de FAST ta önerilen yöntemlere benzer şekilde ikili tanımlayıcı dizisi kullanılmaktadır. Tercihen ortalaması $\mu = 0$ ve standart sapması $\sigma = \frac{1}{25} S^2$ (S, test bölgesi boyutu) olan bir Gauss filtreden geçirilmiş görüntü üzerinde ikili tanımlayıcı testi şu şekilde tanımlanmıştır:

$$\tau(I(x, y), I(x', y')) = \begin{cases} 1, & I(x, y) < I(x', y') \\ 0, & \text{diğer durumda} \end{cases} \quad (2.22)$$

Burada $I(x, y)$ ve $I(x', y')$ noktaları – mümkün oldukça kenarlara yakın şekilde rasgele seçilmiş noktalardır. Calonder ve arkadaşları (2010) rasgele seçimin, belirli bir örüntüye göre seçimden daha iyi sonuç verdiğini çalışmalarında paylaşmışlardır. Rasgele bir seçim örüntüsü Şekil 2.17’de görülmektedir. “n” dizi boyutu (128, 256, 512 ... vb.) olmak üzere v_{brief} tanımlayıcı vektörü 2.23 denklemi ile şu şekilde ifade edilir:

$$v_{brief} = \sum_{1 \leq i \leq n} 2^{i-1} \tau(I(x, y), I(x', y')) \quad (2.23)$$



Şekil 2.17. BRIEF tanımlayıcı vektörü için gereken rasgele eşleşme örüntüsü (Ruble, 2011)

BRIEF tanımlayıcı vektörü ayırt edici olsada, rotasyonel değişimler için çokta doğru sonuç vermemektedir. Rotasyonel değişimlere dayanıklılık sağlamak için v_{brief} tanımlayıcı vektörlerinin yönlendirilmesi gerekmektedir. Yönlendirilmiş vektör hesaplamak için ilk olarak kendimize “n” boyutlu bir test noktaları kümesi oluşturmalıyız.

$$S = \begin{pmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \end{pmatrix} \quad (2.24)$$

Daha önce hesaplamış olduğumuz bölgesel yönlenme açısı θ yı da kullanarak, uygun bir R_θ dönüşüm matrisi ile döndürülmüş bölge şu şekilde olacaktır:

$$S_\theta = R_\theta S \quad (2.25)$$

Yönlendirilmiş BRIEF operatörü artık bizim için ORB tanımlayıcı vektörünü bulacaktır.

$$v_{orb} := v_{brief} | (I(x_i, y_i), I(x_i', y_i')) \in S_\theta \quad (2.26)$$

Orjinalinde tavsiye edilen BRIEF tanımlayıcılarının 12 derecelik aralıklarla değerleri hesaplanır, θ yönlenmesine görede doğru S_θ bölgesi oluşturularak ORB tanımlayıcı vektörü hesaplanabilir.

2.2. İki Boyutlu Hareket Modelleri

Düzlemsel koordinat sistemlerinde, bir görüntüye ait pikseller lineer bir ilişki içerisinde aynı koordinat sistemi içerisinde farklı konumlara hareket edebilirler. İki Görüntü pikselleri arasındaki bu hareketler çeşitli dönüşümler ile tanımlanmıştır.

İki boyutta ötelenme hareketi basit bir biçimde denklem 2.16 ile ifade edilebilir. x' pikselin yeni koordinatı olacak şekilde:

$$x' = [I \ t] \dot{x} \quad (2.16)$$

Burada I (2x2) boyutunda birim matris, $\dot{x} = (x, y, 1)$ pikselin homojen koordinatlarıdır. Bu ifade basit bir öteleme işlemidir ve taşınan görüntüde geometrik özellikler korunur, sadece konum olarak belirli bir yönde ötelenmiş olurlar.

Rotasyonel dönüşüm iki boyutta Euclid dönüşümü olarakta bilinmektedir. Denklem 2.17 ve 2.18 ile ifade edilebilir.

$$x' = [R \ t] \dot{x} \quad (2.17)$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (2.18)$$

2.18 denkleminde R, ortonormal rotasyon matrisidir. Bu dönüşümde görüntünün merkez noktası t kadar ötelenmekle birlikte, merkezi etrafında da θ açısı kadar dönme hareketi yapar.

Görüntüde paralel kenarların korunduğu affine dönüşümde, doğrusal olarak 2.17 denklemi ile aynı işlem yapılır. Ancak R dönüşüm matrisi 2x3 elemanlıdır. Yeni piksel koordinatları 2.19 denklemi ile hesaplanabilir.

$$x' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \dot{x} \quad (2.19)$$

İzdüşümsel dönüşümde ise görüntüde sadece doğrusal çizgiler korunur. Dönüşüm matrisi 9x9 boyutunda ve homojendir. İzdüşümsel dönüşüm pek çok kamera hareketini ve perspektif değişimlerine uyarlanabildiğinden dolayı en sık kullanılan dönüşüm tekniğidir. Piksellerin yeni koordinatları şu şekilde ifade edilebilir.

$$x' \sim H\hat{x} \quad (2.20)$$

Sonuç koordinatlarını normalize edecek olursak:

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \quad (2.21)$$

$$y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}} \quad (2.22)$$

Dönüşüm matrisinin normalize olması açısından h_{22} parametresi genellikle 1 değerini alır. Dolayısıyla dönüşüm matrisi hesaplanırken, elde edilmesi gereken sekiz adet değişken vardır.

2.3 Dönüşüm Matrisi

Önceki bölümlerde bahsedilen öznitelik fonksiyonlarının tespiti ve eşleştirilmeleri işlemleri tamamen iki görüntü arasındaki koordinat değişimi ve geometri ilişkisini veren dönüşüm matrisinin hesaplanması içindir. Eşleşen noktaların koordinat değişimine göre hesaplanan dönüşüm matrisi yardımı ile eklenecek görüntü, ana görüntü üzerinde uygun bir biçimde yerleştirilebilir. Bu çalışmalarda hesapladığımız dönüşüm matrisi genellikle izdüşümseldir. İzdüşümsel dönüşüm matrisini hesaplamak için doğrusal dönüşüm en temel hesaplama yöntemidir. Doğrusal dönüşüm kullanılarak 8 serbestlik derecesinde olan dönüşüm matrisini hesaplamak için tekrar izdüşümsel dönüşüm modeli üzerinden gidilecek olursak 2.20 denklemini ve dönüşüm matrisinin parametrelerini şu şekilde tanımlayabiliriz.

$$w \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.21)$$

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \quad (2.22)$$

Eşitlikleri denklem formunda yazacak olursak:

$$wx' = h_1x + h_2y + h_3 \quad (2.23)$$

$$wy' = h_4x + h_5y + h_6 \quad (2.24)$$

$$w = h_7x + h_8y + h_9 \quad (2.25)$$

Denklemleri w' den bağımsız hale getirirsek:

$$(h_7x + h_8y + h_9)x' - (h_1x + h_2y + h_3) = 0 \quad (2.26)$$

$$(h_7x + h_8y + h_9)y' - (h_4x + h_5y + h_6) = 0 \quad (2.27)$$

2.26 ve 2.27 denklemlerini $A_i H = 0$ şartını sağlayacak şekilde yeniden biçimlendirilecek olursak sırasıyla A_i ve H parametreleri şu şekilde olacaktır.

$$A_i = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{pmatrix} \quad (2.28)$$

$$H = (h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9)^T \quad (2.29)$$

2.28 de A_i matrisinde de görüleceği gibi, her eşleşen nokta çifti bize iki eşitlik vermektedir. H dönüşüm matrisinde h_9 değerini 1 alacağımız göz önüne alınırsa,

hesaplanması gereken değerler için 8 adet eşitlik gerekmektedir. Dolayısıyla doğrusal dönüşüm yönteminin sonuç verebilmesi için en az 4 adet eşleşen nokta çiftine ihtiyacımız vardır. Bu 4 nokta çiftinde aranılan tek özellik 3 çiftin doğrudan olmaması gerektiğidir.

Dönüşüm matrisinin doğru hesaplanması için eşleşen nokta çiftlerinin hem görüntü uzayında hemde gerçek dünya koordinatlarında doğru eşleştikleri kabul edilir. Ancak bu durum pek çok durumda sağlanamamaktadır. Bu durumların başında görüntüler arasında oluşacak gürültüden ve diğer sebeplerden kaynaklanan eşleşme hataları gelmektedir. Dönüşüm matrisinin hesaplanması için yanlış eşleşen nokta çiftlerinden gelecek değerlerin elenmesi gerekmektedir. RANSAC yanlış eşleşme durumlarında en iyi dönüşüm matrisi kestirimin yapan en iyi algoritmalarından biridir.

RANSAC algoritması temel olarak eşleşen nokta çiftlerinin oluşturduğu dönüşüm matrisi modelini, nokta çiftlerinden rastgele alınan örneklerin oluşturduğu modeller arasından en iyisini seçmek üzerine kurmuştur. Algoritmanın işleyişini şu şekilde adımlandırabiliriz.

- 1- İki görüntü arasında eşleşen nokta çiftleri arasından rastgele 4 tanesi seçilir.
- 2- Bu 4 nokta yardımı ile dönüşüm matrisi hesaplanır.
- 3- Dönüşüm matrisi diğer öznitelik noktalarına da uygulanır, dönüşüm sonucu eşleşmeler ve öznitelik değerleri ile eşleştirmesi arasındaki koordinat farkı ile SSD hesaplanır. SSD yardımı ile modele uyan ve uymayan noktalar tespit edilir. Bu adımda SSD hatasına bir eşik değeri verilmesi gerekmektedir. Bu eşik değeri çoğu zaman mevcut ve bir önceki iterasyonda hesaplanan doğru eşleşmeler arasındaki oran olabilir.
- 4- Her eşleşmedeki modele uyanlar ve uymayanlar tespit edilir.
- 5- Belirlenen iterasyon değeri sonlanana kadar 1. adımdan işlemler aynı şekilde tekrar eder. İterasyon sonucunda en fazla doğru eşleşmenin olduğu dönüşüm, ana dönüşüm matrisi olarak atanır.

Şekil 2.18’de SURF algoritması ile elde edilmiş olan öznelik noktalarının eşleşmeleri gösterilmiştir. Üstte yer alan görüntüde 208 adet öznelik noktası, alttakinde ise 224 adet öznelik noktası tespit edilmiş, en az öznelik noktasına sahip olan görüntü baz alınarak, 208 adet eşleşme yapılmıştır. Görüntüde yanlış eşleşmeler çok kolaylıkla gözlenebilmektedir. Yanlış eşleştirmelerin eleminasyonu için RANSAC algoritmasıyla tespit edilmiş 191 adet model oluşturacak doğru nokta Şekil 2.19’da yer almaktadır.

RANSAC algoritmasında en önemli parametre, algoritmanın koşturulması sırasındaki iterasyon sayısıdır. Teorik olarak eşleşen noktalar arasındaki tüm dörtlü kombinasyonları işleme tabi tutmak mümkündür. Ancak iterasyon sayısı, sistemin cevap hızı ile ters orantılı olacağından daha uygun bir değer seçilmesi daima istenmektedir. Bunun için Hartley ve Zisserman (2003) uygun bir iterasyon seçimi için bir formül oluşturmuşlardır.

$$N_{iter} = \frac{\log(1-p)}{\log(1-(1-\epsilon)^S)} \quad (2.30)$$

Formül 2.30 da görüldüğü üzere N iterasyon için istenilen parametreler; p, bir örnek seçiminde modele uygun olmayan hiçbir eşleşme olmaması olasılığı; ϵ , eşleşmelerin modele uymama olasılığı ve S ise görüntü dönüşümü için seçilen örnek sayısıdır.



Şekil 2.18. Tüm eşleşen öznitelik noktaları



Şekil 2.19. Doğru eşleşmeye sahip öznitelik noktaları

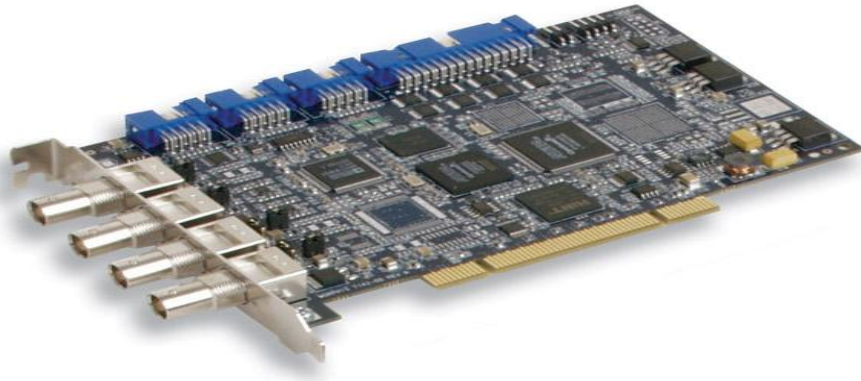
3. MATERYAL VE YÖNTEM

3.1 Uygulamalarda Kullanılan Veri Setleri ve Uygulama Ortamı

Otomatik mozaikleme uygulamalarında en sık kullanılan veri kümesi sabit görüntü dosyalarından oluşan veri setleridir. Bir görüntüleme cihazı ile alınabilecek her türlü görüntü veri setinin elemanı olabilir. Çalışmalarımızda Ikegami elektro-optik kamera üzerinden aldığımız görüntüleri kullanmaktayız. Temel olarak seçtiğimiz çözünürlükler her uygulamada değiştiğinden ayrıca belirtilmiştir. Aynı kamera gerçek zamanlı görüntü işlediğimiz uygulamalarda da kullanılmıştır.

Bir diğer veri seti sabit video üzerinden alınan görüntülerdir. Görüntülenecek alana ait video türleri arasından uygulamalara en elverişli olanlar insansız hava araçları UAV'ler üzerinden alınan görüntülerdir. Kullandığımız görüntüler "HighVizPhotography" grubundan alınan düşük yükseklikte uçan UAV'lerden alınan videolardır. Yüksek çözünürlüklü (1280x720), 30 FPS ve 16 bit görüntü derinliğinde olan görüntüler ayrıca görüntü sabitleme işleminden geçtiğinden ayrıca sabitleme yapılmamıştır.

Gerçek zamanlı uygulamalarda ise sahne yakalama donanımı kullanılmıştır. Özel bir DSP kullanan sahne yakalama donanımı temel olarak aldığı gerçek zamanlı analog video görüntüsünü sayısallaştırarak yazılım ortamına aktarmaktadır. Çalışmalarımızda kullandığımız sahne yakalama donanımı Matrox Morphis, Şekil 3.1 de görülmektedir.



Şekil 3.1. Matrox Morphis sahne yakalama kartı

Görev bilgisayarına PCI girişi üzerinden bağlantı sağlayan sahne yakalama kartı, üzerindeki yazılım arayüzü sayesinde ana proje ile haberleşebilmektedir. 8 bit derinlikte 768x576 piksel görüntü çözünürlüğünde ve PAL formatında gerçek zamanlı görüntü aldığımız sahne yakalama donanımının uygulamalar için tek görüntü kanalını, 25-30 FPS olacak şekilde kullanmaktayız.

Uygulamaların tamamı C/C++ ortamında, Eclipse arayüzü üzerinde gerçekleştirilmiştir. Windows 7 işletim sistemi, Intel i7 CPU üzerinde çalıştırılan yazılımlar birkaç durum dışında tek iş parçacıklı çalışmaktadır ve GPU kullanılmamıştır. Veri görselleştirme ve grafik ekranlar için OpenCV ve MATLAB, temel görüntü işleme algoritmalarının uygulamaları için Intel IPP, diğer matematik ve yazılım hafıza işlemleri için Boost ve Eigen kütüphaneleri kullanılmıştır. Tüm bunlarla birlikte sahne yakalama donanımı ile haberleşmek için MIL kütüphanesinin fonksiyonları kullanılmıştır.

3.2. Mozaik Görüntülerin Birleştirilmesi

Mozaik görüntülerin birleştirilmesi işleminde ilk adım görüntü üzerindeki öznitelik noktalarının çıkartılması ve eşleştirilmeleridir. Bu çalışmada özellikle incelediğimiz 4 öznitelik tespit algoritmasını (SIFT, SURF, FAST, ORB) ardışık iki sahne üzerindeki tespit noktaları Şekil 3.2, Şekil 3.3, Şekil 3.4 ve Şekil 3.5' te gösterilmiştir.



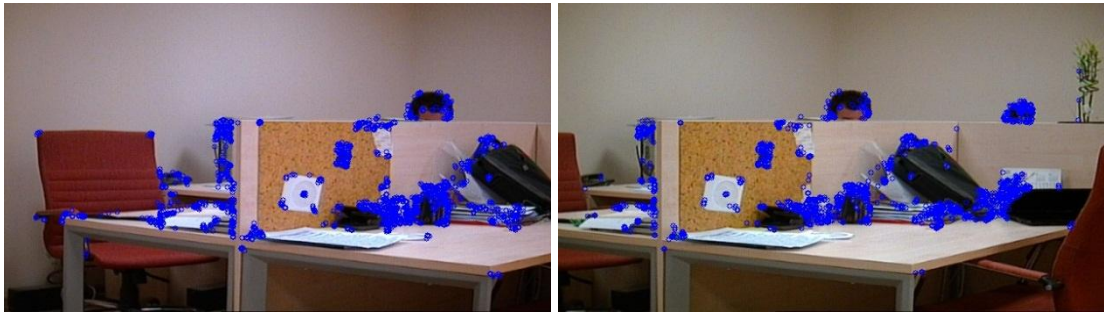
Şekil 3.2. Ardışık iki sahnedeki SIFT noktaları



Şekil 3.3. Ardışık iki sahnedeki SURF noktaları



Şekil 3.4. Ardışık iki sahnedeki FAST noktaları



Şekil 3.5. Ardışık iki sahnedeki ORB noktaları

Eşleşen noktalar bulunduktan sonar dönüşüm matrisi hesaplanmıştır. RANSAC algoritması üzerinden hesaplanan dönüşüm matrisi, 3 farklı yöntemde şu sonuçlarla bulunmuştur.

$$H_{SIFT} = \begin{bmatrix} 0,9398 & -0,0207 & -0,0271 \\ 0,9589 & -130,4590 & 13,2649 \\ 0 & -0,01 & 1 \end{bmatrix} \quad (3.1)$$

$$H_{SURF} = \begin{bmatrix} 0,95 & -0,0282 & -0,0271 \\ 0,81 & -125,56 & 13,2601 \\ 0 & -0,04 & 1 \end{bmatrix} \quad (3.2)$$

$$H_{FAST} = \begin{bmatrix} 0,901 & -0,023 & -0,03 \\ 0,84 & -125,56 & 14,004 \\ 0,003 & -0,04 & 1 \end{bmatrix} \quad (3.3)$$

Sonuçlardan da görüleceği üzere dönüşüm matrisleri (normalize edilmiş değerleri) birbirlerine oldukça yakın bir şekilde hesaplanmıştır. Bu dönüşüm matrisi kullanılarak elde edilen birleştirilmiş görüntü Şekil 3.6'da gösterilmiştir. Şekil 3.6'da görüntü SIFT kullanılarak uygulanan yöntemin sonucudur ve diğer yöntemler de bu sonuca yakın dönüşüm vermektedir. Görüldüğü üzere iki görüntü geometrik olarak üst üste çakıştırılarak geçiş sağlanmıştır ancak geçiş noktasında kontrast farkından dolayı bir uyumsuzluk meydana gelmektedir. Bu dikme bölgesindeki izi gidermek için dikışsiz bütünleştirme yöntemlerinden ayrıca bahsedilmiştir.

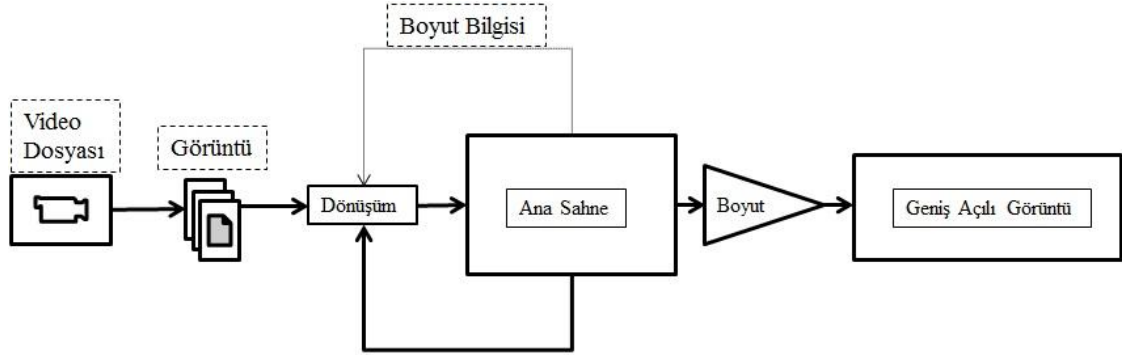


Şekil 3.6. Dönüşüm matrisi yardımı ile birleştirilmiş görüntü

3.3 Video Dosyası Üzerinde Otomatik Mozaikleme

Video dosyası üzerinden görüntü birleştirme uygulamaları, diğer sabit görüntü dosyaları ile yapılan uygulamalar kadar yaygındır. Yöntem olarak aynı metotlar kullanılsada, uygulamada bir takım farklılıklara dikkat etmek gerekmektedir.

Örnek vermek gerekirse video dosyalarından görüntü birleştirme uygulamalarında, görüntüler arası akışın sürekli olması ve akış yönü önemlidir. İstenilen bölgeye ait görüntü; açısız görüş alanı, çözünürlük, saniyede gelen görüntü sayısı ve kameranın hareket yönü gibi pek çok farklı etkene bağlıdır.



Şekil 3.7. Video üzerinden geniş açılı görüntü oluşturma akış şeması

Şekil 3.7’de video üzerinden geniş açılı görüntü oluşturma uygulamasının genel akışı verilmiştir. Beklendiği gibi video dosyası önce video çözümleyici yardımı ile videoyu oluşturan sahneler teker teker akış üzerinden alınır. İlk sahne temel oluşturacak şekilde ana sahnenin – tercihen – ortasına yerleştirilir. Burada ana sahnenin çözünürlüğü görüntüleneceği ekranla sınırlı tutulabileceği gibi, bir arayüz yardımı ile de dinamik olarak boyutu güncellenebilir. Bu durumda ana sahne boyutuna ilişkin olarak eklenecek yeni görüntünün de boyutunun güncellenmesi gerekmektedir.

3.3. Dikişsiz Bütünleştirme İşlemi

Görüntülerin üst üste çakıştırılması çoğu otomatik mozaikleme uygulamasında yeterlidir. Ancak amaç sadece genişaçılı görüntüyü oluşturmak değil, görüntüde

bozulma olmaksızın birleştirme yapılması istendiğinde dikişsiz birleştirme işleminin uygulanması gerekmektedir.

Birleştirme bölgesindeki bu bozulmaya, lensteki açıklık farkı, kontrast değişimi veya geometrik olarak görüntülerin tam oturmaması sebep olabilir. Basit bir şekilde görüntüler arasında yumuşak ve keskin geçiş bölgeleri parametrik olarak hesaplanamamaktadır. Sadece görüntüdeki alfa kanalının ağırlıklandırılması ile geçici olarak bu problem giderilebilir ancak uzun serilerden oluşan birleştirme işlemlerinde bu çözüm toplamsal hatalara sebep olmaktadır. Bu problemin daha genel bir çözümü ile ilgili Laplaciye piramitleri ve Poisson tabanlı iki yöntem geliştirilmiş ve sıklıkla kullanılmaktadır.

3.3.1. Laplaciye Piramitleri ile Görüntü Bütünleştirme

Burt ve Adelson'un (1983) bu konuda yaptığı çalışmalarda, sabit boyutlu genlik yerine frekans uzayına adaptif bant geçiren filtrelerle (Laplaciye piramitleri ile) geçişlerin daha yumuşak olduğu gözlenmiştir.

Bu yöntemin uygulanmasında alışlagelmiş piramitleme yöntemi kullanılmaktadır. Her görüntü öncelikle Laplace filtreden geçirilerek yumuşatılmakta ve piramidin bir üst ölçeği için alt örneklenmektedir. Laplace filtrenin kullanılması sayesinde görüntü geri boyutta elde edilmek istediğinde çok az bir bozulmayla geri elde edilebilmektedir. Bozuk yada kenar bölgelere ait pikseller komşu pikseller üzerinde dağılacaktır.

Daha sonra her ölçek ve filtrelenmiş görüntüler piramidin altına doğru tekrar geri oluşturulmaya başlanır. Her piramid seviyesi toplanır ve aradeğer bulma işlemi ile eksik piksel değerleri hesaplanır.

3.3.2. Poisson Tabanlı Görüntü Bütünleştirme

Perez ve arkadaşlarının (2003) önerdiği yöntemde, fark vektörü uzaylarından faydalanılarak dikişsiz görüntü bütünleştirme yöntemini sunmuştur. Görüntüde

pikselleri kopyalamak yerine eklenecek bölgeye ait fark uzayı vektörleri kopyalanır ve son görüntü pikselleri bu fark vektörleri kullanılarak hesaplanır. Bu hesaplama temel olarak dikiş bölgelerindeki eşleşen ve Dirichlet koşullarını sağlayan fark vektörlerinin Poisson denklemi yardımı ile çözülmesi ile sağlanır.

Şekil 3.8’de poisson tabanlı görüntü bütünleştirilmiş hali görülebilir ve Şekil 3.6’daki birleştirilmiş görüntü üzerinden iyileştirme oranı gözlemlenebilir.



Şekil 3.8. Poisson tabanlı bütünleştirme yöntemi uygulanmış görüntü

4. BULGULAR VE TARTIŞMA

Bu bölümde yer alan çalışmaların değişik ortamlarda yapılan benzetimler ve sonuçları yer almakta; bu çalışmalar sırasında yapılan gözlem ve deney sonuçları üzerindeki bulgulardan da bahsedilecektir.

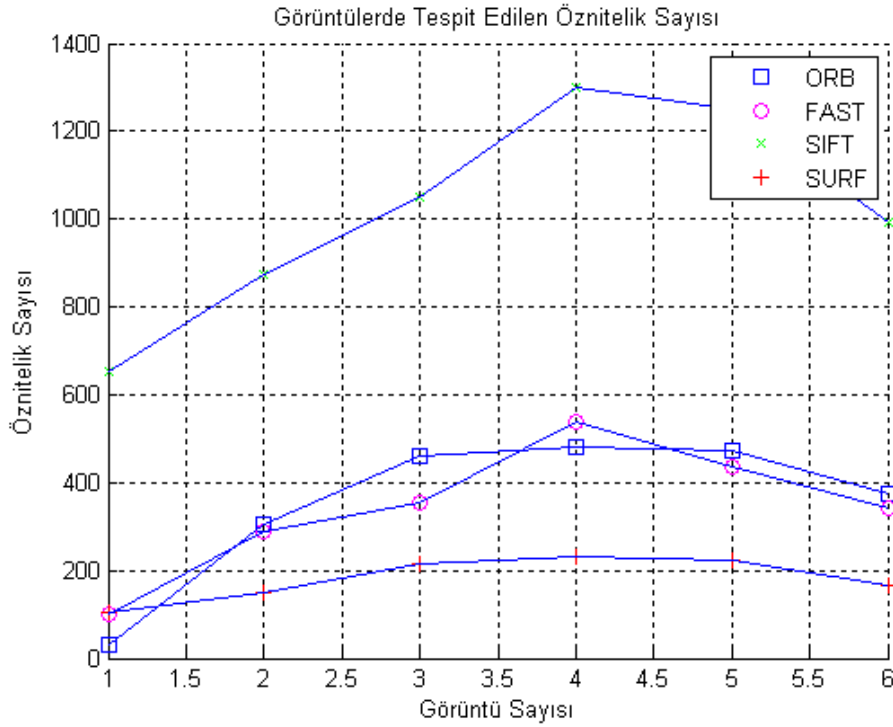
İlk olarak yaptığımız benzetimde kapalı ortamdan alınan altı adet görüntünün birleştirilmesi işlemi yer almaktadır. Bu işlemde kamera hareketi silindirik olarak soldan sağa doğru tripod üzerinde sağlanmıştır. Dolayısıyla referans görüntü ana sahnenin solundan başlayacaktır. Gelen her yeni frame, ana sahneye eklenmeden önce öznitelik noktaları çıkartılarak dönüşüm matrisi hesaplanır ve uygun dönüşümle eklenme işlemi yapılır. SURF noktalarının kullanıldığı bu benzetim sonucu Şekil 4.1 de gözlemlenebilir.



Şekil 4.1. SURF kullanılarak oluşturulmuş geniş açılı görüntü

Altı resmin birleştirilmesi işlemi toplam olarak 6,11s sürmüştür. SIFT ve FAST ile de uygulandığında sonuç olarak benzer görüntü çıktısı alınmıştır. Ancak uygulama hızı, tespit edilen öznitelik noktası sayısı ve doğru eşleşme değerleri arasında ciddi farklılıklar vardır. Şekil 4.2'deki grafikte incelenen dört yönteme ilişkin tespit edilen öznitelik nokta sayısı görülmektedir. Bu grafikten de görüleceği gibi SIFT, diğer iki yönteme göre daha fazla öznitelik noktası tespit etmiştir. Öznitelik noktalarının çokluğu doğru eşleşme çiftlerinin sayısını arttıracak gibi işlem yükünü de arttıracaktır. Görüntü birleştirme uygulamalarında gerçek zamanlı çalışma kriteri aranmadığı durumlarda tespit edilen nokta sayısının çokluğu, gürültüye karşı direnç sağlayacağından aslında

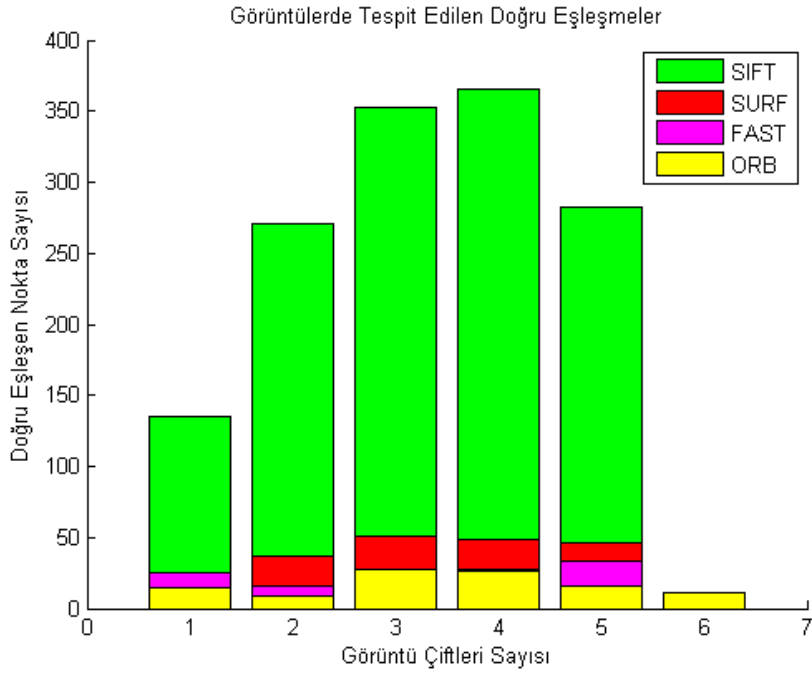
istenilen bir durumdur. Ancak performans gerektiren durumlarda daha az ve hızlı tespit edilmiş noktalara ihtiyaç vardır.



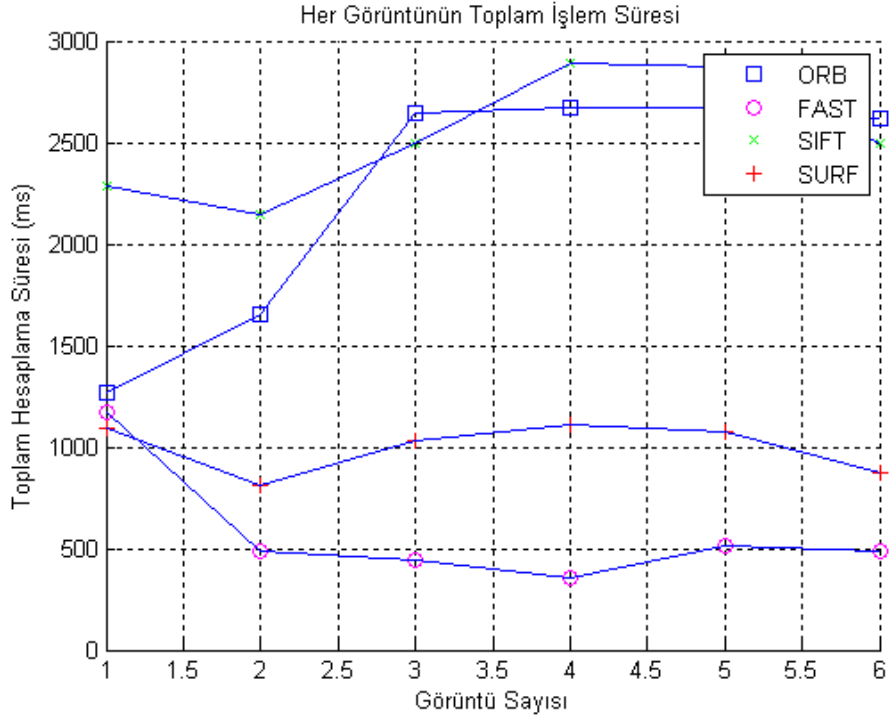
Şekil 4.2. Çalışılan dört farklı yöntem için uygulamada tespit edilen toplam öznitelik nokta sayısı

Dört yöntemde de öznitelik noktalarının doğru eşleşme değerleri Şekil 4.3'te yer alan grafikte gözlemlenebilmektedir. Nokta tespiti fazla olan yöntemlerin, eşleşme değerleride beklendiği gibi diğerlerinden fazladır.

Şekil 4.4'te yer alan grafikte dört yöntemin uygulamadaki performans süreleri yer almaktadır. dört yöntemde öznitelik çıkartma, bir önceki özniteliklerle eşleştirme performansı her işlem adımında ölçülmüş ve milisaniye olarak kaydedilmiştir. Grafikten de görüleceği gibi FAST, diğer yöntemlere göre daha çok daha hızlı sonuç vermektedir. Bu durum FAST yöntemini gerçek zamanlı uygulamalarda bir adım öne geçirmektedir. SURF yönteminin de ortalama 1 saniye gibi bir sürede sonuç vermesi oldukça iyi performans gösterdiğinin göstergesidir.



Şekil 4.3. Çalışılan dört farklı yöntem için uygulamadaki doğru eşleşme sayısı



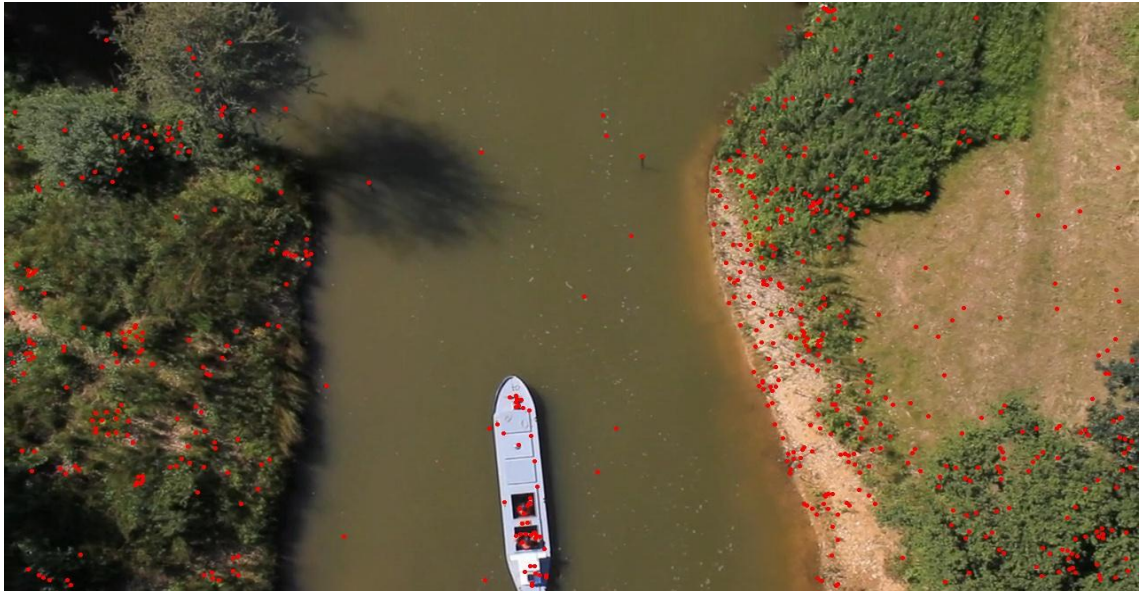
Şekil 4.4. Çalışılan dört farklı yöntem için her adımdaki toplam işlem süresi

Oluşturulan görüntünün Poisson tabanlı bütünleştirme ile işlendikten sonraki sonucu Şekil 4.5’ de görülmektedir.

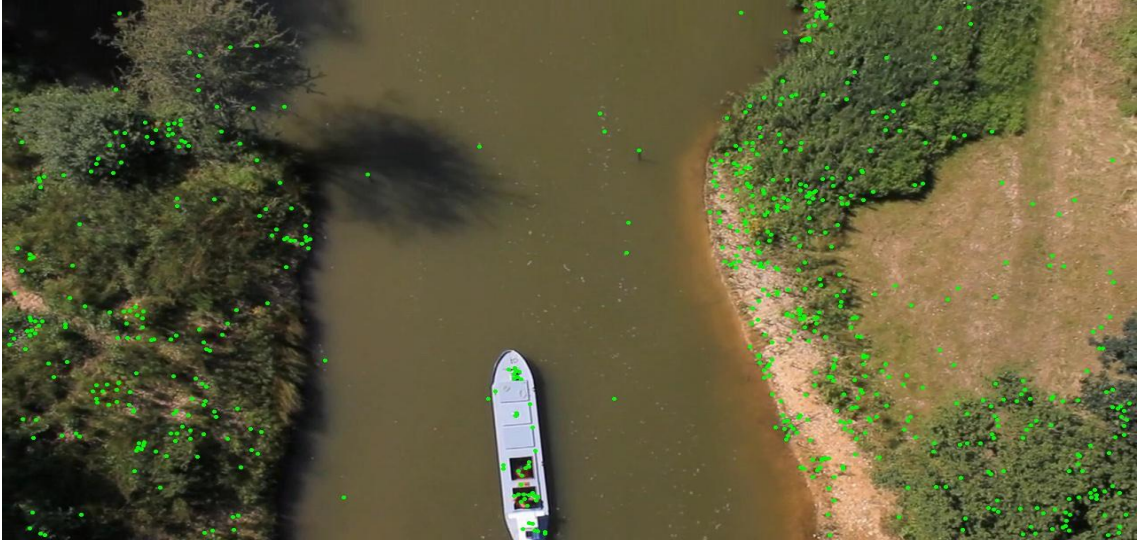


Şekil 4.5. Poisson tabanlı bütünleştirilmiş görüntü

Yöntemlerin bir diğer karşılaştırılmasında, eşleşme sonucu hareket vektörlerinin tespiti üzerindeki performansları ile değerlendirilebilir. İlerleyen bir UAV üzerinden alınmış değişim göstermeyen bir görüntünün ilerleme yönünü gözlemlemek için yine öznitelik noktalarından faydalanılabilir. Eşleşen noktaların oluşturduğu vektörler, optik akış hesaplanması gibi hareket vektörlerini verebilirler. Şekil 4.6’ da yer alan bir insansız hava aracı üzerindeki kameradan alınan 1280x720 piksel çözünürlükteki görüntü üzerinde SURF noktaları bulunmuş ve bir sonraki sahne olan Şekil 4.7’ de yer alan görüntü üzerinde de aynı işlem yapılmıştır.

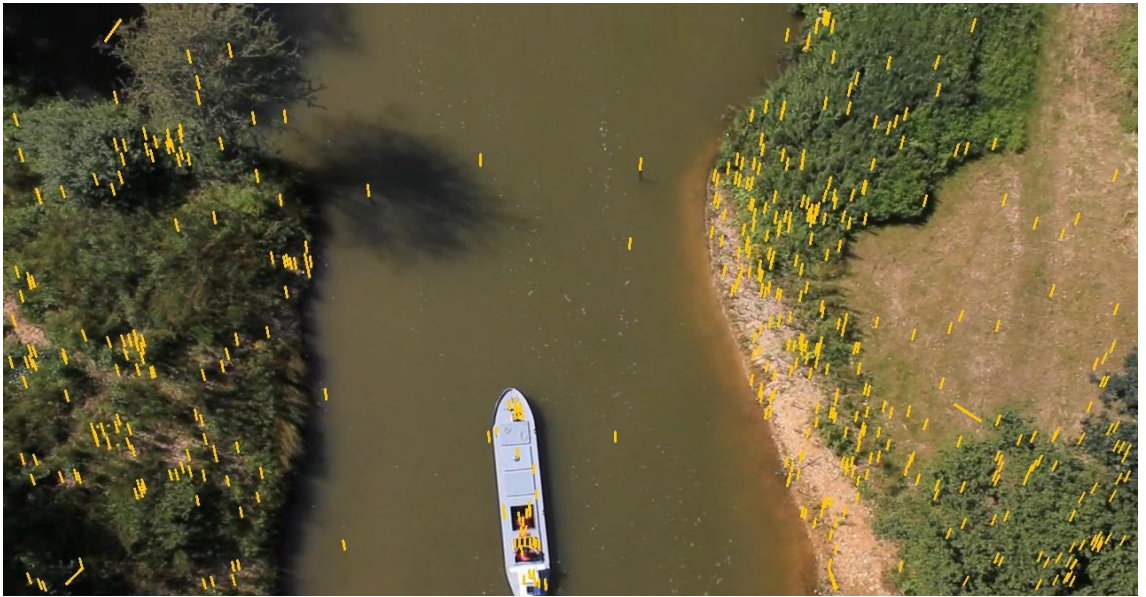


Şekil 4.6. Hava aracından alınan ilk görüntü



Şekil 4.7. Hava aracından alınan ikinci görüntü

SURF noktalarının eşleşme doğruluğu bu simülasyonda oldukça yüksek oranda olmuştur. Birinci görüntüde tespit edilen 430, ikinci görüntüde tespit edilen 378 anahtar noktadan 351 adet nokta doğru eşleşmiştir. İki sahne arasındaki noktaların tespit edilmeleri ve hatalı noktaların elenmesi toplamda 3,21 sn zamanda tamamlanmıştır. Sonuç vektörleri Şekil 4.8’ de görülmektedir.

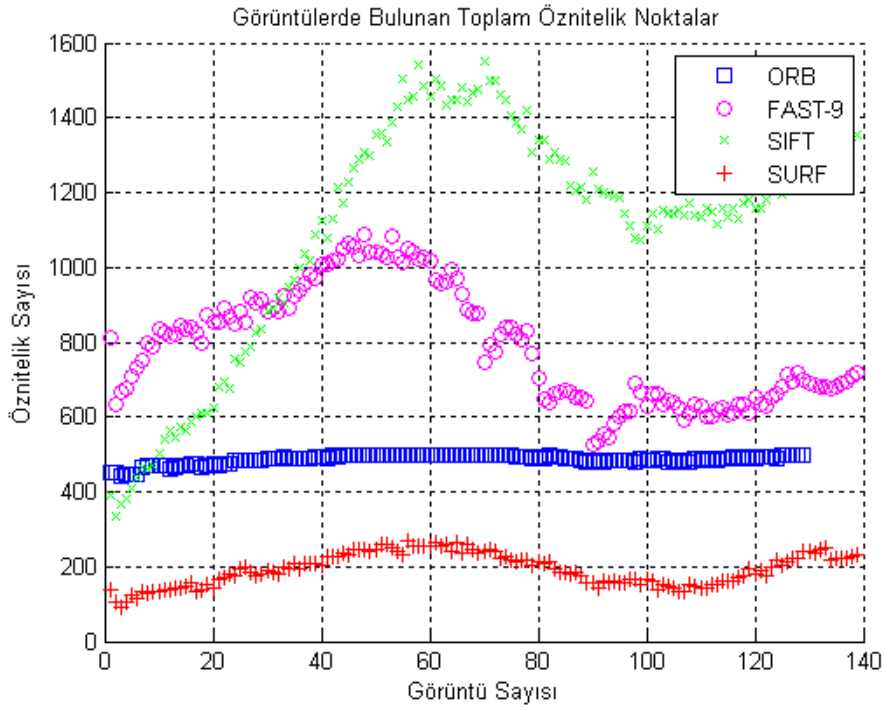


Şekil 4.8. SURF noktalarının eşleşmeleri yardımı ile hesaplanan hareket vektörleri

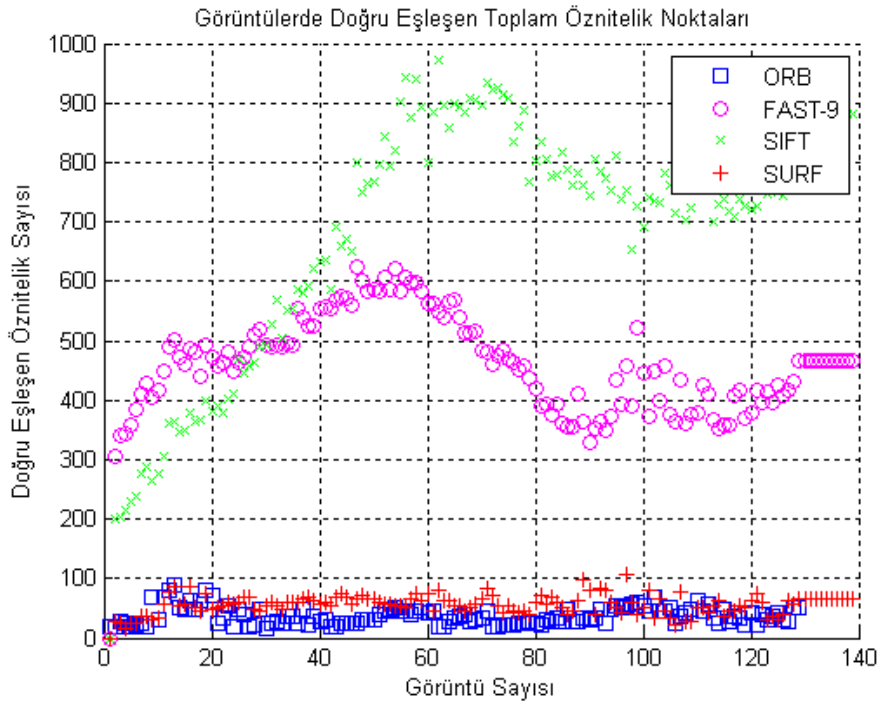
Örnek video üzerinden otomatik mozaikleme yöntemi ile oluşturulan görüntü Şekil 4.9’da verilmiştir. Yaklaşık 5 saniye süren videonun FPS değeri 30 sahne/saniye olmakla birlikte toplamda 140 sahne işlenmiştir. İşlenen her sahnenin birleştirme performansı için önceki örnekte olduğu gibi tespit edilen öznitelik sayısı ve doğru eşleşme sayısı Şekil 4.10 ve Şekil 4.11’de yer almaktadır. Görüntüye bütünleştirme işlemi gerçek zamanlı performansı görebilmek için uygulanmamıştır.



Şekil 4.9. Örnek UAV videosundan oluşturulan geniş açılı görüntü

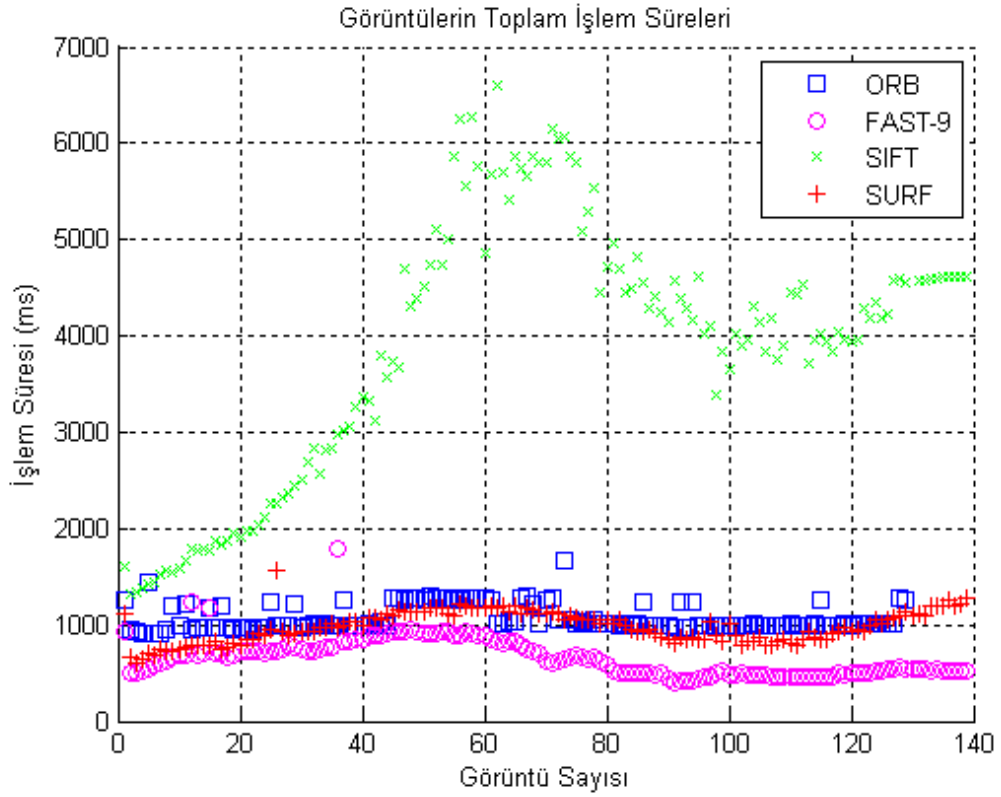


Şekil 4.10. UAV videosundaki çalışmada dört farklı yöntemin tespit ettiği toplam öznelik noktaları



Şekil 4.11. UAV videosundaki çalışmada dört farklı yöntemin doğru eşleşme performansı

Şekil 4.12' de ise uygulamadaki performans sonuçları yer almaktadır.



Şekil 4.12. UAV videosunda uygulanan dört farklı yöntemin zaman performansı

Öznitelik tespitinde SIFT noktaları yüksek sayıda öznitelik noktası verirken doğru eşleştirme performansıda oldukça yüksektir. Yüzey bilgisi ve sadece kenar noktalara yoğunlaşmaması açısından da hedef tanıma bilgileride taşımaktadır. Buna karşın işlem yükü çok fazla olduğundan gerçek zamanlı uygulamalar için pekte elverişli değildir.

SURF öznitelik noktaları ise SIFT'e göre daha hızlı olmakla birlikte eşleşme performansıda oldukça iyidir. Ancak rotasyonel değişimlere karşı fazla direnci olmadığı gözlemlenmiştir. Ayrıca ani kontrast değişimlerinden de kaliteyi düşürecek şekilde etkilenmektedir.

Şekil 4.13'te rotasyonel bir biçimde görüntü alan UAV videosunun ilk (sol) ve son(sağ) sahneleri görülmektedir. Bu görüntüler arasındaki yaklaşık 80 sahne yine üzerinde

çalıştığımız dört yöntemle otomatik birleştirme işlemine tabi tutulmuştur ve sonucu Şekil 4.14’te yer almaktadır.

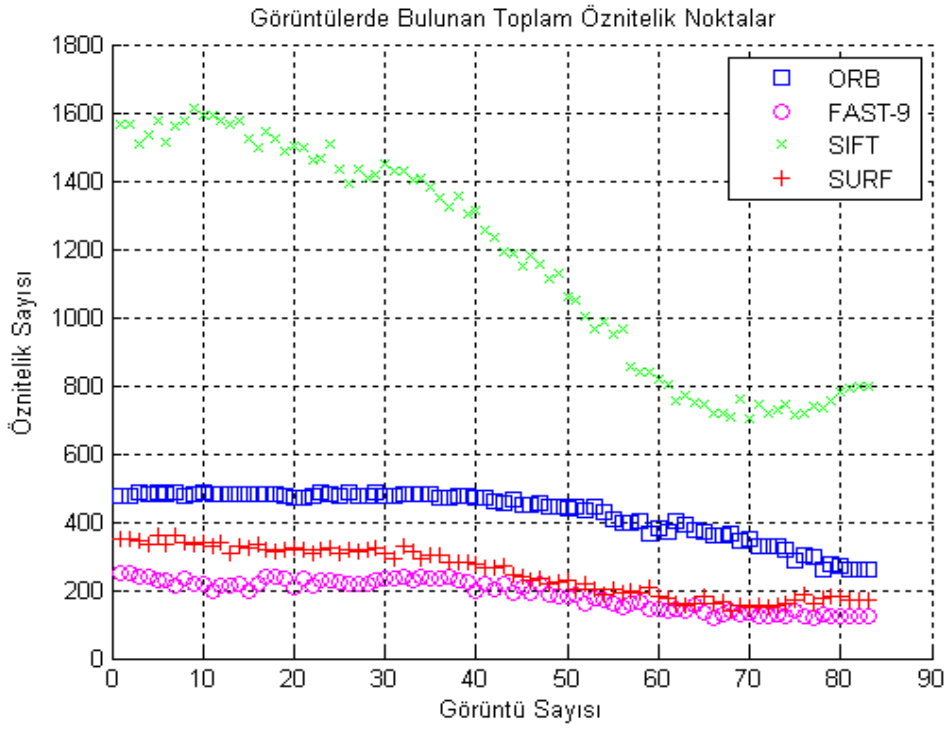


Şekil 4.13. Rotasyonel görüntü alan UAV videosunda ilk ve son sahneler

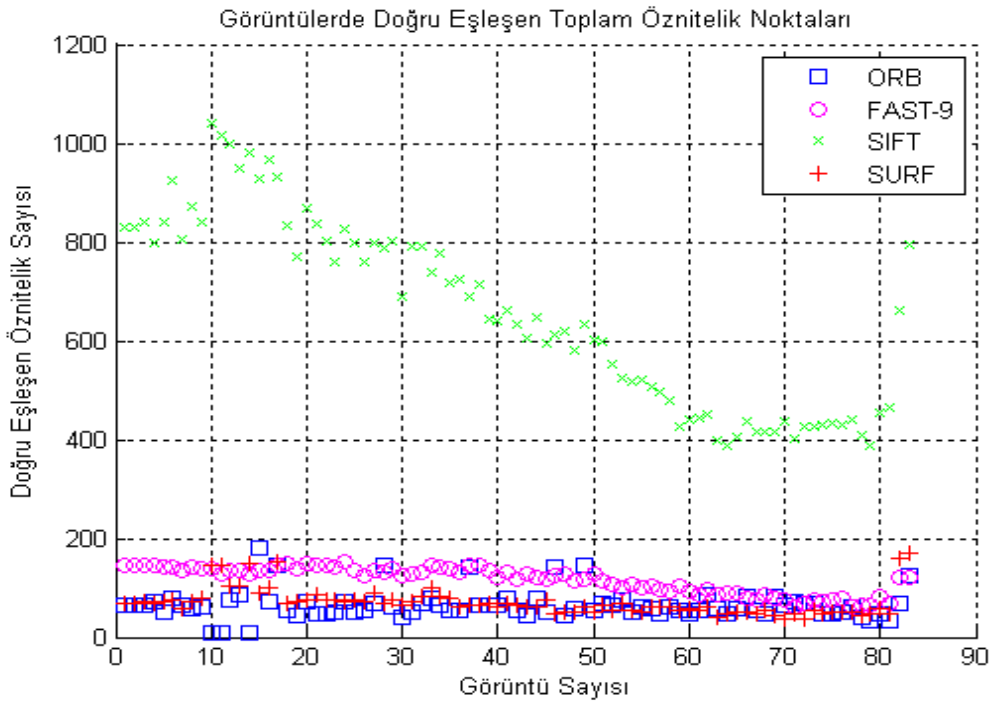


Şekil 4.14. Video üzerinden birleştirilmiş görüntü

Birleştirme işleminin performansına ait sonuçlar Şekil 4.15 ve Şekil 4.16’ de görülebilmektedir. Önceki örneğede benzer olarak FAST en yüksek hızda sonuç elde ederken en fazla öznetelik eşleşmesini yine SIFT sağlamıştır.

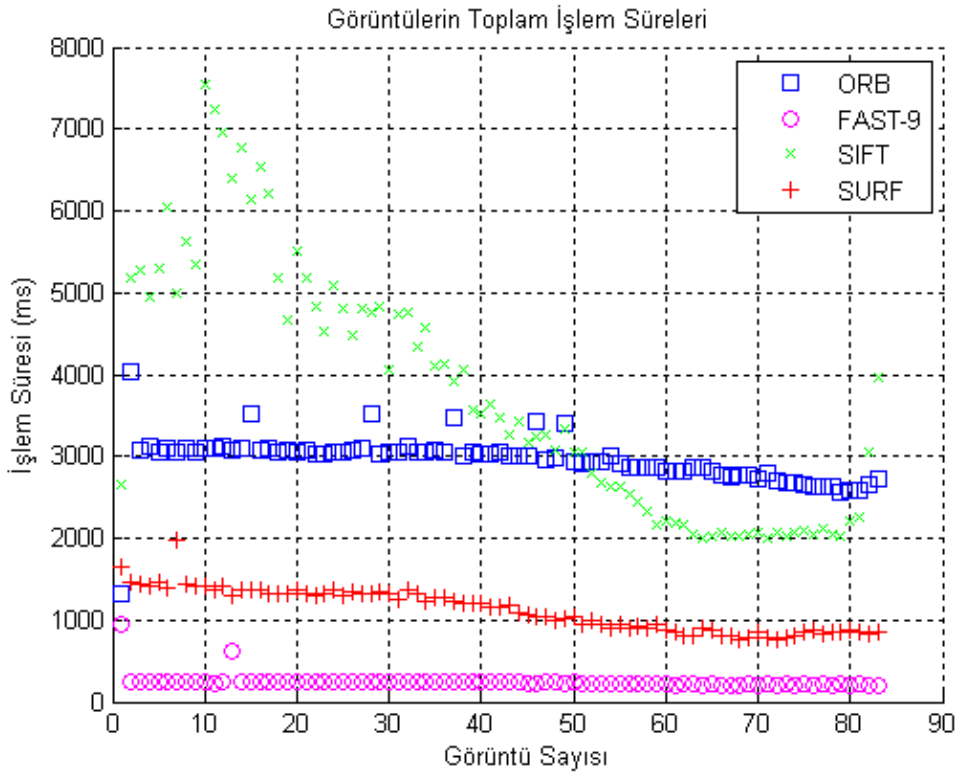


Şekil 4.15. Rotasyonel değişimin olduğu video üzerinden birleştirilmiş görüntüde dört yöntemin tespit ettikleri toplam öznelik nokta sayısı



Şekil 4.16. Rotasyonel değişimin olduğu video üzerinden birleştirilmiş görüntüde dört yöntemin doğru eşleşme performansı

Şekil 4.17’de dört yöntemde uygulamadaki toplam işlem süreleri yer almaktadır.



Şekil 4.17. Rotasyonel değişimin olduğu video üzerinden birleştirilmiş görüntüde dört yöntemin zaman performansı

Görülebileceği gibi ikinci uygulamada da sonuçlar, ilk uygulamaya benzerlik göstermiş; en fazla öznelik tespit ve eşleşmesini SIFT, en hızlı sonucu FAST yöntemi tespit etmiştir. Ancak SIFT’in rotasyonel değişim sürecinde tespit ettiği nokta ve eşleşme sayısının azalması da gözlerden kaçmamıştır. Diğer 2 yöntemde ise bu duruma daha fazla direnç göstermiştir. Yine de görüntünün kalitesinde ve dönüşüm matrisinin hesaplanmasında sayısal olarak ciddi bir farklılık görülmemiştir. ORB ise genel olarak diğer yöntemlere göre daha düşük eşleşme oranına sahip olmuştur ancak SIFT’te göre daha hızlı sonuç vermesi artı olarak görülebilir.

5. SONUÇ

Elde ettiğimiz araştırma sonuçları ve benzetimler gösteriyor ki, öznelik tabanlı birleştirme algoritmalarının performansı, öznelik noktalarının bulunması, değer atanması ve eşleştirilmesi sırasındaki performansıyla doğrudan alakalıdır. Görüntüdeki koordinat dönüşümleri hesapsal olarak fazla maliyetli olmadıklarından, öznelik noktalarının performansı sistemin kalitesi açısından neredeyse belirleyici rol almaktadır.

Bulgular ve tartışma bölümünde de kıyaslandığı üzere, SIFT, SURF ve FAST noktalarında kendine ait artıları ve eksileri vardır. Birbirinin yerine kullanılacak olan bu iki yöntemde, literatürde ve uygulamada çokça kullanılan yöntemlerdir. Gerçek zamanlı çalışmalara uygunluğu ve hızı açısından SURF bizim çalışmalarımızda da bir adım öne çıkmış, diğer yandan SIFT' te yüksek doğruluk oranı ile değerlendirmeye alınmıştır.

FAST beklendiği gibi diğer iki yönteme alternatif olarak çok daha hızlı sonuç vermektedir. Önerilmiş bir öznelik tanımlama sisteminin olmaması uygulamada bir eksik gibi görülsede son zamanlarda her uygulamaya farklı bir şekilde tanımlayıcılar geliştirilmiş ve her tanımlayıcı da kendi çalışma alanında tatmin edici sonuçlar verebilmektedir. Örneğin kendi çalışmalarımızda düz ilerleyen bir UAV videosunda farklı bir tanımlayıcı kullanmamızla birlikte, rotasyonel değişimin olduğu bir videoda farklı tanımlayıcı kullanmak daha elverişli olmuştur.

Otomatik mozaikleme uygulamalarında çok yüksek oranda eşleşme hassasiyeti gerekmediğinden, FAST her uygulamada geniş açılı görüntü oluşturmada istediğimiz performansı gösteren yöntem olmuştur. Ancak parlaklık değişimine fazla direnç göstermemesi de dikkatimizi çekmekle birlikte, basit adaptasyonu ve çoğu uygulamada hız bakımından iyi performansa sahip sonuçlar vermesi son derece dikkat çekicidir.

ORB metodu, son zamanlarda sıklıkla kullanılan yöntemlerin başında gelmesine rağmen, bizim uygulamalarımızda yeterli sonucu gösterememiştir. Hızlı tespit ve tanımlama vektörü çıkartmasına rağmen, öznelik noktalarının eşleşmelerinde yeterli

kalitede sonuç veremediği için çoğu zaman daha fazla öznitelik noktası tespit etmesi için parametreler yüksek değerli seçilmiştir. Bu durum hız performansında belirgin ölçüde etkilediğinden gerçek zamanlı uygulamalarda kullanılmasını zorlaştırmıştır.

Bütünleştirme işlemleri, her sistem için uzun hesaplama süreleri isteyen uygulamalardır. Bundan dolayı gerçek zamanlı otomatik mozaikleme yapan sistemlerde kullanılmaları pek mümkün olmamıştır, bundan dolayı gömülü sistemler üzerinde bütünleştirme işlemi yapan donanım tasarımı da yok denecek kadar azdır. Ancak sabit veri kümesi üzerinde zaman sınırı olmayan sistemlerde görüntü kalitesini son derece arttırdıklarından vazgeçilmez birer araç olmuşlardır. Benzer şekilde de doğru dönüşüm modelini bulmak için kullandığımız RANSAC algoritmasında performansı etkileyen adımlardan birtanesidir. İleriki çalışmalarda PROSAC (Progressive Sample Consensus) ile değiştirilmesi düşünülebilir.

İleriki çalışmada, bu çalışmada elde ettiğimiz bulgular eşliğinde tespit ettiğimiz en başarılı çözümün donanım üzerinde tasarlanması yer alacaktır. Özellikle paralel işlem kabiliyetine sahip bir donanımda hız konusunda problem yaşayan ancak sağlıklı sonuç veren yöntemler de uygulanabileceği gibi, bütünleştirme işlemide hızlı bir şekilde uygulanarak daha kaliteli sonuçlar elde edilebilir.

KAYNAKLAR

- Bay, H., Ess, A., Tuytelaars, T., Gool, L.V. 2008.** SURF: speeded up robust features. *Computer Vision and Image Understanding*, 110(3): 346-359.
- Brown, M., Lowe, D. G. 2002.** Invariant features from interest point groups. British Machine Vision Conference, 2-5 September 2002, Cardiff, UK.
- Burt, P., Adelson, E. 1983.** The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4): 532-540.
- Calonder, M., Lepetit, V., Strecha, C., Fua, P. 2010.** BRIEF: binary robust independent elementary features. 11th European Conference on Computer Vision, 5-11 September 2010, Hersonissos, Heraklion, Crete, Greece.
- Crowley, J. L., Stern, R. M. 1984.** Fast computation of the difference of low pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2): 212-222.
- Drummond, T., 2012.** High speed matching and tracking.
<http://www.qualcomm.com/media/events/high-speed-matching-and-tracking> (Erişim Tarihi: 02.01.2013)
- Evans, C. 2009.** Notes on the OpenSURF Library.
<http://www.chrisevansdev.com/computer-vision-opensurf.html> (Erişim Tarihi:02.01.2013)
- Fischler, M., A., Bolles, R. C. 1981.** Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 381-395.
- Hartley, Z., Zisserman, A. 2004.** Multiple View Geometry in Computer Vision. Cambridge University Press, UK, 672 pp.
- Harris, C., Stephens, M.J. 1988.** A combined corner and edge detector. Fourth Alvey Vision Conference, Manchester, UK.
- Irani, M., Anandand, P. 1999.** About direct methods. *Vision Algorithms: Theory and Practice*, 267-277.
- Lowe, D. G. 1999.** Object recognition from local scale-invariant features. 7th International Conference on Computer Vision, 20-27 September, 1999, Corfu, Greece.
- Lowe, D. G. 2004.** Distinctive Image Features from Scale Invariant Keypoints. *International Journal of Computer Vision*, 60(2): 91-110.

- Perez, P., Gangnet, M., Blake, A. 2003.** Poisson image editing. *ACM Transactions on Graphics*, 313-318.
- Ramachandran, M., Chellappa, R. 2006.** Stabilization and mosaicing of airborne videos. In Proceedings of the International Conference on Image Processing, October 8-11 2006, Atlanta, Georgia, USA.
- Ruble, E., Rabaud, V., Konolige, K., Bradski, D. 2011.** ORB: An Efficient Alternative to SIFT or SURF. In International Conference on Computer Vision, 8-11 November, 2011, Barcelona, Spain.
- Rosin, P. L. 1999.** Measuring corner properties. *Computer Vision and Image Understanding*, 73(2): 291 – 307
- Rosten, E., Drummond, T. 2006.** Machine learning for high-speed corner detection. http://www.edwardrosten.com/work/papers.html#rosten_2006_machine (Erişim Tarihi: 02.01.2013)
- Rosten, E., Drummond, T. 2009.** Multiple Target Localisation at over 100 FPS. British Machine Vision Conference, September 2009, London, UK.
- Rosten, E., Drummond, T., Taylor, S. 2009.** Robust feature matching in 2 μ s. Computer Vision and Pattern Recognition Workshops, 20-25 June, 2009, Cambridge, UK
- Shi, T., Tomasi, C. 1994.** Good features to track. 9th IEEE Conference on Computer Vision and Pattern Recognition, June 21-23, Seattle, USA.
- Szeliski, R. 2006.** Image Alignment and Stitching: A Tutorial Foundations and Trends in Computer Graphics and Vision, Microsoft Research, USA, 103pp.
- Torr, P. H. S., Zisserman, A. 1999.** Feature based methods for structure and motion estimation: Vision Algorithms Theory and Practice, Ed: Triggs, W., Zisserman, A., Szeliski, R., LNCS, pp: 278-295.
- Viola, P., Jones, M. 2001.** Rapid object detection using a boosted cascade of simple features. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 8-14 December 2001, Hawaii, USA.
- Yi, Z., Zhiguo, C., Yang, X. 2008.** Multi-spectral remote image registration based on SIFT. *IET Electronics Letters*, 44(2): 107-108

ÖZGEÇMİŞ

Adı Soyadı : Mehmet EFENDİOĞLU
Doğum Yeri ve Tarihi : Bursa, 18.12.1986
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Bursa Cumhuriyet Lisesi, 2005
Lisans : Uludağ Üniversitesi, 2009

Çalıştığı Kurum ve Yıl : Yüksel Savunma Sistemleri, 2011-...
İletişim (e-posta) : m.efendioglu@outlook.com