

**BAŐKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİR GÖRÜNTÜDEKİ NESNENİN BİR BAŐKA GÖRÜNTÜDE
BULUNMASI**

FUNDA HANİFE ÇETİN

YÜKSEK LİSANS TEZİ

2011

**BİR GÖRÜNTÜDEKİ NESNENİN BİR BAŞKA GÖRÜNTÜDE
BULUNMASI**

**RECOGNITION OF A PICTURED OBJECT IN DIFFERENT
IMAGES**

FUNDA HANİFE ÇETİN

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK-ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2011

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma, jürimiz tarafından **ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :.....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye (Danışman) :.....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye (Var ise Eş Danışman) :.....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye :.....(İmza).....
(Unvanı, Adı ve Soyadı)

ONAY

Bu tez/...../..... tarihinde, yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

...../...../.....

Prof.Dr. Emin AKATA

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

TEŐEKKÜR

Bu tez alıŐmasının ortaya ıkmasında en byk emeĐi olan, tecrbe ve zamanını benimle paylaŐan sevgili hocam ve tez danıŐmanım Prof. Dr. Emin Akata'ya;

Bilgi ve birikimlerini, deĐerli vakitlerini benimle paylaŐan sevgili hocalarım, Yrd. Do. Dr. Mustafa DoĐan'a, Do. Dr. Ali Ziya Alkar'a, Yrd. Do. Dr. Mehmet Demirer'e ve Yrd. Do. Dr. Umut Sezen'e;

Sevgili arkadaŐım Yk. Mh. Atilla zgr'e ve İbrahim Canpunar'a;

Aileme ve canım eŐim Umut etin'e, bana verdikleri moral, mutluluk ve destek iin teŐekkr ederim.

ÖZ

BİR GÖRÜNTÜDEKİ NESNENİN BİR BAŞKA GÖRÜNTÜDE BULUNMASI

Funda Hanife ÇETİN

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Nesne tanıma üzerinde uzun yıllardır yapılan çalışmalarda, farklı birçok yöntem denenmiştir, fakat kullanılan yöntemler, gerçek zamanlı uygulamalar veya gerçek ortam resimlerinde istenildiği kadar etkili sonuçlar verememektedirler. Kesin tanıma yapma amacıyla, nesnelerin bölgesel veya genel özelliklerine göre ayrı ayrı veya bu özelliklerin bir arada kullanılması fikrine dayanan yöntemler üzerinde araştırmalar devam etmektedir.

Bu çalışmada, fotoğraflanmış bir nesneyi, bir başka arama ortamında, oryantasyondan bağımsız olarak, fotoğrafma açısı, ölçek ya da aydınlanma farklılığı, gürültü ve bozulmadan etkilenmeden tanıyabilecek bir yöntem geliştirilmiştir. Geliştirilen bütünleştirilmiş nesne tanıma yöntemi, öncelikle nesnenin bölgesel özelliklerini kullanarak, resimler arasında benzer noktaları eşleme ve sonrasında eşleşen noktalar etrafındaki bölgede benzerlik arayarak nesneyi bulma işlemi yapmaktadır. Bütünleştirilmiş nesne tanıma uygulaması ile, kullanılan yöntemlerin tüm üstünlüklerinden yararlanılarak, tanıma işleminin başarısı artırılmıştır.

Anahtar Sözcükler: Nesne tanıma, bölgesel ayırt edici öznitelikler, SIFT, şablon eşleme.

Danışman: Prof. Dr. Emin AKATA, Başkent Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü.

ABSTRACT

RECOGNITION OF A PICTURED OBJECT IN DIFFERENT IMAGES

Funda Hanife ÇETİN

Başkent University Institute of Science

Department of Electrical and Electronics Engineering

Throughout the years there have been several different methods used for object recognition. However the methods used are far from providing satisfactory results for real time applications and for real environment images. Researches for accurate recognition of objects, in relation to their regional or general features, are currently being carried on.

In this study, a methodology for the recognition of a pictured object in a different natural environment, regardless of its orientation, illumination conditions, picturing angle, picture size, noise and distortion has been developed. The proposed methodology finds objects by initially using the object's local features to match the similar points between the images, then searches similarities around the neighborhood of these matched points. The accuracy of the recognition, proved to attain a considerable increase, by means of the proposed methodology.

Keywords: Object recognition, local distinctive features, SIFT, template matching.

Advisor: Prof. Dr. Emin AKATA, Başkent University Department of Electrical and Electronics Engineering

İÇİNDEKİLER LİSTESİ

Sayfa

TEŞEKKÜR	i
ÖZ.....	ii
ABSTRACT	iii
ŞEKİLLER LİSTESİ	vi
ÇİZELGELER LİSESİ.....	viii
SİMGELER VE KISALTMALAR LİSESİ	ix
1. GİRİŞ.....	1
2. NESNE TANIMA	3
2.1 Problem Tanımlama.....	4
2.2 Genel Yaklaşımlar	7
2.3 Resimlerin Bilgisayar Ortamında Değerlendirilmesi	12
2.4 Nesneyi Tanımlayan Öznitelikler.....	13
2.5 Oryantasyondan Bağımsız Öznitelik Çıkarımı.....	15
2.5.1 Ölçek uzayında uçdeğer kestirimi.....	16
2.5.1.1 Oktav ve ölçek kavramları	16
2.5.1.2 Gauss farkları piramidi oluşturulması.....	18
2.5.1.3 Bölgesel uçdeğer tespiti.....	18
2.5.2 Anahtar nokta konumlandırma.....	20
2.5.2.1 Kararsız uçdeğerlerin elenmesi	22
2.5.3 Anahtar noktalara yönelim ataması	24
2.5.4 Bölgesel resim tanımlayıcılar oluşturulması.....	28
2.6 Şablon Kullanarak Eşleme	32
2.6.1 Şablon ile eşlemenin genel özellikleri	33
2.7 Bütünleştirilmiş Nesne Tanıma.....	35
2.8 Benzerlik Ve Sınıflandırma.....	36
3. YAPILAN ÇALIŞMALAR	40
3.1 Bütünleştirilmiş Nesne Tanıma Yönteminin Uygulanması.....	42
3.2 Veriseti Üzerinde Elde Edilen Sonuçlar.....	58

3.2.1 Anahtar nokta eşleştirme başarısı	61
3.2.2 B.N.T. yönteminin başarısı	65
4.SONUÇLAR	70
4.1 Gelecek Çalışmalar Planı.....	74
KAYNAKLAR LİSTESİ.....	75
EKLER.....	80
EK 1. İlgin Dönüşümler.....	80
EK 2. DoG ve LoG Fonksiyonlarının Benzerliği.....	82
EK 3. Ara Değer Kestirimi	84
EK 4. Pikseller Arası Komşuluk Araştırılması.....	87
EK 5. B.N.T. Yöntemine İlişkin Diğer Çıktı Örnekleri.....	89
EK 6. Başarı Testlerinde Kullanılan Resimler.....	123

ŞEKİLLER LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 Nesne tanıma temel aşamaları	4
Şekil 2.2 Nesne tanımayı zorlaştıran durumlar	5
Şekil 2.3 $m \times n$ pikselden oluşan bir sayısal resimde, (i, j) noktasındaki gri seviye genlik değeri $(f(i, j))$	12
Şekil 2.4 $m \times n$ pikselden oluşan bir sayısal resmin, Matlab ortamındaki ifadesi	13
Şekil 2.5 Bir dairenin bakış yönü değiştirildiğinde elips şeklinde görünmesi	14
Şekil 2.6 Ölçek, oktav kavramlarının gösterilmesi ve Gauss farkının alınması	17
Şekil 2.7 Uçdeğerlerin aranmasında kullanılan komşu pikseller	19
Şekil 2.8 Uçdeğerlerin en alt ve en üstteki ölçeklerde aranmaması	20
Şekil 2.9 Resim üzerinde yapılan dönüşüm sonucu ara değerlerin oluşması	21
Şekil 2.10 Uçdeğerlerin bulunabileceği konumlar	23
Şekil 2.11 Anahtar nokta etrafında eğimlerin inceleneceği bölge	25
Şekil 2.12 Araştırma bölgesinde eğim genlik ve yönelimi	26
Şekil 2.13 Anahtar nokta civarında kuantalama	27
Şekil 2.14 Anahtar nokta etrafında 128 boyutlu vektör oluşturulması	29
Şekil 2.15 4×4 'lük alt pencereler içinde eğim genlik ve yönelimleri histogramının oluşturulması	29
Şekil 2.16 Ağırlık matrisinin resim üzerine uygulanışı	30
Şekil 2.17 Gauss ağırlık fonksiyonunun eğim genlik ve yönelimlerine uygulanışı.	30
Şekil 2.18 (i, j) koordinatında bulunan $K5$ merkez pikseli ve komşularının $K5$ 'e göre gösterimi	32
Şekil 2.19 Lena resmi üzerinde şablon eşleme algoritmasının uygulanması	33
Şekil 2.20 Yazı karakteri tanıma amacıyla şablon eşleme algoritmasının uygulanması	34

Şekil 2.21 B.N.T. yönteminin ana parçaları.....	35
Şekil 2.22 İki kategoriye sınıflandırma	36
Şekil 3.1 B.N.T yöntemine ait akış diyagramı	41
Şekil 3.2 Model resim üzerinde kenar bulma	42
Şekil 3.3 Model resim üzerinde ön işlemeden sonra kenar bulma	43
Şekil 3.4 Model resimde ön işlemler ve kenar iyileştirme.....	44
Şekil 3.5 Anahtar noktaların belirlenmesi.....	45
Şekil 3.6 Öznitelik vektörlerinin model nesne üzerinde gösterilmesi.....	47
Şekil 3.7 Öznitelik vektörlerinin ilgili resimler üzerinde eşleme için kullanılması ...	48
Şekil 3.8 Eşleşme miktarının vektörler arasında aranan fark ile değişimi	50
Şekil 3.9 SIFT öznitelikleri kullanarak muhtemel nesne konumlarının eşleştirme ile gösterilmesi.....	51
Şekil 3.10 Eşleşen noktaların etrafında belirlenen tüm bölgeler	52
Şekil 3.11 Şablon ve arama ortamı resimleri üzerinde seçilen ve kırılan bölgeler	52
Şekil 3.12 Benzerliğinin en çok olması beklenen arama bölgesi.....	53
Şekil 3.13 Şablon eşleme için kullanılacak şablon ve arama ortamı.....	54
Şekil 3.14 Şablon eşlemede ilişikliğin frekans uzayında gösterilmesi	55
Şekil 3.15 İlişiklik değerlerinin bazı eşik değerlerinden geçirilmesi	56
Şekil 3.16 Seçilen model ile benzerliği en çok olan noktanın orjinal arama ortamı resminde gösterilmesi.....	57
Şekil 3.17 Performans testlerinde kullanılan model resimler.....	58

ÇİZELGELER LİSESİ

- Çizelge 3.1 Performans değerlendirmesi için kullanılan karışıklık matrisi
- Çizelge 3.2 Birinci nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.3 İkinci nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.4 Üçüncü nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.5 Dördüncü nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.6 Beşinci nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.7 Altıncı nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.8 Yedinci nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.9 Sekizinci nesne için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.10 Tüm nesnelere için, nokta eşleştirme performans değerlendirmesi
- Çizelge 3.11 Birinci nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.12 İkinci nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.13 Üçüncü nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.14 Dördüncü nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.15 Beşinci nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.16 Altıncı nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.17 Yedinci nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.18 Sekizinci nesne için, B.N.T. performans değerlendirmesi
- Çizelge 3.19 Tüm nesnelere için, B.N.T. performans değerlendirmesi

SİMGELER VE KISALTMALAR LİSESİ

B.N.T.	Bütünleştirilmiş Nesne Tanıma
SIFT	Scale Invariant Feature Transform
PCA	Principal Components Analysis
n-SIFT	n-dimensional SIFT
PCA-SIFT	Principal Components Analysis with SIFT
MRI	Magnetic Resonance Imaging
CT	Computed Tomography
DoG	Difference of Gaussian
DoM	Difference of Means
LoG	Laplacian of Gaussian
SAD	Sum of Absolute Differences
SSD	Sum of Squared Differences
MaxAD	Maximum Absolute Difference
TP	True-Positive
FN	False-Negative
FP	False-Positive
TN	True-Negative
SDT	Similar Distance Threshold
LSM	Least Square Method
D_i	Uzaklık ölçüm sonucu
cor	İlişiklik katsayısı
$L(x,y, \sigma)$	Resmin ölçek uzayı fonksiyonu
$G(x,y, \sigma)$	Değişken ölçekli Gauss fonksiyonu
$I(x,y)$	Algoritmaya giren resim
$m(x,y)$	Eğim genlik değeri
$\theta(x,y)$	Eğim yönelimi
σ	Gauss fonksiyonunda ölçek parametresi, bulanıklaştırma miktarıdır.
$D(x,y,\sigma)$	Ölçek uzayı fonksiyonu
α	Genliği büyük olan özdeğer
β	Genliği küçük olan özdeğer
n	Bulunan anahtar nokta sayısı
I	Arama yaptığımız resim

x	Şablon resim
g	$n \times m$ ebatlarındaki şablon resim
(r,c)	Şablon resmin sol üst köşesinin koordinatı
\bar{x}	Şablon resimdeki ortalama gri seviyesi
y	Arama yapılan resmin, ilgili parçası
\bar{y}	Arama yapılan resmin ortalama gri seviyesi
N	Zamanda şablon resmin alanı (satır x sütun)
x	Yerel örnek noktası
\hat{x}	Uçdeğerin konumu
$D(\hat{x})$	Uçdeğerin fonksiyon değeri

1. GİRİŞ

Bilgisayarlı görme uygulamalarında, nesne tanıma algoritmalarına sıklıkla yer verilir. Nesne tanıma, medikal görüntü işlemede tanı koyma, savunma sanayiinde güvenlik veya hedef tanıma gibi amaçlarla birçok farklı alanda kullanılmaktadır [17]; [59]. Kullanılacak nesne tanıma yöntemini, kullanım amacı belirleyecektir. Nesne tanıma görünüş tabanlı veya öznitelik tabanlı olarak yapılabilmektedir [60].

Görünüş tabanlı nesne tanımda, nesneye ait şablon veya model olarak belirlenmiş bir resim kullanılır. Bu tür tanımlar sabit koşullar altında başarılıdır. Aynı nesne farklı resimlerde, bakış açısındaki farklılıklardan veya kısmi olarak görünümünün engellenmiş olmasından, hatta sadece aydınlanma farklılıklarından dolayı tamamen farklı bir nesne gibi algılanabilir.

Amacımız doğal resimlerde nesne tanımak olduğundan koşulların değişkenliğini kabul ederek, yani, farklı bakış açılarında, farklı ölçeklerde, nesne başka nesnelere tarafından engellenirken ve kaçınılmaz olarak aydınlanma farklılıkları varken, nesne tanımayı gerçekleştirmektir. Bu koşullar altında, tanıma yöntemi olarak öznitelik tabanlı yöntemler düşünülmüştür.

Öznitelik tabanlı nesne tanıma yöntemleri ise, nesnenin global veya bölgesel özniteliklerinin kullanımına göre kendi içinde ayrılır [27]. Değişken koşulların varlığı ve en önemli zorlayıcı faktör olan nesnenin başka nesnelere tarafından engellenmiş olma durumu sebepleriyle, bölgesel özniteliklerden faydalanarak tanıma işlemi gerçekleştirilmeye çalışılmıştır. Böylece nesnelere bütününden bağımsız olarak aranabilir olmuştur.

Bu tez çalışmasının diğer bölümleri şu şekilde düzenlenmiştir; ikinci bölümde, problemin tanımlanmıştır ve bu alanda daha önce yapılmış çalışmalar özetlenmiştir.

Üçüncü bölümde, çalışma ortamına resimlerin aktarılmasından ve resimlerin ayırt edici tanımlayıcılarından bahsedilmiştir. Nesne ayırt edici tanımlayıcısı olan öznitelik vektörünün çıkarılmasında kullanılacak, ölçekten bağımsız öznitelik

ıkarım yntemi, ařamalarıyla aıklanmıřtır. ıkarılan zneliklerin zellikleri ve stnlklerinden bahsedilmiřtir. Yine bu blm ierisinde, řablon eřleme algoritması incelenmiř ve bu algoritmanın istenilen uygulamalar iin tek bařına yetersizlięi anlatılmıřtır.

nc blm altında ayrıca, bahsedilen yntemlerin bir arada kullanılması ile geliřtirilen bir btnleřtirilmiř nesne tanıma (B.N.T.) yntemi nerilmiř ve bu yntemin problemin zmne getirdięi stnlkler incelenmiřtir.

Drdnc blmde, B.N.T. yntemi, bir veri seti zerinde denenmiř ve elde edilen sonular, performans deęerlendirmeleri de yapılarak verilmiřtir.

2. NESNE TANIMA

Nesne tanımayı iki alt başlığa ayrılabiliriz.

Görünüş tabanlı nesne tanıma: Nesneye ait bir şablon veya model olarak belirlenmiş bir resim kullanılır. Şablon, arama ortamında gezdirilerek uygun düşen eşleşmeler, mesafe ölçümlerinden de faydalanılarak aranır [41]; [17].

Öznitelik tabanlı nesne tanıma: Nesnenin global veya bölgesel özniteliklerinin kullanımına göre kendi içinde ayrılır. Nesneye ait öznitelikler, arama ortamından çıkarılan öznitelikler ile karşılaştırılır ve uygun düşen eşleşmeler, mesafe ölçümlerinden de faydalanılarak aranır [7].

Görünüş tabanlı nesne tanıma, nesnenin görünüşünün öneminin büyük olmasından dolayı, görünüşü etkileyen durumlarda nesne tanıma zorlaşmaktadır. Dolayısı ile doğal ortam resimleri ile çalışırken ve dinamik uygulamalarda bu yöntemin uygulanabilirliği azdır.

Öznitelik tabanlı nesne tanıma yöntemlerinde, değişmez özniteliklerin aranması esastır. Mevcut koşullar düşünülerek, hangi durumlarda değişmezlik beklendiği belirlenmeli ve beklentileri karşılayacak şekilde öznitelikler seçilmelidir. Özniteliklerin global veya bölgesel seçilmesine de duruma göre karar verilir. Örneğin, resimdeki bir insan yüzü olup olmadığına karar vermek için global özniteliklerden, resimdeki insanların yüzlerinden kimlik tespiti yapılmak isteniyorsa, bölgesel özniteliklerden faydalanmak doğrudur.

Global öznitelikler: Nesnenin tanıma, nesneyi tanımlayan global özellikler kullanılacak ise tercih edilebilir. Global tanımlayıcılar, nesnenin genel renk yapısı ve renk dağılımları, homojen doku yapısı, nesnenin dış hatları ve merkezi gibi tanımlayıcılardır [17].

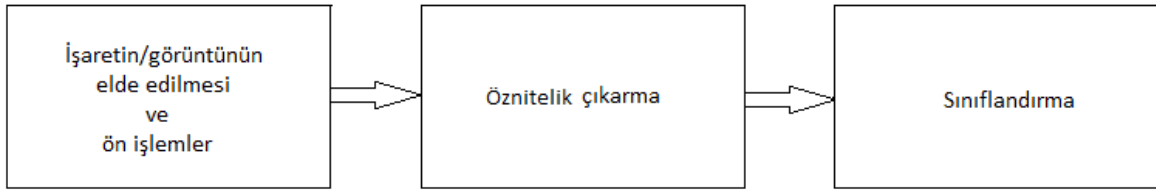
Bölgesel öznitelikler: Global özelliklerin tanımlamada yetmediği, nesnenin engellenmesi gibi durumlarda, bölgesel öznitelikler kullanılmalıdır. Bölgesel

tanımlayıcılar; merkeze olan uzaklıklar, çevreyi oluşturan eğrilerin eğrilik dereceleri, köşeler gibi tanımlayıcılardır [33].

Global ve bölgesel öznitelikler bir arada da kullanılabilir.

Doğal ortamda gözlenen bir nesnenin fotoğrafı (model) ile, bu nesneyi arayacağımız başka bir doğal ortam fotoğrafı (arama ortamı) elimizde olduğunda, yapılacak işlemler sırası ile; model ve arama ortamına ait özniteliklerin çıkarılması, model nesneye ait özniteliklerin, arama ortamında elde edilen öznitelikler ile karşılaştırılması, sonuçta da bir sınıflandırma yapılmasıdır [41]; [13].

Gerçek ortama ait bir fotoğraf algoritmaya girdi olarak verilir ve tanımayı sağlayacak ayırt edici öznitelikler belirlenir. En küçük farkın aranması ilkesine göre sınıflandırma yaparak, nesne tanıma gerçekleştirilir (Şekil 2.1).



Şekil 2.1 Nesne tanıma temel aşamaları

2.1 Problem Tanımlama

Doğal resimlerde, fotoğraflama sırasında ideal olmayan ortam koşulları ve farklı bakış açısından fotoğraflama sebebiyle nesne tanımayı zorlaştıracak bir çok durum oluşmaktadır. Şekil 2.2’de görüldüğü gibi, nesnenin boyutunun olduğundan farklı görünmesi, başka nesnelere tarafından görüntüsünün engellenmesi ve aydınlanma şiddeti ve aydınlanma yönü farklılıkları gibi etmenlerin tanıma işlemini engellememesi için, bu gibi durumlardan etkilenmeyecek şekilde çalışan bir tanıma algoritması kullanılmalıdır.

(a) Model nesnenin yüzeyinin karışık desenli olması



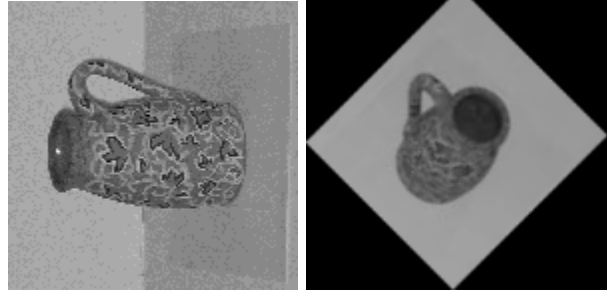
(b) Farklı koordinatlarda bulununan nesne



(c) Yeniden ölçeklenmiş nesne



(d) Çeşitli açılarda döndürülmüş nesne



(e) Başka nesneler ile görünümü kısmen engellenmiş nesne



Şekil 2.2 Nesne tanımayı zorlaştıran durumlar

(f) Aydınlanma farklılığı
olan nesne



Şekil 2.2 devam ediyor

Model olarak verilen bir nesnenin, farklı ortamlarda Şekil 2.2 'de görünen koşullar altında tanınabilmesini sağlayacak aloritmadan beklenenler şunlardır:

- Ötelemeden bağımsızlık: nesneyi, arama ortamında farklı koordinatlarda bulunurken, tanıyabilmek.
- Ölçekten bağımsızlık: nesneyi, arama ortamında olduğundan farklı büyüklüklerde görünürken tanıyabilmek.
- Döndürmeden bağımsızlık: nesneyi, arama ortamında x,y ve z düzlemlerine göre farklı açılındırmalar altında bulunduğunda tanıyabilmek.
- Kısmi engellenmeden bağımsızlık: nesneyi, başka nesnelere tarafından görünümü kısmen engellendiği durumlarda tanıyabilmek.
- Aydınlanmadan bağımsızlık: nesneyi, farklı aydınlanma koşulları altında bulunduğunda tanıyabilmek.

Öteleme, ölçekleme, döndürme gibi işlemler, noktaların düzlem üzerindeki konumlarını değiştiren dönüşümlerdir. Bu dönüşümler geometrik şekillerin farklı geometrik şekillere dönüşmesine sebep olabilir. Doğruları, birbirleriyle kesişim özelliklerini de değiştirmeden, başka konumlardaki doğrulara dönüştüren yani; çoklu doğruları yine çoklu doğrulara, çokgenleri yine çokgenlere dönüştüren dönüşümlere ilgin dönüşümler (affine transformations) denir [61], (Bkz. Ek 1).

2.2 Genel Yaklaşımlar

Bölgesel öznitelik seçimleri, doğru parçalarına, kenar veya bölgesel gruplandırmalara bağlı olarak yapılagelmiştir. Noktalar üzerinde çalışma fikri doğduğunda ise, daha ayırt edici olmaları sebebiyle çalışmalar köşe noktaları üzerinde yoğunlaşmıştır.

Bölgesel ilgi noktaları kullanarak resim eşleştirme çalışmalarına, 1981 yılında Moravec'in köşe sezinleyicisi (corner detector) kullandığı çalışmasıyla başlanmıştır [35]. Moravec'in sezinleyicisi, resimdeki küçük detaylar ve birbirine yakın konumdaki kenarlar üzerinde daha iyi sonuçlar verecek şekilde Harris and Stephans [15] tarafından geliştirilmiştir. Harris sezinleyicisi hareket takibi ve üç boyutlu yapıların yeniden kazanımı (reconstruction) için kullanmıştır [16]. Bunun yanında Harris köşe sezinleyicisi birçok başka resim eşleştirme görevi için halen sıkça kullanılmaktadır. Bu tür özelliklere sahip sezinleyiciler genellikle köşe sezinleyicisi olarak adlandırılırlar fakat, bu sezinleyiciler sadece köşeleri seçmediklerinden, tüm yönlerde yüksek eğime sahip, önceden bilinen ölçeklerdeki, resimlerin konumlarının belirlenmesi için de tercih edilmektedirler. Bu yaklaşım, başlarda stereo ve kısa mesafe hareket takibi için kullanılmış, sonraları daha karmaşık problemler için de kullanılabilir biçimde geliştirilmiştir. Zhang et al. [56] benzer eşleşmeleri seçmek için, her köşenin etrafında ilişiklik penceresi (correlation window) kullanarak Harris köşelerinin geniş resim alanlarında da eşleştirme için kullanılabilirliğini göstermişlerdir. Zhang et al. [56], kaydetme (registration) için yine bu noktaları kullanmıştır. Daha sonra, geometrik kısıtlamaları tanımlayan ana matrisin çözülmesi ile, iki görünüm arasındaki hatalı eşleşmeler yani aykırılıklar ortadan kaldırılmıştır. Aynı zamanda geometrik kısıtlamalar Torr [52] tarafından, resmin içindeki hatları belirgin nesnelerin hareketindeki aykırılıkları ayıklamak amacıyla da kullanılmış ve uzun mesafe hareket eşleştirmeleri yapılmıştır.

Schmid & Mohr [47], da Harris sezinleyicisi kullanarak seçtikleri noktalar ile bölgesel resim tanımlayıcıları oluşturmuşlar ve yönelimden bağımsız vektörler tanımlayarak, başarılı sonuçlar elde etmişlerdir. Bölgesel değişmez öznitelik eşleştirmesinin geliştirilmesi ile geniş resim veritabanları içinde öznitelik aramasını

da başarıyla yapmışlardır. İlgili noktalarının seçilmesinde Harris köşelerini ve eşleştirmede, ilişiklik penceresi yerine, yerel resim bölgesinin döndürmeden bağımsız tanımlayıcılarını kullanmışlardır. Bu özniteliklerin, iki resim arasındaki rastgele yönelim değişimleri altında eşleştirilmesine izin verir. Üstelik, kısmi engelleme (occlusion) ve karışık ortamda bulunma durumlarında bile, çok sayıda öznitelik eşleştirmesinin, eşleşmiş özniteliklerin üzerinde benzerlik sınıflandırması yapılmasıyla, genel tanıma yapılabileceğini göstermişlerdir.

Harris köşe sezinleyicisi, resmin ölçek değişimlerine karşı hassas olduğundan, farklı ölçeklerdeki resimlerin eşleştirilmesi için iyi bir çalışma ortamı oluşturmamaktadır. Lowe [25] çalışmasında, ölçekten bağımsızlığı yakalamak üzere, bölgesel tanımlayıcılar kullanmıştır.

Crowley and Parker [9] ölçek değişimi altında kararlı tanımlayıcılar üzerine ilk çalışmaları, betimlenen sivri uç ve tepelerin ölçek uzayında ifadesi ve bir ağaç yapısıyla buların birbirine bağlanması olarak yapmışlardır. Daha sonra bu ağaç yapısı, rastgele ölçeklerde resimler ile eşleştirme olanağı sağlamıştır. Grafik tabanlı eşleştirme çalışmaları altında Shokoufandeh, Marsic and Dickinson [50], dalgacık katsayılarını kullanan, daha ayırt edici öznitelik tanımlayıcıları elde etmişlerdir. Lindeberg [23]; [24] öznitelik çıkarımında istikrarlı ve uygun bir ölçek tanımlama üzerine yoğun olarak çalışmıştır. Bunu bir ölçek seçimi problemi olarak tanımlamıştır.

Son zamanlarda, bütün ilgin dönüşümleri için değişmez olan bölgesel öznitelik çıkarımları üzerine ilginç çalışmalar yapılmıştır, Baumberg [2], Tuytelaars and Van Gool [53], Mikolajczyk and Schmid [31], Schaffalitzky and Zisserman [44], Brown and Lowe [5] tarafından yapılan çalışmalar bunun tipik örnekleridir. Yapılan çalışmalarda düzlemsel bir yüzey üzerindeki özniteliklerin eşleştirilmesi için, çoğu kez resim bir bölgesel ilgin çerçevesine (local affine frame) yeniden ölçeklenmiştir fakat, ilgin dönüşüm altında tamamen bağımsızlık yine de yakalanamamıştır. Çünkü; resimlere ait tüm koordinat uzayını tarama işleminin maliyetinin büyüklüğü sebebiyle, başlangıç ölçek ve konumları, kullanılan ilgin dönüşüme bağlı yaklaşımlarla seçilmiştir.

Lowe [26]'a göre ilgin dönüşümlerden tümüyle bağımsızlık birçok uygulama için gerekli olmayabilir. Çünkü, üç boyutlu nesnelere doğrusal olmayan değişimleri ve engelleme etkilerini azaltmak için alınacak eğitim görselleri, nesnenin bakış açısının değiştirilmesi ile elde edilir ve elde edilecek en iyi görseller en azından her 30 derecede bir görüntülemeyi gerektirmektedir (tanımanın en yakın eğitim görselinin en azından 15 derecelik yakınında olması demektir).

Nesne sınırlarına yakın olan karışık arkaplanın nesne seçimini zorlaştırmasının önüne geçmek için, bazı öznitelik türleri resimlerin dış hatlarını veya bölge sınırlarını kullanır. Matas et al. [28], uç bölgelerin dengeli bir şekilde en yüksek kararlılıkta, yüksek sayılarda eşleşen öznitelikler üretebileceğini göstermiştir. Mikolajczyk et al. [31], üstüste geçmiş dar şekilli karmaşık arkaplanlarda bile kararlı öznitelikler bulmayı sağlayan, ilişkisiz yakın kenarları gözardı eden, yeni bir bölgesel kenar tanımlayıcısı geliştirmiştir.

Nelson and Selinger [38], resim dış hatlarının gruplanması temeline dayanan yerel özniteliklerle iyi sonuçlar elde etmişlerdir. Benzer olarak Pope and Lowe [40] da, özellikle yüzey detayı ve deseni (texture) az olan nesnelere, nesne dış hatlarının hiyerarşik gruplandırılması temeline dayanan öznitelikleri kullanmışlardır.

Diğer görünüm temelli tanıma yaklaşımlarında, Murase and Nayar [36], özuzay (eigenspace) eşleştirmelerini, Swain, ve Ballard [51], renk histogramlarını, Schiele, Bernt, and Crowley [45], alıcı alan (receptive field) histogramlarını kullanmışlardır. Tüm bu yaklaşımlar yalıtılmış nesnelere veya önceden parçalanmış nesnelere üzerinde başarılı bir şekilde çalışmaktadır. Fakat, karışık ve kısmen engellenen bulunan resimlerde, daha global özniteliklere ihtiyaç duyulmasından dolayı sorunlar yaşanmaktadır. Ohba and Ikeuchi [39], pek çok küçük yerel öz(eigen)-pencereler kullanarak, karışık resimlere özuzay yaklaşımını başarıyla uygulamıştır. Fakat, bu yaklaşım şablon eşleştirme gerektirdiğinden, her bir yeni resimdeki her bir pencere için yoğun ve maliyetli araştırmalar gerektirmektedir.

Görsel tanımlamanın tarihi, öznitelik ölçümlerinde kullanılacak, farklı resim öznitelik dizileri içermektedir. Carneiro ve Jepson [6], çalışmalarında, yerel mekan frekansları genliğini kullanmak yerine, aydınlanmadan bağımsızlığı daha başarılı

şekilde sağlayan, faz temelli yerel öznitelikleri tanımlamışlardır. Schiele and Crowley [46], bir resmin bölgeleri içindeki ölçüm dağılımlarını özetleyen, çok yönlü histogramlar oluşturmuşlardır. Bu öznitelik tipleri özellikle şekli bozulabilen desenli nesnelerin tanınmasında yararlıdır. Basri and Jacobs [3], tanımlama için yerel bölge sınırlarının ayrılmasının önemini göstermiştir.

Bunlara ilave olarak; renk, hareketlilik, nesnenin arka plandan çıkarılması, bölge şekli tanımlayıcıları gibi özellikler de öznitelik tanımlamalarında kullanılabilir. Yerel öznitelik yaklaşımı kolaylıkla alışılmadık öznitelik türleriyle birleştirilebilir. Çünkü, sayıca fazla öznitelikler bir miktar hesaplama maliyeti getirmekle birlikte, doğru eşleştirme şansını artıracaktır.

Köşe noktaları üzerinden bölgesel nesne tanımlayıcıları oluşturan nesne tanıma yöntemlerinin karşılaştığı ortak sorun; tek bir ölçekte çalıştıklarında başarılı olmaları, farklı ölçeklerde çalışıldığında ise, her ölçekte belirlenen noktaların farklı konumlarda olması sebebiyle ölçekten bağımsızlığın sağlanamaması olmuştur.

Lowe [25] 'un geliştirdiği, ölçekten bağımsız öznitelik dönüşümü, SIFT (the Scale Invariant Feature Transform) [25]; [26] yöntemi ile, köşe noktaları kullanılarak, nesnenin bölgesel öznitelikleri ölçekten bağımsız olarak çıkarılmıştır.

Mikolajczyk and Schmid [32], birçok yaklaşımı değerlendirmiş ve SIFT algoritmasını yaygın resim bozuculara karşı en dayanıklı algoritma olarak belirlemişlerdir.

Ledwich and Williams [21] ev ve ofis gibi iç mekanların avantajlarını kullanarak, SIFT özniteliklerinin karmaşıklığını ve ortamı tanımlayan öznitelik sayısını azaltmışlar böylece öznitelikleri sadeleştirmişlerdir. Özniteliklerin boyutlarının ve karmaşıklığının azaltılması ile eşleştirme süresi kısaltılmıştır. Beraberinde çok küçük miktarlarda kesinlik kayıpları olmuştur. Çalışmalarını, resim yeniden kazanımı ve iç mekanlarda robotlarda konum saptanması üzerinde uygulamışlardır. Konumlandırma filtresinin kesinliğinin de SIFT özniteliklerinin ölçek bilgisinin kullanılması ile artırılabilceğini göstermişlerdir.

Ke and Sukthankar [19] çalışmalarında, SIFT içerisinde kullanılan farklı bölgesel resim tanımlayıcıları geliştirmişlerdir. Öznitelik noktasının komşuluğu içinde resim eğimlerinden kararlı yönelimler çıkaran tanımlayıcılar kullanmışlardır. Farklı olarak, eğitim bölgelerinin normalize edilmesi için yumuşatılmış ağırlık histogramları (smoothed weighted histograms) yerine PCA (Principal Components Analysis) kullanmışlardır. PCA tabanlı bölgesel tanımlayıcıların daha ayırt edici ve resim bozucularına karşı daha kararlı olduğunu, standart SIFT temsiline göre daha tanımlayıcı olduğunu, resim yeniden kazanımlarında ise, bu uygulama ile duyarlılığın arttığını, eşleşme süresinin kısaldığını ileri sürmüşlerdir.

Goedem'e, Tuytelaars and Gool [11], bağımsız çalışan robotlarda görüntü sensörlerinin zengin çevre bilgileri vermeleri sebebiyle eldeki bilgi çokluğunun robot tarafından değerlendirilmesinin zorluğu üzerinde bir çalışma yapmışlardır. Bakış açısı büyük ölçüde değiştiğinde de tanımlama yapılabilmesi ve bölgesel öznitelikler ile resimlerin birbirinden ayrılması sağlanmış, sonuç olarak geniş temelli bölgesel öznitelikler ile çok yönlü bir yol izleme, yöngüdümlü metodu geliştirmişlerdir.

Çok yönlü biyometrik uygulamalarda şimdiye kadar birçok örüntü tanıma ve sınıflandırma teknikleri kullanılmıştır. Bicego et.al [4], SIFT özniteliklerini biyometrik uygulamalarda yüz kimliklendirme amacıyla, çok sayıda resim üzerinde incelemişlerdir ve bu yaklaşımın gelecek vaad ettiğini gözlemlemişlerdir.

Moeslund , Hilton and Krüger [34], otomatik insan hareket tespiti ve takibi, eylem ve davranış anlama, duruş ve hareket tanıma amacıyla SIFT özniteliklerini kullanmışlardır.

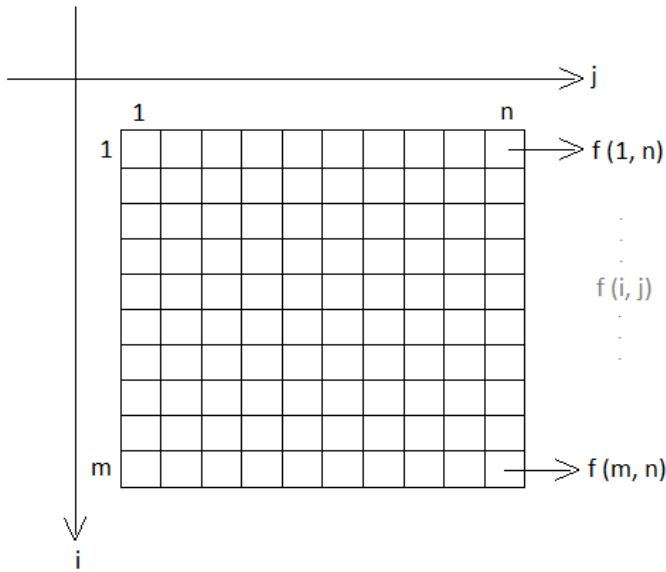
Zhou et.al [57], SIFT tabanlı bir ortalama kayma algortimasını, doğal ortamlarda, nesne takibi amacıyla kullanmışlardır. Ortalama kayma, renk histogramları aracılığı ile benzerlik aramasını yönlendirmek için uygulanmış, benzer bölgelerin üstdeğer ihtimal tahminini yapabilmek için beklenti enbüyüklenme (expectation–maximization) şeması içerisinde, bir olasılık dağılımı geliştirilmiştir. Bu destek mekanizması, iki ölçümden biri kararsız olduğunda, istikrarlı izleme performansı sağlamıştır.

Cheung and Hamarneh [8], çok biçimli (multimodal) medikal resim eşleştirme tekniği tabanlı, n-boyutlu bir SIFT algoritması (n-SIFT) geliştirmişlerdir. Üç boyutlu MRI (magnetic resonance imaging) resimleri ve dinamik dört boyutlu (üç boyut ve zaman) CT (computed tomography) datalarında, etkili eşleştirmeler yaparak, alakalı resimlerde birbirini karşılayan işaret noktalarının (landmark) bulunmasında açık kaynak kodlu bir yazılım sunmuşlardır.

Bu çalışmada, yapılmış çalışmalardan farklı olarak belirlenen öznelikler sonucu yapılan eşleşmelerde, algoritmanın doğru olarak gösterdiği yanlışlar (false-positive) için bir kontrol aşaması eklenmiş ve gerçekte doğru olmayan eşleşmeler elenerek çalışmanın kesinliği artırılmıştır.

2.3 Resimlerin Bilgisayar Ortamında Değerlendirilmesi

Gri seviye sayısal resimler Şekil 2.3 'deki gibi, ilgili piksel koordinatına karşılık gelen, bir $f(i, j)$ gri seviye genlik değeri ile gösterilir.



Şekil 2.3 $m \times n$ pikselden oluşan bir sayısal resimde, (i, j) noktasındaki gri seviye genlik değeri $(f(i, j))$

MATLAB®¹ ortamında ise, resimler bir matris olarak algılanır. Şekil 2.4'de görüleceği gibi, resmin pikselleri matrisin elemanlarını oluşturur. Pikseller üzerinde yapılan tüm işlemlerde matris hesaplama yöntemleri kullanılır. Resmin belli bir parçasından bahsedildiğinde bu bir altmatristir. Resmin filtrelenmesi gibi durumlarda ise, matris çarpımları kullanılır.

$$I = \begin{pmatrix} f_{11} & \mathbf{K} & f_{1n} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ f_{m1} & \mathbf{L} & f_{mn} \end{pmatrix}$$

Şekil 2.4 mxn pikselden oluşan bir sayısal resmin, Matlab ortamındaki ifadesi

2.4 Nesneyi Tanımlayan Öznitelikler

Graves and Batchelor, yaptıkları çalışmalarında [14, s.62], nesnenin şekline dayalı öznitelik çıkarımları için, şu görüşleri ileri sürmektedir:

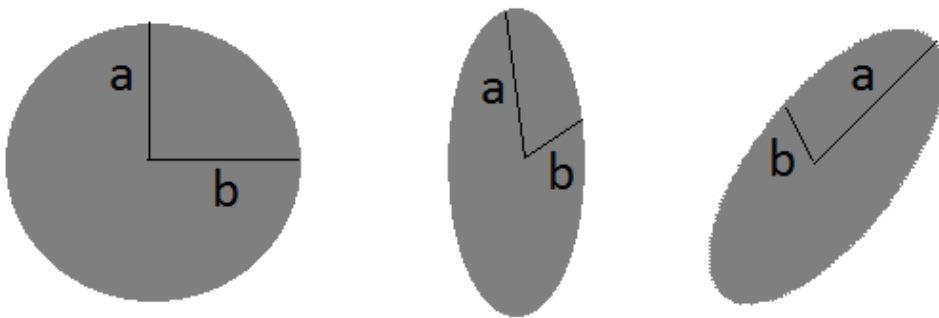
“Nesnenin iki boyutlu düzlem üzerinde kapladığı bölge düşünülerek, aşağıdaki gibi bazı şekil tanımlayıcıları kullanılabilir.

- Bölgenin kenarı üzerindeki en uzak noktanın, bölgenin merkezine olan uzaklığı,
- Bölgenin kenarı üzerindeki en yakın noktanın, bölgenin merkezine olan uzaklığı,
- Yarıçapı, ilk iki maddede hesaplanan uzaklıkların ortalamasına eşit olan çemberin dışına çıkan yayların sayısı,
- Bölgenin kenarında bulunan noktaların merkeze uzaklığının açısız pozisyon cinsinden ifadesi. Böylece, silüet polar koordinat terimleri ile ifade edilir. (Bu tek değerli bir fonksiyon değildir.)

¹ MATLAB yüksek seviyeli bir programlama ortamıdır.
<http://www.mathworks.com/products/matlab/>

- Dairesellik = Alan / Çevre². Bu değer dış hatları parçalı olan düzensiz şekiller için sıfıra yaklaşırken, bir daire için maksimum değer olan $\pi/4$ 'e yaklaşır.
- Nesne üzerinde bulunan hücrelerin sayısı
- Nesne kenarlarında, dıştan içeriye giren koyların sayısı
- Şekil üzerindeki, devam eden eğri sayısı ile, bu eğriler arasında kalan boşluk sayısının birbirinden çıkarılması ile elde edilen, Euler numarası
- Gerçek resim alanı ile, resme ait dışbükey zarf (convex hull) alanının oranı
- Gerçek resim alanı ile, resme ait çevrel daire (circumcircle) alanının oranı
- Resim alanının, resme ait iskeletteki toplam dal uzunluklarının karesine oranı
- İskeletteki eklem yerleri ile dalların bitişi arasındaki mesafeler
- Majör ve minör eksenler üzerine izdüşümler oranı“

Bu öznitelikler gerçek resimler üzerinde çalışırken, daha önce bahsedilen ve Şekil 2.2'de gösterilen değişken koşullar sebebiyle ayırt edici öznitelikler olarak kullanılamamaktadır. Örneğin Şekil 2.5'de görüldüğü gibi $a=b=r$ olan bir daire, sadece farklı yönlerden bakıldığında elips şeklinde görünecektir. Bu daire için, yukarıdaki gibi hesaplamalar, örneğin orta noktadan kenara olan uzaklıkların oranı gibi öznitelikler, artık ayırt edici olmayacaktır.



Şekil 2.5 Bir dairenin bakış yönü değiştirildiğinde elips şeklinde görünmesi

2.5 Oryantasyondan Bağımsız Öznitelik Çıkarımı

Köşe noktaları üzerinden bölgesel nesne tanımlayıcıları oluşturan nesne tanıma yöntemlerinin hepsinin karşılaştığı sorun; tek bir ölçekte çalıştıklarında başarılı olmaları, farklı ölçeklerde çalışıldığında ise, her ölçekte belirlenen noktaların farklı konumlarda olması sebebiyle ölçekten bağımsızlık sağlanamamasıydı.

Ayrıca nesnenin ölçeğinin bilinmesi, resim tanımlayıcıları oluşturmak ve bir çok ölçek arasında eşleme yapabilmek için gereklidir. Bu sebeplerden, köşe sezinleyicisinin belirlediği noktaların hangi ölçekte belirlendiği bilgisini de vermesi gerekir.

SIFT [25] algoritması nesnelerin bölgesel özniteliklerinin ölçekten bağımsız olarak çıkarılmasında kullanılan bir yöntemdir. Çıkarılan bu öznitelikler nesnenin ölçeği ve yöneliminden bağımsız olduğundan gerçek resimlerde, farklı açılardan görünen nesne veya ortamları eşleştirme amacıyla kullanılabilir [26].

SIFT algoritması, bir resmi girdi olarak alır ve bir resim öznitelik kümesi geri döner. Tespit edilen her nokta bir konum, bir ölçek, bir yönelim ve bir tanımlama vektörüne sahiptir. Bu özniteliklerden oluşturulacak vektörler, eşleştirme aşamasında kullanılacaktır.

Öznitelik çıkarma süreci, uygulanması kolay olan aşamalarla başlar, elemelerden geçen bölgeler üzerinde daha karmaşık hesaplamalara geçilir. Böylece işlem maliyeti azaltılmış olur.

Algoritma dört ana adımdan oluşur [26].

- Ölçek uzayı uçdeğer kestirimi
- Anahtar nokta konumlandırma
- Anahtar noktalara bağımsız yönelimler atanması
- Bölgesel resim tanımlayıcı vektörler oluşturma

2.5.1 Ölçek uzayında uçdeğer kestirimi

Resmin belirli oranlarda tekrar tekrar yeniden ölçeklenmesi ve filtrelenmesi ile oluşturulan ölçek uzayı, aranan nesneyi farklı koşullar altında da tanıyabilmeyi sağlayacak özneliklerin çıkarımına imkan vermektedir. Ölçek uzayını oluşturan bu farklı ölçeklenmiş ve filtrelenmiş resimler, nesnenin içinde olduğu karşılaşılabilecek muhtemel arama ortamlarında, ne gibi özneliklere sahip olacağı bilgisini verir.

Aranan nesnenin farklı ölçeklerde nasıl noktalar vereceği önceden belirlendiğinde, karşılaşılabilecek arama ortamlarındaki ölçek farklılıkları sorun olmayacaktır. Ölçek uzayı oluşturulması ve anahtar noktaların sabit konumlarının belirlenmesi ile, resmin farklı ölçeklerinde her defasında tespit edilecek noktaların konumlarının değişmemesi sağlanır. Böylece ölçekten bağımsız sabit anahtar noktalar belirlenebilecektir.

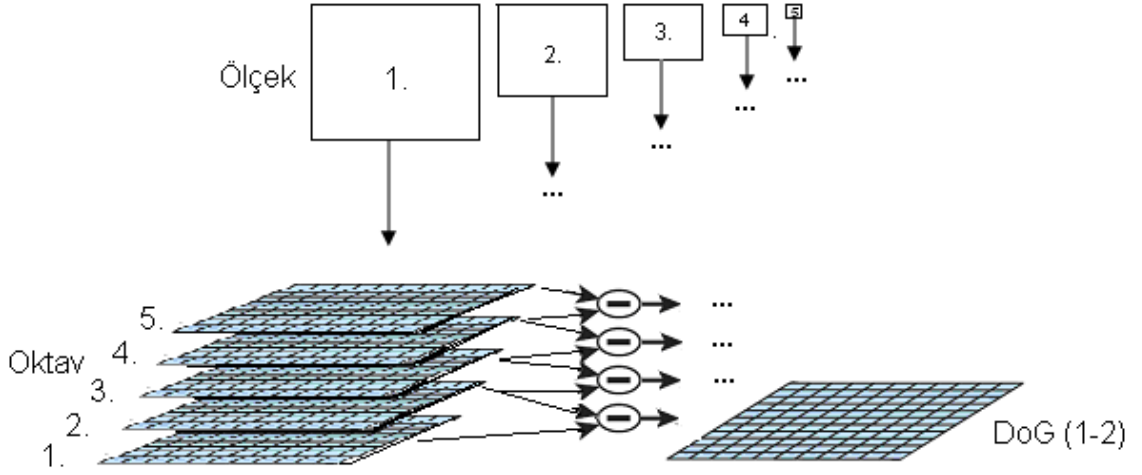
SIFT algoritması, Gauss farkları piramidi yani DoG (difference-of-Gaussian) piramidi oluşturulması ve bu piramitteki bölgesel uçdeğerlerin konumlarının belirlenmesi ile başlar. DoG resimlerde uçdeğerlerin konumlarının belirlenmesi ile anahtar nokta adayları belirlenmiş olur. Anahtar nokta olduğu kesinleşen noktalar daha sonra bir öznelik vektörü oluşturmada kullanılacaktır.

2.5.1.1 Oktav ve ölçek kavramları

Oktav (octave); bir resmin Gauss ile filtrelenmesi sonucu elde edilen yeni resmin bulanıklık derecesidir. Eşitlik 2.1'deki Gauss ifadesinin σ değeri değiştirilerek bulanıklık miktarı değiştirilir ve farklı oktavlar elde edilir.

Ölçek; resmin yeniden boyutlandırılması için kullanılan, resmin boyutunu büyüten veya küçülten katsayıdır.

Oktav ve ölçek sayısı, üzerinde işlem yapılacak olan orijinal resmin ebatına bağlıdır. Ne kadar oktav ve ölçek olacağına, uygulama sonuçlarına bakılarak karar verilmelidir. Fakat, Lowe [25] dört oktav ve her bir oktav içinde beş bulanıklaştırmayı algoritma için ideal olarak tavsiye etmiştir (Şekil 2.6).



Şekil 2.6 Ölçek, oktav kavramlarının gösterilmesi ve Gauss farkının alınması

Bulanıklaştırma :

Resmi Gauss operatörü ile katlamak, yani Gauss operatörünü her piksele uygulamak bulanıklaşmış resmi oluşturur. Bulanıklaştırma ile alçak geçiren filtreleme kastedilmektedir. Bulanıklaştırılmış resim orjinalinden çıkarıldığında, ani parlaklık değişimlerinin olduğu noktaları belirginleştiren bir yüksek geçiren filtre gibi sonuç verir [14].

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.1)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.2)$$

x ve y : resmin koordinatları

$L(x, y, \sigma)$: resmin ölçek uzayı fonksiyonu

$G(x, y, \sigma)$: değişken ölçekli Gauss fonksiyonu

$I(x, y)$: algoritmaya giren resim

σ : ölçek parametresi, bulanıklaştırma miktarıdır. σ değeri arttıkça bulanıklık artar.

2.5.1.2 Gauss farkları piramidi oluşturulması

Koenderink (1984) ve Lindeberg (1994) bazı kabul edilebilir yaklaşımlar altında tek olası ölçek uzayı çekirdek (kernel) matrisinin Gauss fonksiyonu olduğunu göstermişlerdir. Böylece bir resmin ölçek uzayı, resmin Gauss ile katlanmasından oluşur.

$$D(x, y, \sigma) = DoG * I(x, y) \quad (2.3)$$

$D(x, y, \sigma)$: Resim ile DoG katlaması sonucu oluşan fonksiyon

DoG işlemi iki ardışık ölçeğin birbirinden çıkarılması ile gerçekleştirilir (Şekil 2.6).

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \quad (2.4)$$

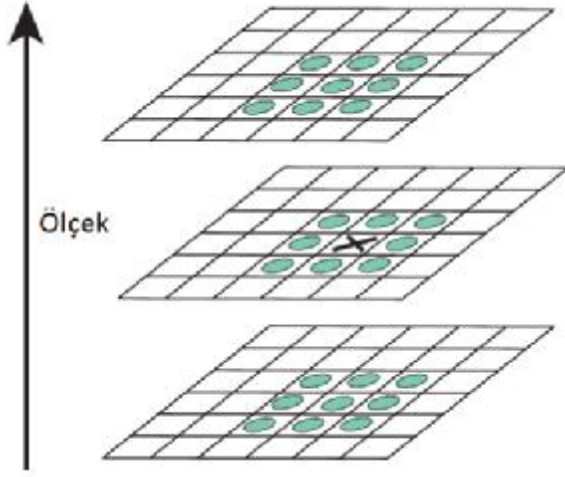
$$D(x, y, \sigma) = L(x, y, k\sigma) * L(x, y, \sigma) \quad (2.5)$$

DoG, LoG (Laplacian of Gaussian) yaklaşımına oldukça yakındır (Bkz. Ek 2). Neredeyse tamamen aynı sonuçlar üretmelerine karşılık, LoG'ın ölçekten bağımsız olmaması, gürültüye karşı çok hassas olması ve DoG yöntemine göre işlem karışıklığı olması sebebiyle, DoG kullanımı tercih edilmiştir.

Hem LoG, hem de DoG kenar bulma algoritmalarında ve diğer görüntü işleme uygulamalarında ön işlem olarak, kenar iyileştirme amacıyla kullanılmaktadır.

2.5.1.3 Bölgesel uçdeğer tespiti

Anahtar noktaların belirlenmesi için önce LoG veya DoG resimlerde uçdeğerlerin konumu belirlenir. Uçdeğerler önce kabaca konumlandırılır, bunun için de her piksel tüm komşuları ile tek tek karşılaştırılır. Komşuluk pikselin bulunduğu ortam ile komşu ortamlar arasında da aranır.

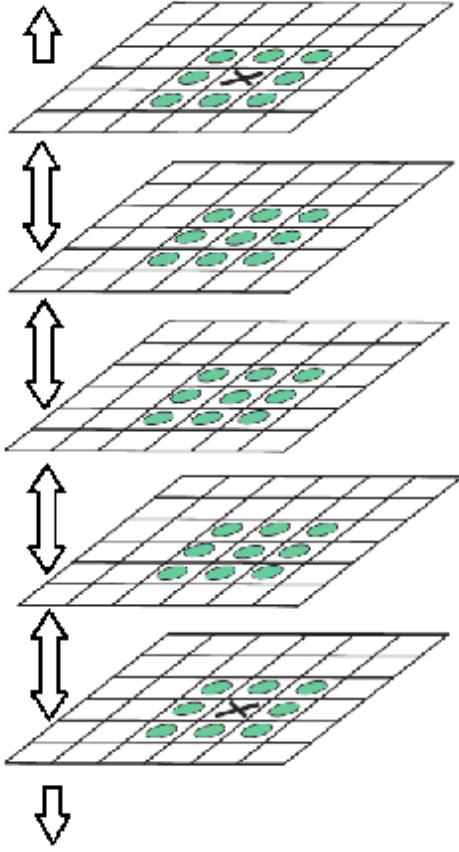


Şekil 2.7 Uçdeğerlerin aranmasında kullanılan komşu pikseller (Lowe, [2]'den değiştirilerek)

Şekil 2.7'de X ile gösterilen piksel için dairesel işaretlenmiş diğer pikseller komşulardır. Üst ve alt ortamlarda 9ar komşu, aynı ortamda 8 komşu olmak üzere toplam 26 komşu vardır.

X bu 26 piksel ile kıyaslanır ve en büyük veya en küçük değere sahip ise; X bir uçdeğer olarak atanır. İlk birkaç kıyaslamada en küçük veya en büyük değer olmadığı anlaşılırsa, bu piksel için kıyaslamaya devam etmek gereksizdir.

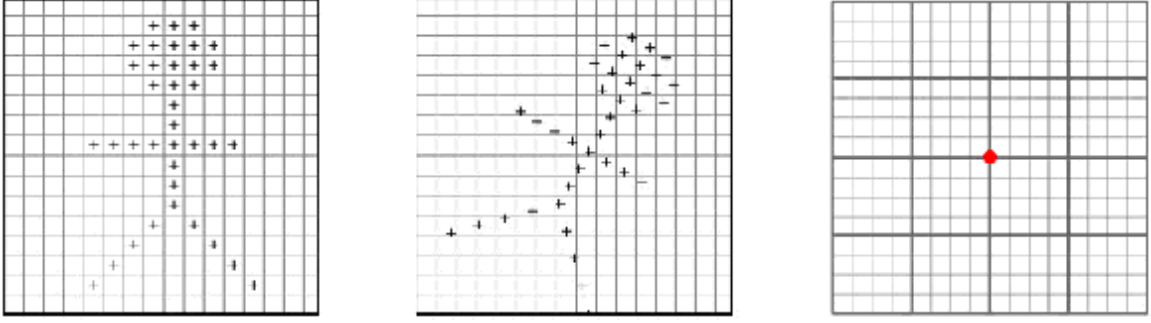
Uç değerler en üstteki ve en alttaki ortamlarda aranmazlar, çünkü karşılaştırma için kullanılacak piksellerden, üstten veya alttan 9 adet piksel eksiktir (Şekil 2.8).



Şekil 2.8 Uçdeğerlerin en alt ve en üstteki ölçeklerde aranmaması

2.5.2 Anahtar nokta konumlandırma

Uçdeğerler neredeyse her zaman, tam olarak bir piksel üzerinde bulunmazlar, genelde piksellerin arasında bulunurlar (Şekil 2.9). Bu sebeple bulunan uçdeğerler yaklaşık değerlerdir. Aday noktalar belirlendikten sonra ölçek uzayında gerçek konumları için ara değer kestirimi (interpolation) yapılır. (Bkz. Ek 3). Ara değer kestirimi eğim genlik ve yönelim hesaplaması için gereklidir.



Şekil 2.9 Resim üzerinde yapılan dönüşüm sonucu ara değerlerin oluşması²

Lowe [25], çalışmasında bu noktaları çalıştığı ölçekte ve örnek noktanın merkezine yerleştirmiştir. Daha sonra, Brown and Lowe [5], uçdeğerin kestirilmiş konumunu belirlemek amacıyla yerel örnek noktasına bir üç boyutlu, ikinci dereceden fonksiyon uydurma yöntemi geliştirmişlerdir. Bu yöntemin noktaların belirlenmesinde kullanılması ile eşleştirme ve kararlılıkta büyük oranda başarı artışı görülmüştür. Yöntem, $D(x,y,\sigma)$ ölçek uzayı fonksiyonunun Taylor açılımını kullanarak ara değer kestirimi yapmaktadır. Yerel örnek noktası $x = (x, y, \sigma)^T$, kullanılarak Taylor denklemi yazılırsa;

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.6)$$

D : ölçek uzayı fonksiyonu

x : yerel örnek noktası

Bu eşitliğin x'e göre türevi alınır ve sıfıra eşitlenirse, aranan uçdeğerler hesaplanabilir ve ara piksellerin gerçek konumları bulunur. Ara piksellerin kullanılması algoritmanın kararlılığını ve kesinliğini artırır.

² http://idlastro.gsfc.nasa.gov/idl_html_help/Interpolation_Methods.html, Ağustos 2011.

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2.7)$$

\hat{x} : uçdeğerin konumu

Hessian matrisi ve D'nin türevi, birbirine komşu örnek noktaların farklarını kullanarak yaklaştırılır. En büyük özdeğerin, en küçük özdeğere oranı büyüdükçe, yani komşu örnek noktalar arasındaki mesafe arttıkça, bu köşelerden uzaklaşıldığı, bir düzlemde veya kenar üzerinde gezinildiği anlamına gelir. Anahtar nokta belirleme aşamasında, daha çok tanımlayıcı bilgi içereceğinden, bu oranın küçük olması istenir.

2.5.2.1 Kararsız uçdeğerlerin elenmesi

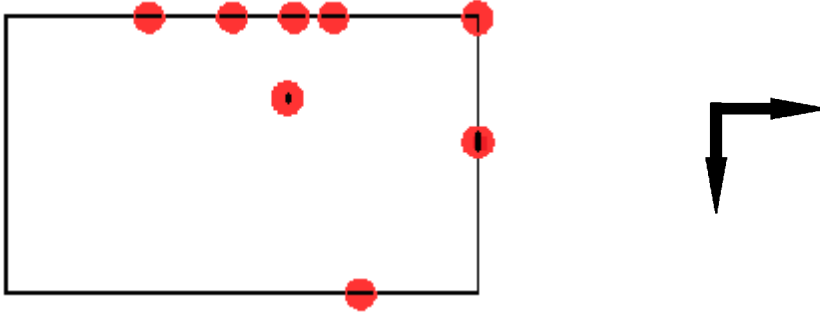
Düşük kontrastlı veya kenar üzerinde bulunan noktalar muhtemel kararsızlık nedeni ile elenir. Eşitlik 2.8'den elde edilecek $|D(\hat{x})|$ değeri düşük kontrastlı, dolayısı ile gürültüye karşı dayanıksız uçdeğerlerin belirlenmesinde kullanılır. Bu çalışmada, resim piksel değerleri [0, 1] aralığına ölçeklenmiştir ve $|D(\hat{x})|$ hesaplamasında, 0.03'den küçük çıkan değerler reddedilmiştir. Bu eşik kriteri ile yapılan test ölçümleri, 0.03 değerinin belirlenen nokta sayısında ortalama %12.5 azalma yarattığını göstermiştir.

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (2.8)$$

$D(\hat{x})$: uçdeğerin fonksiyon değeri

DoG resimler kenarlara güçlü cevaplar verirler. Fakat aynı kenar üzerinde belirlenen her nokta birbirine benzemektedir. Bu da noktaların ayırt ediciliğini azaltır.

Uçdeğerler üzerinde iki eğim hesabı yapılır. Şekil 2.10'da görüldüğü gibi, noktanın bulunduğu yere bağlı olarak üç durum mümkündür.



Şekil 2.10 Uçdeğerlerin bulunabileceği konumlar

- Düz bir alan ise; iki eğim de küçük olacaktır.
- Kenar ise; bir eğim büyük (kenara dik), diğeri eğim küçük (kenar boyunca) olacaktır.
- Köşe ise; iki eğim de büyük olacaktır.

Anahtar nokta olarak köşeler tercih edilir. Hessian Matrisi ile noktanın bir köşe olup olmadığı da kolayca görülebilir. Eşitlik 2.9'da 2x2 boyulu bir Hessian Matrisi görülmektedir. DoG fonksiyonunda zor ayırt edilebilen bir zirve, kenar üzerinde yüksek temel eğrilik (principal curvatures) geçişine sahiptir fakat, dikey yönde bu değer küçüktür. Temel eğriler, anahtar noktanın ölçek ve konumunda, 2x2'lik Hessian Matrisiden hesaplanabilir. Türevler bitişik örnek noktalarına ait genlik farkının alınması ile hesaplanır.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.9)$$

Hessian matrisi simetrik matris olduğundan, matrisin izi ve determinantı sırasıyla Eşitlik 2.10 ve Eşitlik 2.11' den hesaplanabilir.

$$Tr(H) = D_{xx} + D_{yy} = a + b \quad (2.10)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = ab \quad (2.11)$$

α : genliđi büyük olan özdeđer (eigenvalue)

β : genliđi küçük olan özdeđer

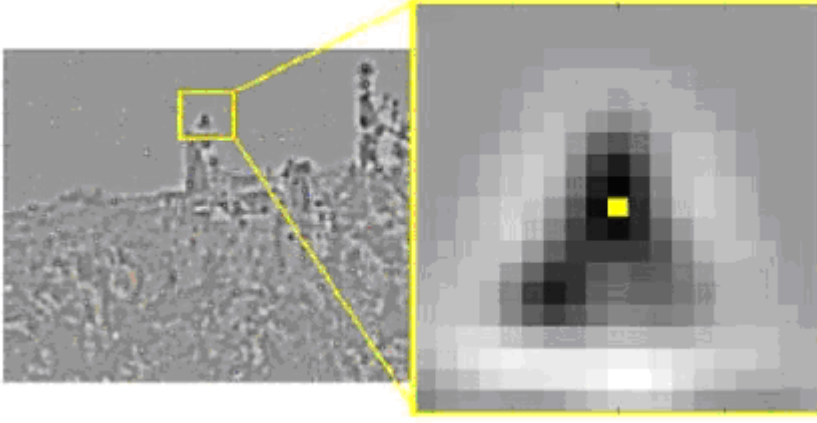
Özdeđerlerin toplamını H'in izinden (trace), çarpımını ise belirteninden (determinant) hesaplayabiliriz.

$$\frac{Tr(H)^2}{Det(H)} = \frac{(a+b)^2}{ab} = \frac{(rb+b)^2}{rb^2} = \frac{(r+1)^2}{r} \quad (2.12)$$

Artık diđer işlemlerde, sadece elemeden geçen kararlı anahtar noktalar kullanılacaktır. Bu anahtar noktaların tespit edildiđi ölçek bilindiđinden, ölçekten bağımsızlık da sağlanmış olur.

2.5.3 Anahtar noktalara yönelim ataması

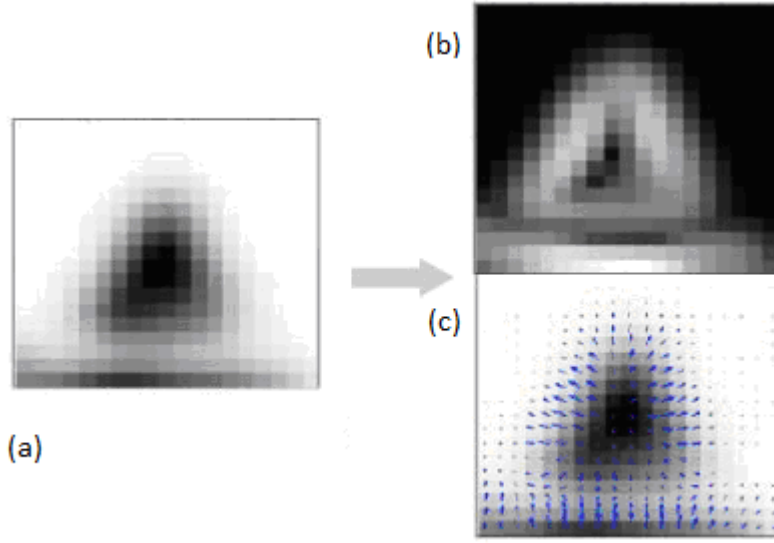
Döndürmeden bağımsızlığı sağlayabilmek için anahtar noktalara yönelim ataması yapmak gereklidir. Filtrelemeden geçen anahtar noktalara, bölgesel ölçek uzayı eğimlerinin baskınlığına göre bir veya daha fazla sayıda standart yönelimler atanabilir. Bunun için de belirlenen anahtar noktalar etrafındaki (Şekil 2.11) eğimlerin yönü ve genlikleri kullanılarak (Şekil 2.12), en belirgin yönelim seçilir. Bu yönelim de oradaki anahtar noktaya ait yönelim olarak atanır. Sonraki tüm hesaplamalar ise bu yönetime göre yapılacaktır.



Şekil 2.11 Anahtar nokta etrafında eğimlerin inceleneceği bölge [58]

Yönelim hesaplamak için kullanılan alanın ebatı, ölçeğe bağlıdır. Ölçeğin büyük olması ile bu alanın büyümesi sağlanır.

Yönelim atandıktan sonra her bir anahtar nokta tanımlayıcısı için, göreceli konum, ölçek ve yönelim hesaplaması yapılabilir. Oryantasyondan bağımsızlık bu şekilde sağlanmış olur. Anahtar noktalar, ölçek, yönelim ve konumlarını ifade eden vektörler ile gösterilirler.



Şekil 2.12 Araştırma bölgesinde eğim genlik ve yönelimi ([58]'den değiştirilerek)

- (a) Bulanıklaştırılmış resim parçası
- (b) Eğim genliği
- (c) Eğim yönelimleri

Eğimin genliği ve yönelimi şu formüller ile hesaplanır:

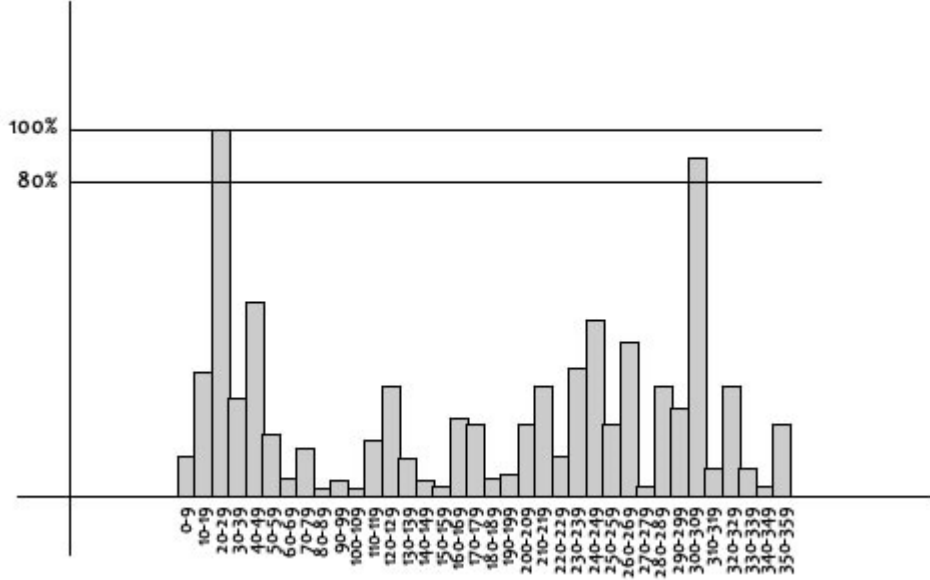
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.13)$$

$$q(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / ((L(x+1, y) - L(x-1, y)))) \quad (2.14)$$

L : Gauss ile bulanıklaştırılmış resim.

Kullanılan bulanıklaştırılmış resim, anahtar noktanın belirlendiği ölçekteki bulanık resimdir. Bu ölçekteki tüm $L(x, y)$ resimleri için, eğim genlik değeri $m(x,y)$ ve yönelimi $\theta(x,y)$ hesaplanır.

Anahtar nokta etrafındaki tüm piksellerin eğim yönelimleri kullanılarak, bir oryantasyon histogramı oluşturulur. Histogram Şekil 2.13’de görüldüğü gibi, 360 derecenin bölündüğü, 36 basamaktan oluşur. Her nokta, eğim genliğine göre ağırlıklandırılarak bu histograma eklenir.



Şekil 2.13 Anahtar nokta civarında kuantalama [58]

Örneğin araştıma bölgesindeki bir noktanın eğim yönelimi 72.564 derece ise, bu nokta 70-79 derece dilimine, yani 8. basamağa girecektir. Noktanın eğiminin genliği de orantılı olarak dilime eklenir. Bu işlemler tüm anahtar noktalar etrafındaki pikseller için yapılır.

Yukarıdaki histogramda tepe yapmış olan genlik 20-29 dereceleri arasında olduğundan, bu anahtar nokta için atanan yönelim 3. dilimdir. Ayrıca en yüksek genliğin en az %80 değerine sahip olan başka dilimler de varsa, aynı yerde bulunan, aynı ölçek değerine sahip, yönelimi farklı olan başka anahtar noktalar da seçilmelidir. Yukarıdaki histogramda, 300-309 dereceleri arasında benzer bir durum vardır. Dolayısıyla burda aynı yerde ve ölçekte bulunan, fakat yönelimi 31.dilim olan başka bir anahtar nokta daha seçilmelidir [26].

Özetle, yönelim atamak için histogram ve yönelim hesaplama alanından faydalanılır. Histogram ile, en baskın yönelim veya yönelimler seçilir, gerekirse

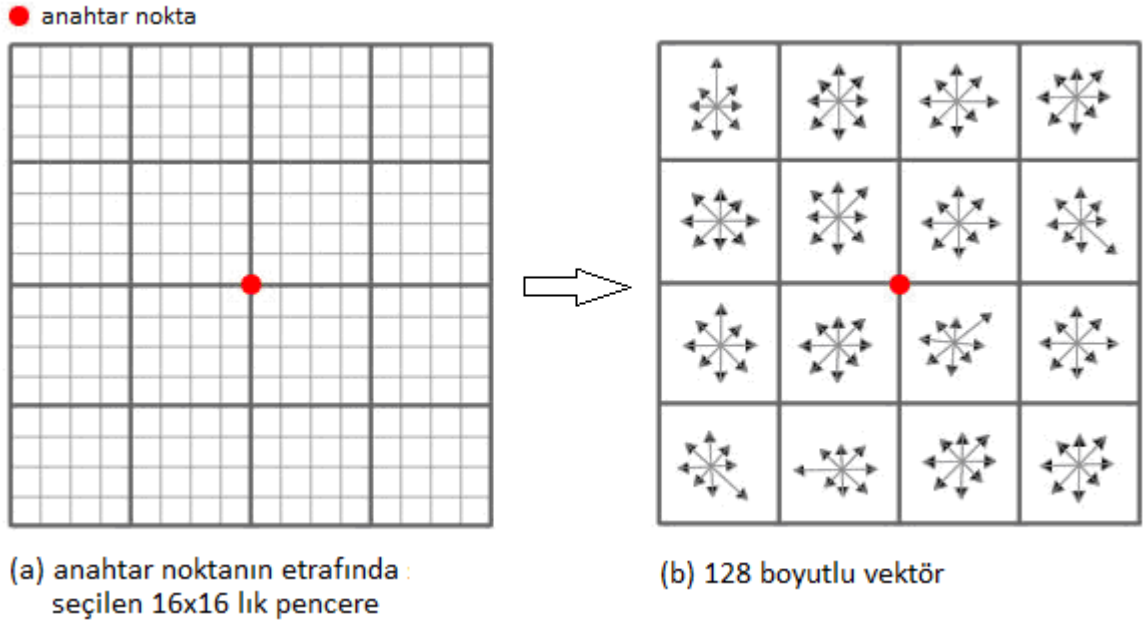
başka anahtar noktalar da belirlenir. Fakat, bu anahtar noktalar için daha ayırıcı özelliklere ihtiyaç vardır. Aydınlanma farklılıkları veya üç boyutlu bakış açısı değişiklikleri durumunda tanımayı sağlayabilmek için, bölgesel ayırt edici tanımlayıcılar gereklidir. Burada da “öznitelik” kavramı devreye girer.

Resimden çıkarılan öznitelikler, en yakın komşu yaklaşımına göre nesne modelini betimlemek amacıyla sıralanır ve “öznitelik vektörü” oluşturulur.

2.5.4 Bölgesel resim tanımlayıcılar oluşturulması

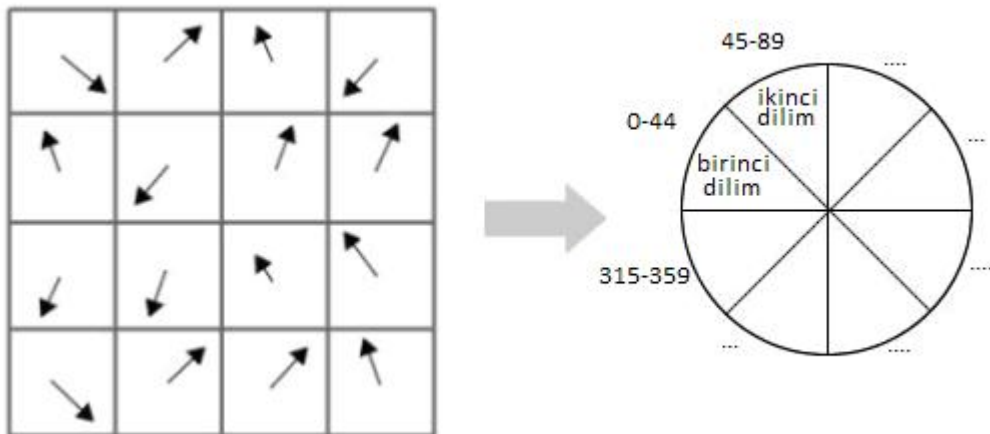
Her bir anahtar noktayı, diğer anahtar noktalardan ayırt edici olarak tanımlamak gerekmektedir. Anahtar noktaları tanımlamak amacıyla, noktanın bulunduğu bölgenin özelliklerinden faydalanılır.

Anahtar nokta etrafında 16x16’lık bir bölge alınır (Şekil 2.14 (a)). 16x16’lık alan, 4x4’lük çerçevelerden oluşan 16 parçaya ayrılır. Her bir parça içindeki bölgesel eğim yönelimlerinin histogramı hesaplanır ve bu histogramlar bir vektör içinde sıralı olarak birleştirilir. Aydınlanma farklılıklarından etkilenilmemesi için ise, her tanımlama vektörü birim uzunluğa normalize edilir. Büyük eğim değerlerinin etkilerini azaltmak için, eşik değerinden geçirilip, normalize edilme işlemi yinelenir.

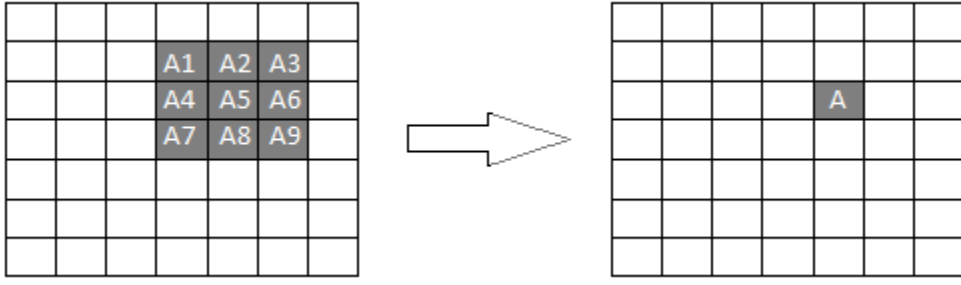


Şekil 2.14 Anahtar nokta etrafında 128 boyutlu vektör oluşturulması ([58]'den değiştirilerek)

Bu 4x4 boyutlu her bir çerçevede eğim genlik ve yönelimleri hesaplanır. Bu yönelimler 8 dilimlik bir histograma, genlikleri de dikkate alınarak yerleştirilir (Şekil 2.15)

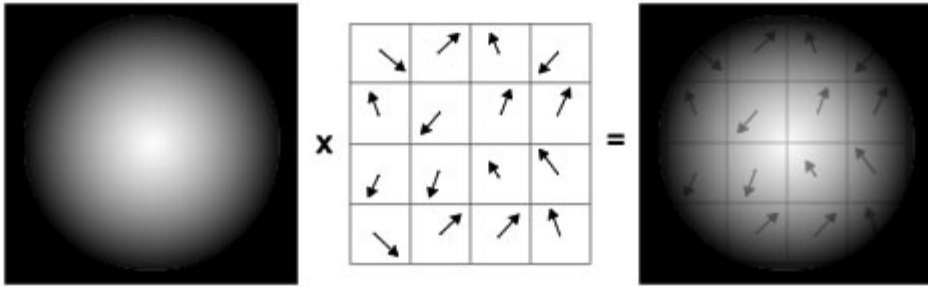


Şekil 2.15 4x4'lük alt pencereler içinde eğim genlik ve yönelimleri histogramının oluşturulması



Şekil 2.16 Ağırlık matrisinin resim üzerine uygulanışı

Ayrıca, anahtar noktaya olan uzaklık da önemlidir. Bu nedenle Şekil 2.16'da görülen, standart ağırlık matrisi kullanımından farklı bir uygulama gerekir. Nokta, anahtar noktadan uzaklaştıkça histograma eklenen değeri de küçülür. Bu işlem "Gauss ağırlık fonksiyonu" ile yapılır (Şekil 2.17).



Şekil 2.17 Gauss ağırlık fonksiyonunun eğim genlik ve yönelimlerine uygulanışı

[58]

Bu fonksiyon iki boyutlu bir çan eğrisi gibidir. Yönelimlerin genlikleri ile çarpıldığında, uzaklığa göre ağırlıklandırılmış yönelimler elde edilir.

Tüm 4x4 'lük bölgeler için, yani 16 kez aynı işlem tekrarlanır. Her bir bölgeden histogramın 8 dilimine ait, 8 sayı (genlik değerleri toplamı) elde edilir ve 16 bölge için toplam (16x8) 128 adet sayı elde edilmiş olur (Şekil 2.14 (b)).

Dört elemanlı bir anahtar nokta tanımlayıcısı Eşitlik 2.15 'de gösterilmiştir. Her bir anahtar civarında belirlenen 128 boyutlu anahtar nokta tanımlayıcı öznelik vektörü ise Eşitlik 2.16 'da gösterilmiştir.

$$X_n=[X_{\text{satır}}, X_{\text{sütün}}, X_{\text{genlik}}, X_{\text{yönelim}}] \quad (2.15)$$

$$F_n=[f_{n,1}, f_{n,2}, \dots, f_{n,128}] \quad (2.16)$$

n : bulunan anahtar nokta sayısı

Bu 128 sayılı öznelik vektörü, “kareler toplamının kareköküne bölme” yöntemi ile normalize edilir. Her bir normalize edilmiş öznelik vektörü, bir anahtar noktayı ayırt edici olarak tanımlar.

Bu aşamalarda eğim yönelimleri kullanıldığı için, resimler arası eşleştirmede açısal değişiklikler sorun yaratacaktır. Döndürmeden bağımsızlığı yeniden yakalamak için, anahtar noktanın açısı, her bir yönelimden çıkarılmalıdır. Böylece her bir eğim yönelimi, anahtar noktanın yönelimine göre, göreceli olacaktır.

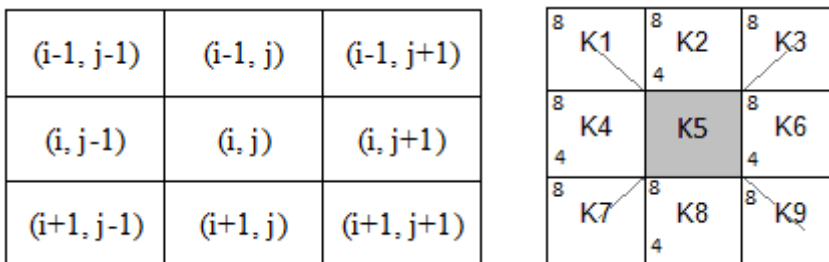
2.6 Şablon Kullanarak Eşleme

Şablon eşleme (template matching) yöntemi, şablon olarak belirlenmiş bir nesneyi, arama ortamında yakalamaya çalışır.

Şablon eşleme yöntemi, en çok rakam ve harf gibi küçük ve basit nesnelere tanınmasında kullanılır. Ayrıca, bir mobil robotu yönlendirmede, kalite kontrolde, yüz tanımda, medikal görüntü işleme gibi birçok uygulamada da bu yöntemden faydalanılmaktadır. Uygulama kolaylığı sebebiyle tercih edilen bu yöntemin dezavantajı, ne arandığının tam olarak bilinmesinin gerekliliği nedeniyle dinamik uygulamalar için yöntemin uygulanabilirliğinin olmaması, şablon resmin ölçülerinin daima, arama ortamı resminin ölçülerinden küçük veya eşit olması zorunluluğu ve şablon resmin tamamının eşleme için aranması yani, şablon içinden bir parçanın ayrıca eşlemesinin yapılamamasıdır.

Bu çalışmada kullanılma sebebi ise, uygulama kolaylığı ve anahtar noktalar etrafındaki iki boyutlu komşuluk ilişkilerini (Bkz. Ek 4). incelemek için çok uygun bir yöntem olmasıdır.

Komşuluk kavramı Şekil 2.18'de görüldüğü gibi K5 pikselinin etrafındaki bitişik diğer pikselleri ifade eder. Komşular olarak, 4-komşu (K2, K4, K6 ve K8) veya 8-komşu (K1, K2, K3, K4, K5, K6, K7, K8 ve K9) seçilebilir.

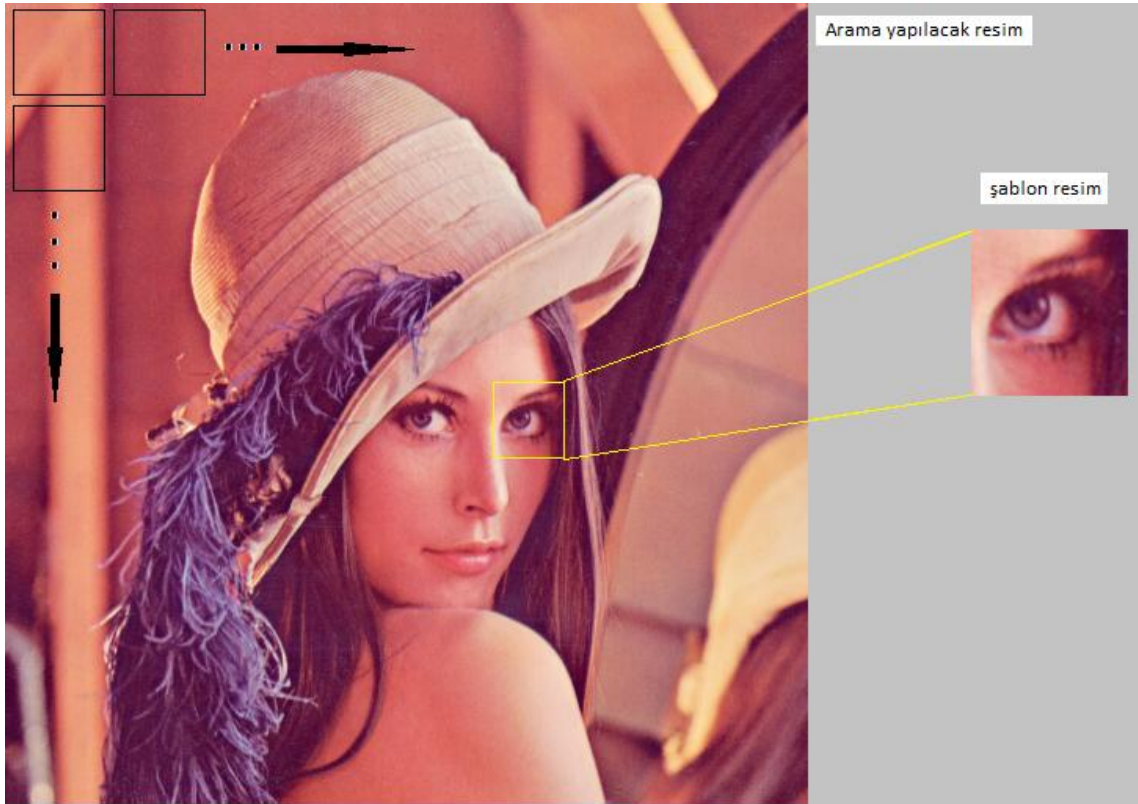


Şekil 2.18 (i, j) koordinatında bulunan K5 merkez pikseli ve komşularının K5'e göre gösterimi

2.6.1 Şablon ile eşlemenin genel özellikleri

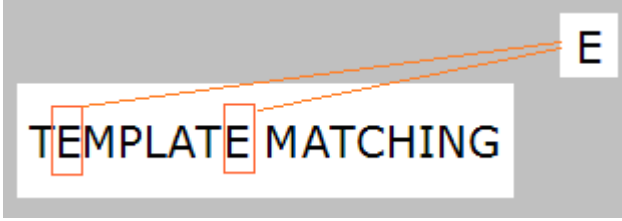
Aranacak nesnenin tam olarak bilinmesi gerekliliği, şablon eşleme yöntemini dinamik uygulamalar için kullanışsız kılmaktadır. Şablon resmin tamamı eşleme için aranır, yani şablon içinden bir parçanın ayrıca eşlemesi yapılamaz. Arama yapılacak ortamın aranan nesneyi, şablona çok yüksek benzerlik oranı ile buldurması halinde ancak eşleme yapabilir. Yöntem en yüksek başarıyı, resmin kendisinden alınan bir parçanın yine aynı resim üzerinde aradığı durumlarda verir. Eşleme yakalanabilmesi için, aranan nesne, arama ortamında aynı veya daha küçük ölçülerde bulunmalıdır.

Aranan şablon, arama yapılacak resim üzerinde, sol üst köşeden başlanarak, sağ alt köşeye kadar mümkün olan tüm pozisyonlarda kaydırılarak, resimler arası piksel-piksel bir sayısal değer hesaplaması yapılır (Şekil 2.19). Hesaplanan bu sayı, şablon ile resim parçasının eşitlik değerini belirtir. Resim ile şablon arasındaki en yüksek uyum, yani en iyi eşleşme aranır. Kullanılan yöntem göre bu sayı ya 1'e ya da 0'a yaklaştıkça benzerlik artar.



Şekil 2.19 Lena resmi üzerinde şablon eşleme algoritmasının uygulanması

Yazı karakterleri tanıma gibi amaçlarla iki seviyeli resimlerde, şablon eşleme Evet/Hayır yaklaşımını kullanır (Şekil 2.20). Gri seviye resimlerde, seçilen benzerlik ölçümlerine göre, benzerlik hesaplanarak, en yakın eşleme bulunur.

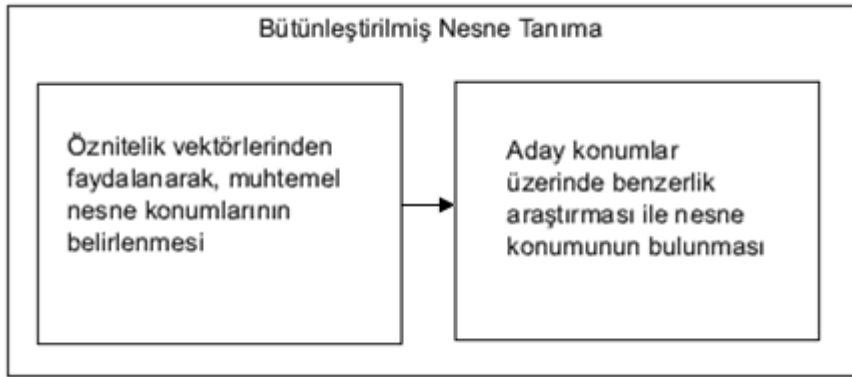


Şekil 2.20 Yazı karakteri tanıma amacıyla şablon eşleme algoritmasının uygulanması

2.7 Bütünleştirilmiş Nesne Tanıma

Çalışmalar sonucu görülmüştür ki, sadece SIFT öznitelik vektörleri ile yapılan nokta nokta eşleştirmeler, benzer eğimlere sahip başka resim bölgelerinde de eşleşmeler bulabilmektedir. Eşleşmeler yüksek hata oranlarına (sonuçlar bölümünde istatistiksel veriler ayrıca sunulmuştur) sahip olmasa da eşleştirmenin doğruluğunun arttırılması için bir çözüm arayışına geçilmiştir.

Karşılaşılan problemler üzerine tasarlanan, B.N.T. (Bütünleştirilmiş Nesne Tanıma) yönteminde, SIFT öznitelikleri kullanılarak yapılan ön eşleme sonuçları, şablon eşleme yöntemi ile kontrol edilmiş ve gerçekte doğru olmayan eşleşmeler elenmiştir. Bu şekilde oluşturulan, B.N.T. yönteminin ana parçaları Şekil 2.21'de, yöntemin algoritma akış diyagramı ise Şekil 3.1'de gösterilmiştir.

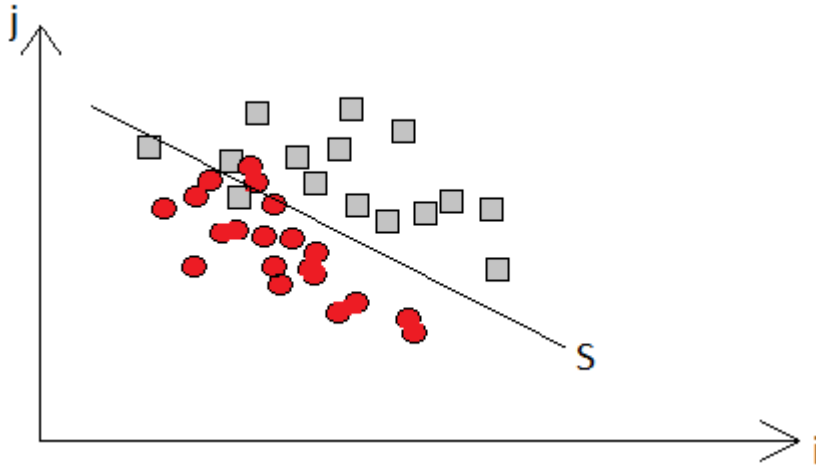


Şekil 2.21 B.N.T. yönteminin ana parçaları

SIFT ile eşleştirilen noktaların birden fazla nesneye işaret ettiği durumlarda, en yüksek benzerlik oranına sahip eşleşme bölgeleri ayrılmıştır ve eşleşmenin tek nesne üzerinde olması sağlanmıştır.

2.8 Benzerlik Ve Sınıflandırma

İki vektörün benzerliğinin araştırılmasında, vektör elemanlarının benzerliğine bakılır. Vektörler arası benzememe (dissimilarity) oranı mesafe ölçümleri ile hesaplandığında, en küçük fark, en yüksek benzerliği (similarity) gösterecektir. Bu ölçümlere göre araştırılan elemanın hangi grupta yer aldığına karar verilerek sınıflandırma yapılır (Şekil 2.22).



Şekil 2.22 İki kategoriye sınıflandırma

Genel mesafe ölçümü ifadesi, Eşitlik 2.16'da gösterilmiştir.

$$D_p(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (2.17)$$

Eşitlik 2.16'daki p değişkeninin aldığı değere göre bulunan mesafe, farklı isimler alır. Mesafe ölçümü için kullanılan yöntemlerden bilindik birkaç tanesi; mutlak farklar toplamı, kareler toplamı ve en büyük mutlak farktır.

Mutlak farklar toplamı, SAD (Sum of Absolute Differences), aynı zamanda Taxicab, City Block veya Manhattan Mesafesi olarak da bilinir.

$$D_1(I_j, g) = \sum_{i=1}^n |I_{i,j} - g_i| \quad (2.18)$$

Kareler farkları toplamı, SSD (Sum of Squared Differences), aynı zamanda Euclidean Mesafesi olarak da bilinir.

$$D_2(I_j, g) = \sqrt{\sum_{i=1}^n |I_{i,j} - g_i|^2} \quad (2.19)$$

En büyük mutlak fark, MaxAD (Maximum Absolute Difference), aynı zamanda Uniform Mesafesi olarak da bilinir.

$$D_\infty(I_j, g) = \lim_{x \rightarrow \infty} \sum_{i=1}^n |I_{i,j} - g_i|^x \quad (2.20)$$

Eşitlik 2.19'daki ifade basitleştirilirse

$$D_\infty(I_j, g) = \max_i |I_{i,j} - g_i| \quad (2.21)$$

Chebyshev, Minkowski, Hamming, Mahalanobis mesafeleri gibi başka ölçümler de kullanılmaktadır.

Euclidean mesafesi, mesafe ölçümleri arasında, en yaygın kullanılanıdır. İki veya üç boyutlu uzayda kullanılabilen bir yöntemdir [14]. Benzerlik ölçümü olarak, Euclidean mesafesi kullandığımızda, iki gri seviye resim arası fark ifadesi aşağıdaki gibi olur.

$$d(I, g, r, c) = \sqrt{\sum_{i=1}^n \sum_{j=1}^m (I(r+i, c+j) - g(i, j))^2} \quad (2.22)$$

I : arama yaptığımız resim

g : n x m ebatlarındaki şablon resim

(r,c) : şablon resmin sol üst köşesinin koordinatı

Şablon resmi arama ortamı üzerinde gezdirdiğimizde, her bir hareket için bir ilişiklik değeri hesaplanır. İlişiklik öngörülerin gerçeğe ne kadar yakın olduğunun göstergesidir. İlişikliğin yüksek olması için, karşılaştırılan verilerin uyumlu şekilde artıp azalması beklenir. Şablon ile o anda şablon altında kalan arama ortamına ait resim parçası arasındaki ilişiklik karşılıklı pikseller üzerinden hesaplanır. Gri seviye resimler için ilişiklik ifadesi

$$cor = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1} (y_i - \bar{y})^2}} \quad (2.23)$$

olarak yazılabilir. Burada:

x : şablon resim

\bar{x} : şablon resimdeki ortalama gri seviyesi

y : arama yapılan resmin, ilgili parçası

\bar{y} : arama yapılan resmin ortalama gri seviyesi

N : ilgili resim parçasındaki piksel sayısı aynı zamanda şablon resmin ebatı
(sıra x sütun)

cor : İlişiklik katsayısı

olmaktadır. cor deęeri, -1 ile +1 arasında deęer alır. İki resim arasındaki benzerlik arttıkça, bu deęer de büyür.

İlişki hesaplama sonucu her defasında sol üst pikselde kaydedilirse, bir benzerlik haritası elde edilir. En büyük benzerlięin (veya minimum farkın) olduęu yeri bulmak için, ilişiklik haritasındaki maksimum noktalar hesaplanır. Parlaklık deęerinin fazlalığı, şablon ile resim arasındaki benzerlięin büyüklüğünü gösterir.

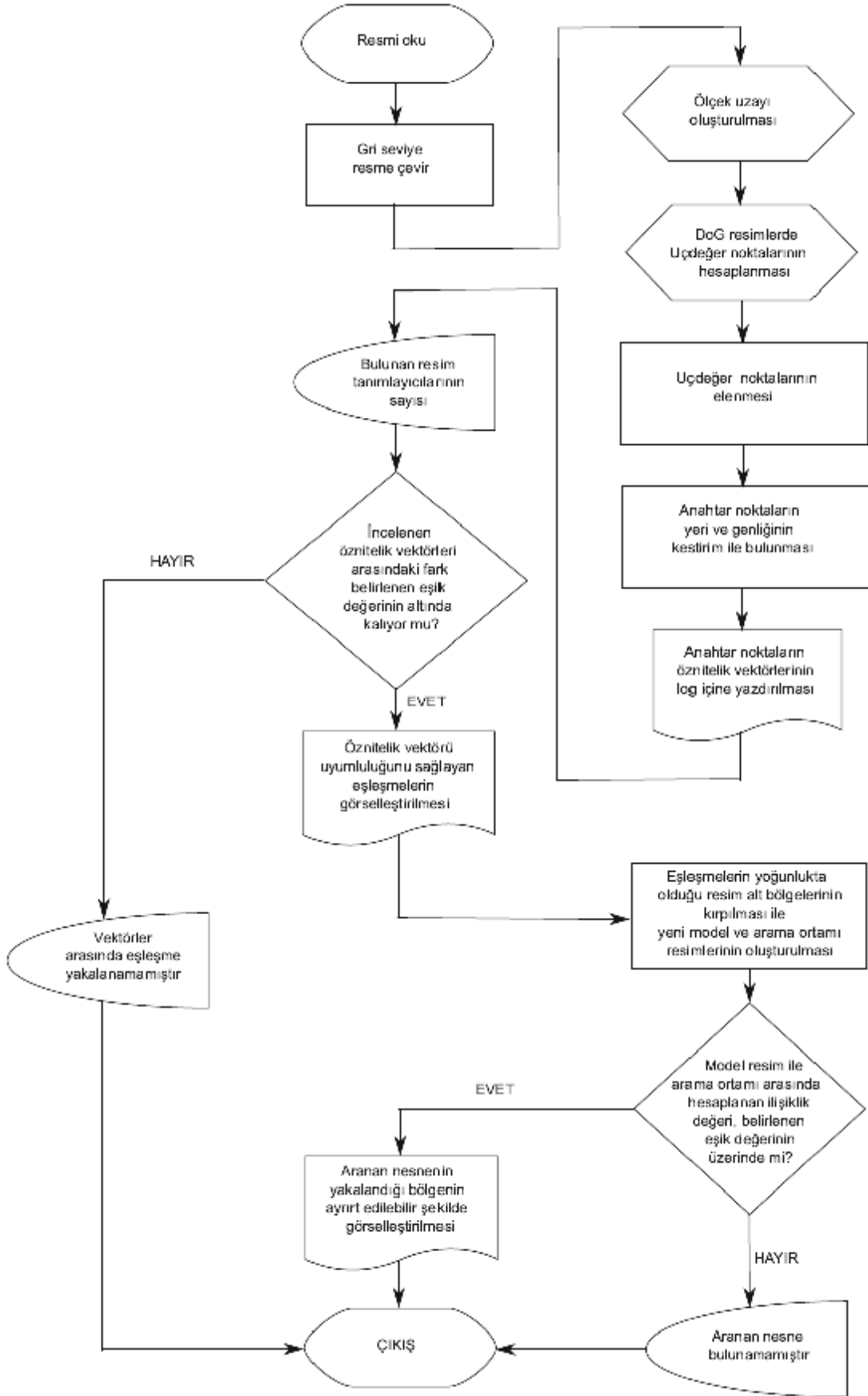
3. YAPILAN ÇALIŞMALAR

Çalışmada kullanılan resimler Caltech 101 Office Datasets [42]'inden alınmıştır. Verisetini oluşturan resimler JPEG formatında, 1280x960 ve 2200x1700 çözünürlüklerindedir. Veriseti 8 ayrı nesnenin, farklı bakış açılarından alınmış fotoğraflarından ve 51 adet birden çok nesnenin bulunduğu, karışık test resimlerinden oluşmaktadır. Veriseti renkli resimlerle sunulmuş fakat bizim çalışmamızda verisetinin eski versiyonu olan gri seviye resimler kullanılmıştır.

Çalışmada veriseti üzerinde yapılan işlemler sırası ile şöyledir:

- Anahtar noktaların belirlenmesi.
- Anahtar noktalara ait öznitelik vektörlerinin, hem model nesne hem de arama ortamı için çıkarılması.
- Vektörler arasında uyumluluğun en küçük fark hesaplanması ile aranması.
- Uyumlu vektörlerin ait olduğu anahtar noktaların, model nesne ve ortam resmi arasında karşılıklı olarak eşleştirilmesi.
- Eşleşme yakalanan noktalar etrafında, bir doğruluk testi yapılması ile, ortam resminde konumu belirlenen aranan nesnenin gösterilmesi.

B.N.T yöntemine ait algoritma akış diyagramı Şekil 3.1 'de verilmiştir.



Şekil 3.1 B.N.T yöntemine ait akış diyagramı

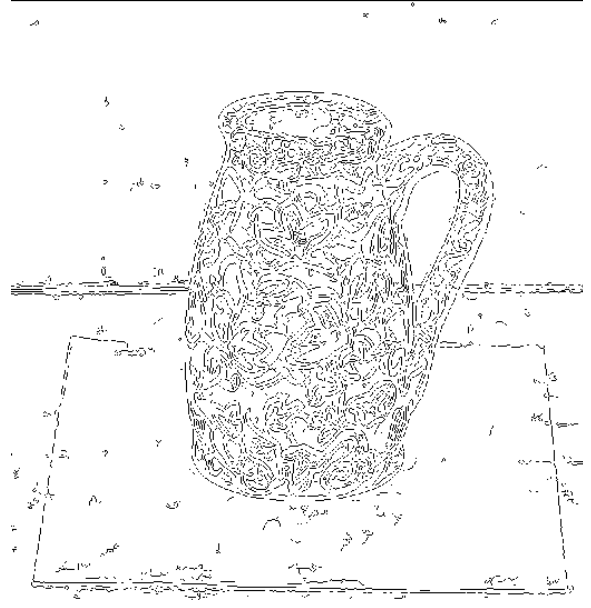
3.1 Bütünleştirilmiş Nesne Tanıma Yönteminin Uygulanması

Resim tanımlayıcıları olan anahtar noktaların, ayırt edici noktalar olması gerekmektedir. Anahtar noktaların köşe noktaları içinden belirlenmesi, köşe noktalarının özelliklerinden dolayı tercih edilmiştir. Kenarlar üzerinden veya düzlemsel bölgeler içinden seçilecek noktalar, köşeler kadar ayırt edilebilir özelliklere sahip olmayacaktır.

Resim üzerinde bir kenar ve köşe bulma algoritması çalıştırarak işe başlamak gerekirse, resim üzerinde bazı ön işlemler yapmak yararlı olacaktır. Şekil 3.2'de görülebileceği gibi ön işlemler olmadan resim üzerinde kenar bulmak, fazladan kenar ve köşe noktaları getireceğinden, bunları birbirinden ayırt etmek de o derece zorlaşacaktır.



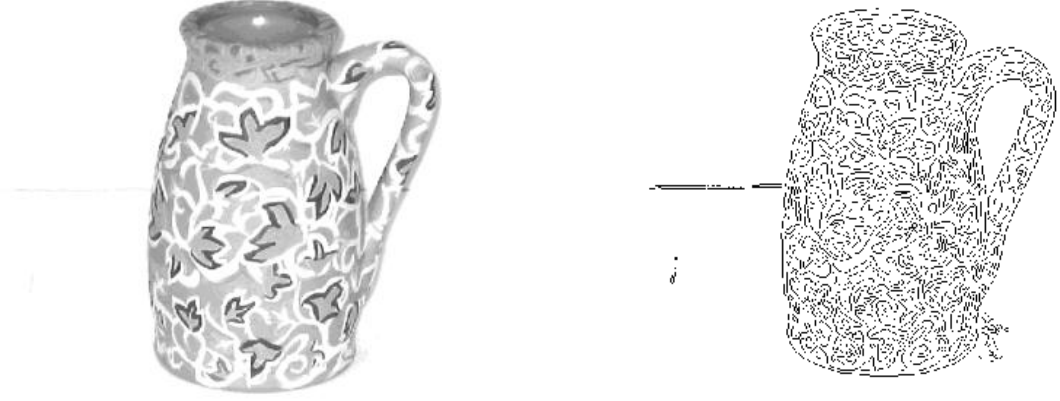
(a) Model nesnenin orjinal resmi



(b) Resim üzerinde ön işleme yapılmadan elde edilen bir kenar bulma işlemi sonucu

Şekil 3.2 Model resim üzerinde kenar bulma

Ön işleme yapmanın gerekliliği kesinleştikten sonra, önce gürültü azaltımı, sonra kenar iyileştirme (enhancement) yapılmış bir resim üzerinde kenar bulma algoritması çalıştırıldığında elde edilen sonuç Şekil 3.3'de görülmektedir.



(a) Ön işlemeden geçirilmiş model nesne

(b) Ön işleme yapıldıktan sonra kenar bulma işlemi sonucu

Şekil 3.3 Model resim üzerinde ön işlemeden sonra kenar bulma

Resim üzerinde yapılan ön işlemler ve kenar keskinleştirme, Şekil 3.4'de görülmektedir.

Kenarları iyileştirilmiş resimler üzerinde uçdeğerler aranacaktır. Uçdeğerlerden köşe noktaların belirlenmesi için Harris sezinleyicisi [15] veya Shi and Tomasi [49]'nin en küçük özdeğer yöntemi tercih edilebilir. Harris sezinleyicisi doğruluktan ödün vererek hesaplama maliyetini düşürerek çalışmaktadır. En küçük özdeğer yönteminin ise hesaplama maliyeti daha yüksek ama bununla birlikte özdeğerleri hesapladığı için doğruluğu Harris sezinleyicisine göre daha yüksektir. Çalışmalarımızda en küçük özdeğer yöntemi tercih edilmiştir.



(a) Model nesnenin orjinal resmi



(b) Gauss fonksiyonu ile bulanıklaştırılmış resim



(c) Orjinal resim ile gauss farkının alınması (resmin algılanabilmesi amacıyla, resim log transformundan geçirilmiş ve negatifi alınmıştır)



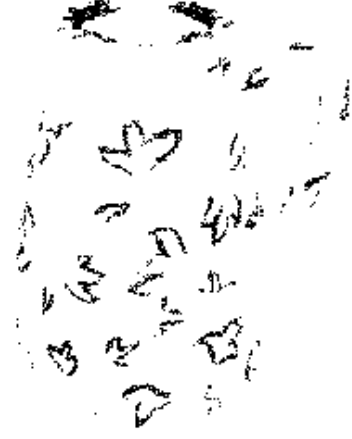
(d) Kenar keskinleştirme, (a) ve (c)'deki resimlerin matris ifadelerinin toplanması sonucu gerçekleştirilmiştir.

Şekil 3.4 Model resimde ön işlemler ve kenar iyileştirme

Kenar iyileştirilmiş resim üzerinde uçdeğerlerin (bölgesel en küçük ve bölgesel en büyük değerlerin bütünü) belirlenmesi ve köşe noktalardan anahtar noktaların seçilmesi işlemleri Şekil 3.5'de gösterilmiştir.



(a) Kenar iyileştirilmiş resim üzerinde bir eşik değeri olmadan belirlenen uçdeğerler



(b) 8 komşuluk için, bölgesel uçdeğerler



(c) Köşe metrik matrisi, muhtemel köşeleri göstermektedir.



(d) Anahtar nokta olarak atanacak, köşe noktalarının, belirlenmesi

Şekil 3.5 Anahtar noktaların belirlenmesi

Anahtar noktalara ait öznitelik vektörlerinin çıkarılması SIFT yöntemine göre yapılmıştır. Üçüncü bölümde anlatılan SIFT aşamalarına göre, uygulamaya önce ölçek uzayı oluşturularak başlanmalıdır. Lowe [26], dört oktav ve beş bulanıklaştırma önerdiğinden, çalışmalar bunun doğrultusunda gerçekleştirilmiştir.

Özniteliklerini bulmak istediğimiz resim, beş kez ard arda, Gauss ile filtrelenir ve her yeni resim yarı boyutuna ölçeklenir. Dört farklı boyutlu resim kümesi yani dört oktav içinde bulunan beşer adet bulanık resim elde edilir. Bu işlemler sonucu, toplamda yirmi resimden oluşan bir ölçek uzayı elde edilmiş olur. Filtrelenmiş ve boyutu indirgenmiş resimde, resim bilgisine ait kayıplar en az ölçüde olmaktadır. Filtrelenerek resim üzerinde daha geniş alana yayılan bilgiler, resmin boyutları azaldığında da resim içerisinde bilgi kalmasını sağlar.

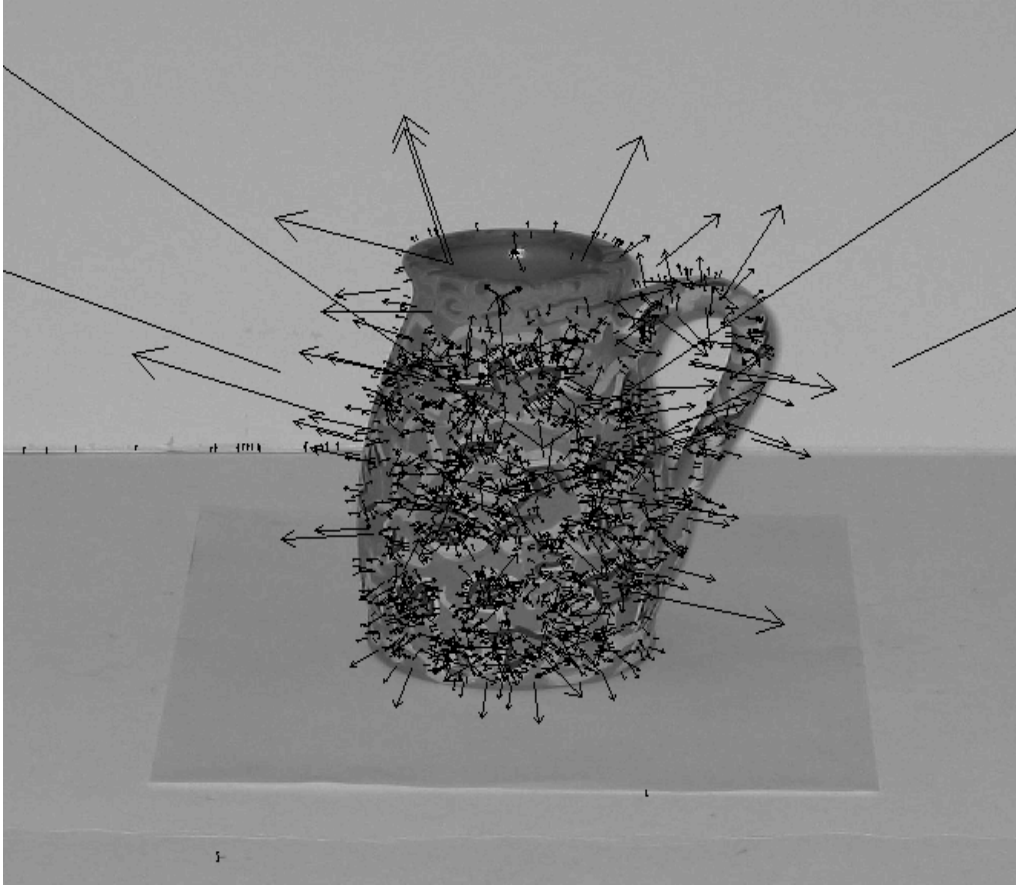
Ölçek uzayı oluşturduktan sonra, DoG resimler elde edilmelidir. DoG piramidi her oktavda, fark alma işlemlerinin yapılması ile oluşturulur. Bulanıklaştırılmış resim orijinalinden çıkarıldığında, ani parlaklık değişimlerinin olduğu noktaları belirginleştiren bir yüksek geçiren filtre gibi sonuç verir. Dolayısıyla, DoG işlemi, resim üzerindeki kenarlar ve köşeleri belirginleştirir (Şekil 3.4). Uçdeğerler belirginleştirilen kenar ve köşelerden elde edilir (Şekil 3.5). Bu uçdeğerler anahtar nokta olmaya aday noktalarıdır.

DoG piramidi ile, uçdeğer araştırmasında kullanılan komşu sayısı 26'dır. Bir noktanın uçdeğer olup olmadığına, bulunduğu bölgede en büyük veya en küçük sayısal değere sahip olmasına göre karar verilir. Değerlendirme pikselin bulunduğu ortamdaki 8-komşusu ve piramitteki konumuna göre üst ve alt ortamlardaki 9'ar komşusu arasında yapılır (Bkz Ek 4).

Şekil 3.5 'de görünen uçdeğer çıkarımı, sadece ilk oktavda bulunan orijinal resim ve bulanıklaştırılmış diğer resim arasında gösterilmiştir. Dolayısı ile tek ortam için 8-komşu arasında uçdeğer araştırması yapılmıştır. Bu işlem, tüm oktavlar altında, komşu tüm resimler arasında, 26 komşuluk göz önüne alınarak yapılır.

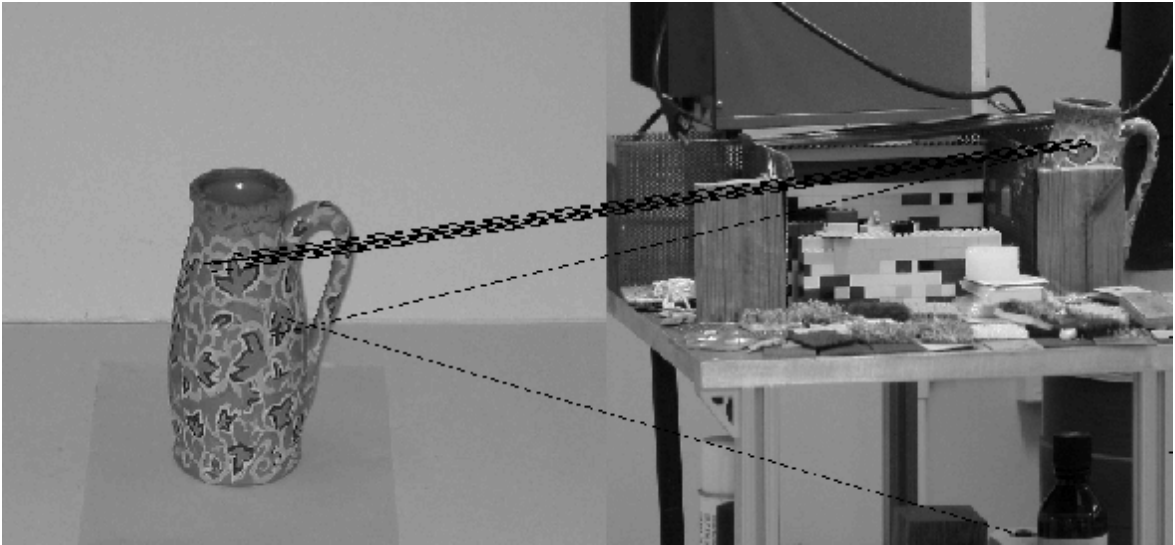
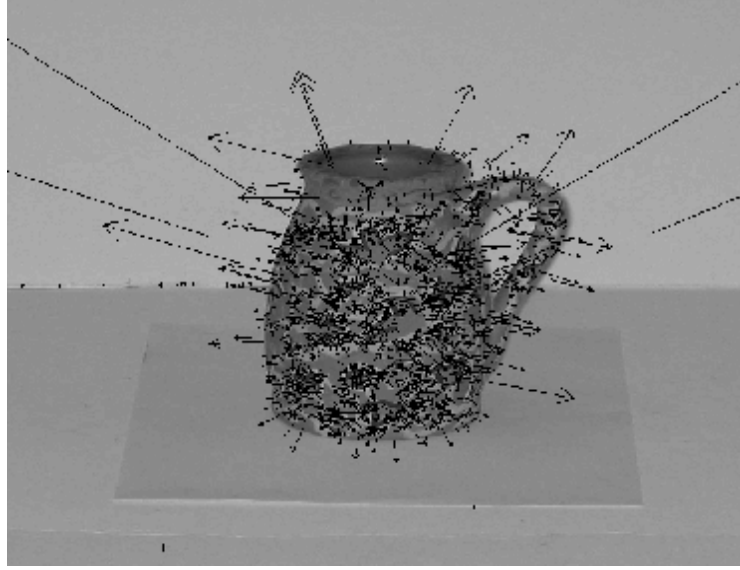
Uçdeğer olarak seçilen noktalar üzerinde, üçüncü bölümde açıklandığı gibi iki aşamalı bir eleme yapıldığında geriye anahtar noktalar kalır (Şekil 3.5 (d)).

Anahtar noktalar üzerinde çıkarılan öznitelikler Şekil 3.6'da ikinci dereceden eğimler hesaplanarak, eğimin genlik ve yöneline göre, bölgesel normalize edilmiş vektörler ile gösterilmiştir. Bölgesel resim özellikleri kullanıldığı için, bu vektörler oryantasyondan bağımsız tanımlayıcılardır.



Şekil 3.6 Öznitelik vektörlerinin model nesne üzerinde gösterilmesi

Şekil 3.7'de model nesneye ait bir resim ve arama ortamına ait ikinci bir resim gösterilmiştir. Resimler üzerinde elde edilen öznitelik vektörleri görülmektedir. Bu vektörlerin anahtar noktaları eşleme için kullanımları da aynı şekil altında görülmektedir.



Şekil 3.7 Öznitelik vektörlerinin ilgili resimler üzerinde eşleme için kullanılması

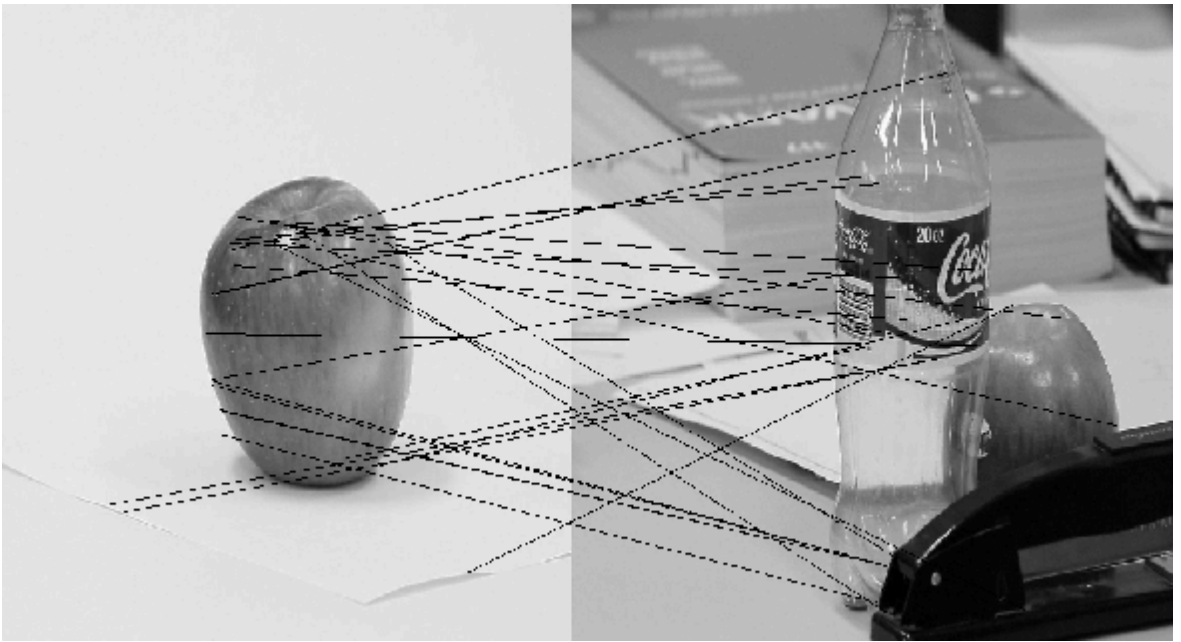
Şekil 3.7’de görünen özellik çıkarımı ve eşleştirme sırasında, resimlerde sırasıyla, 1975 ve 4895 anahtar nokta belirlenmiş ve 11 eşleşen nokta bulunmuştur.

Anahtar nokta tanımlayıcıları arasında uygunluk ararken, uygunluk için aradığımız en küçük fark değerini yukarı çekerek, eşleşme sayısını arttırabiliriz. Resimler arası hiç eşleşme bulunamadığı bazı durumlarda, vektörler arası fark değerinin artması ile ciddi oryantasyon farklılıkları ve başka nesnelere tarafından engellenme durumlarında da istenilen sonuçlar yakalanabilir.

Örneğin, verisetinden birinci nesneye ait train3 ve test3 resimlerini kullanarak, elma gibi yüzeyinde ve dış hatlarında belirgin ayrımlar bulunmayan nesnelere için eşleme aranırken Şekil 3.8 (a)’da görüldüğü gibi fark değeri %60 iken hiçbir eşleşme bulunamayabilir. Şekil 3.8 (b)’de, vektörler arası farkın arttırılması ve %85’e çıkarılması ile eşleşmeler yakalanabilmektedir.



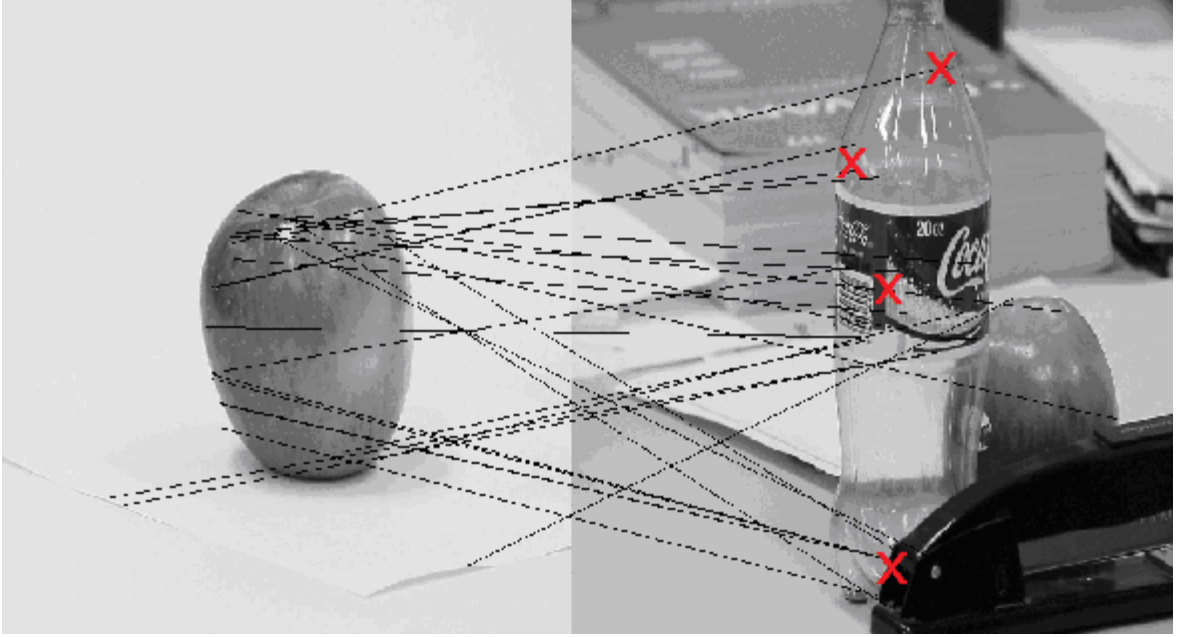
(a) Düşük fark ile eşleşme yakalanamaması



(b) Farkın büyütülmesi ile eşleşme yakalanması

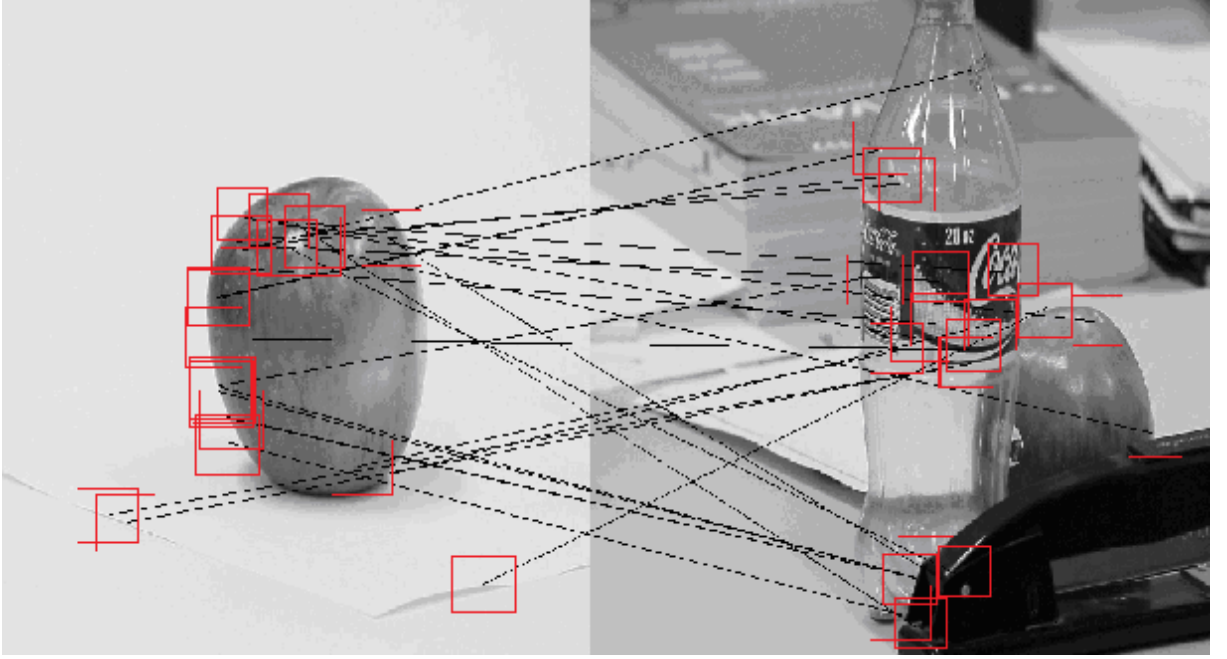
Şekil 3.8 Eşleşme miktarının vektörler arasında aranan fark ile değişimi

Eşleşmeler yakın özellikler gösteren anahtar noktalar arasında yapıldığından, bir kısım eşleşmeler gerçekte yanlıştır (Şekil 3.9). Eşleşmelerin anlamı, nesnenin muhtemel konumlarını arama ortamı üzerinde göstermektir. Yanlış eşleşmeleri doğru olanlardan ayırmak için üçüncü bölümde bahsedilen şablon eşleme algoritmasından faydalanılmıştır.



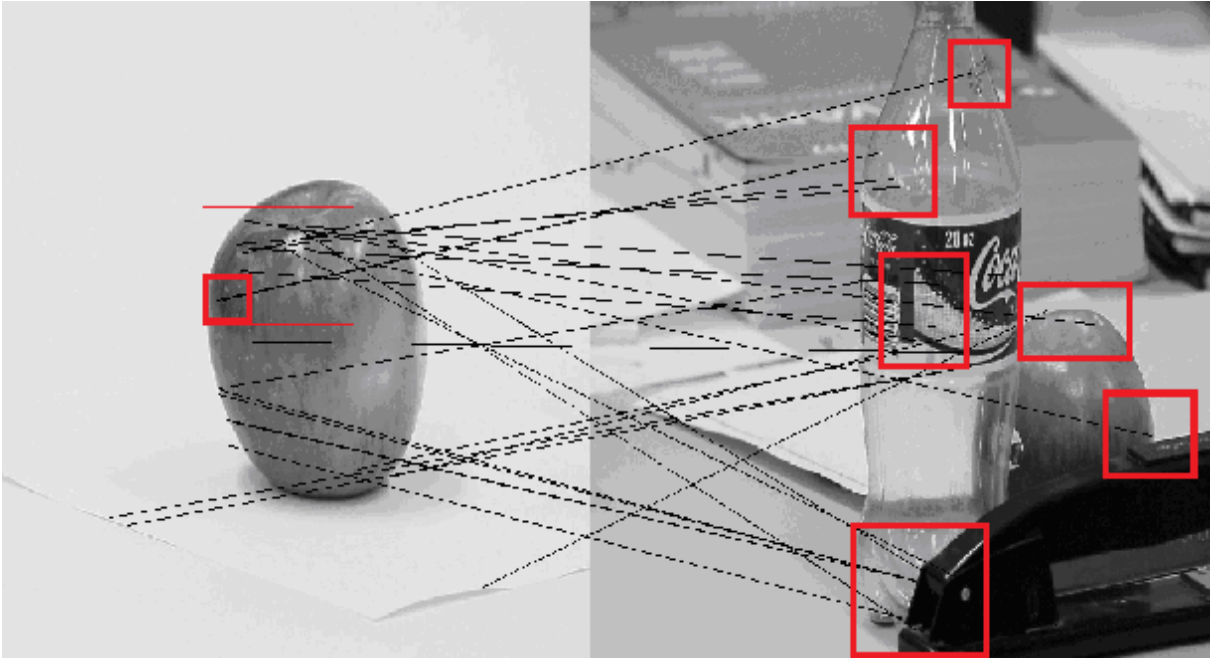
Şekil 3.9 SIFT öznelikleri kullanarak muhtemel nesne konumlarının eşleştirme ile gösterilmesi

Hem model nesnede hem de arama ortamında, eşleşen anahtar noktalar etrafında seçilebilecek tüm bölgeler Şekil 3.10'da görülmektedir. Birbirine yakın noktalara ait bölgeler yüksek kesişim oranına sahip olduğundan, anahtar noktaların yoğun olarak konumlandığı bölgelerde, aynı bölgelerin tekrar tekrar incelenmesini engellemek, dolayısı ile işlem maliyetini azaltmak amacıyla, kesişim oranı %60'ın üzerinde olan bölgeler birleştirilmiştir (Şekil 3.11)



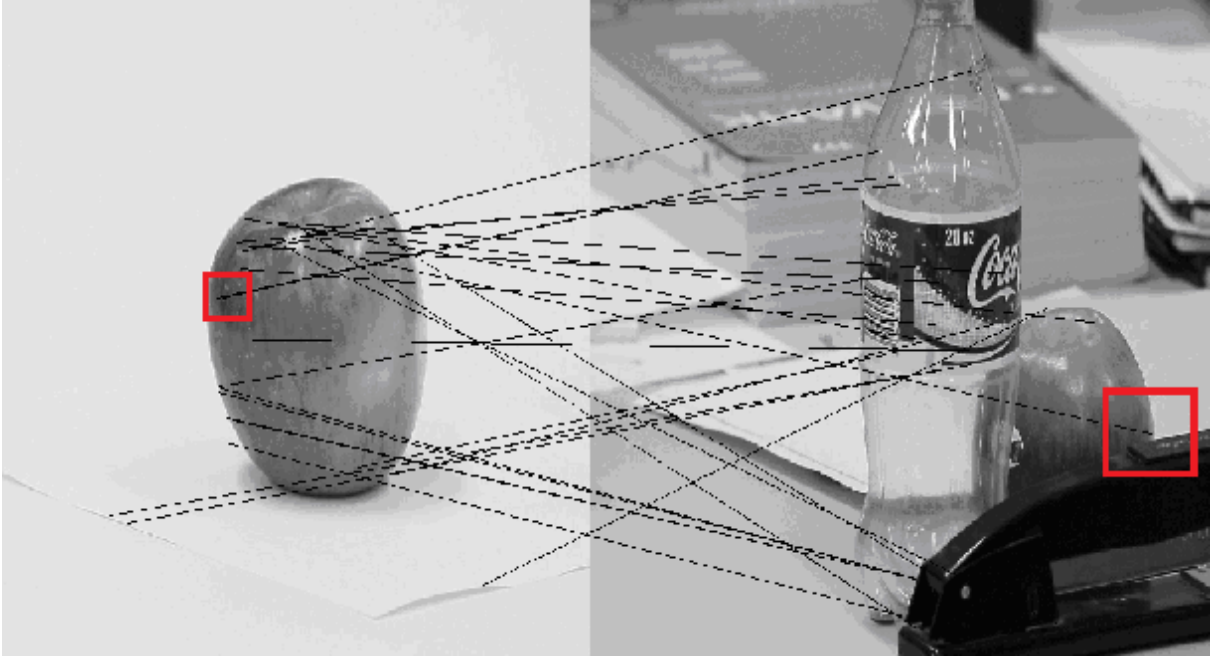
Şekil 3.10 Eşleşen noktaların etrafında belirlenen tüm bölgeler

Şekil 3.11'de görünen, model nesneden kırılan bölge, şablon resmi; arama ortamından kırılan bölge ise, yeni arama ortamını oluşturur (Şekil 3.13)



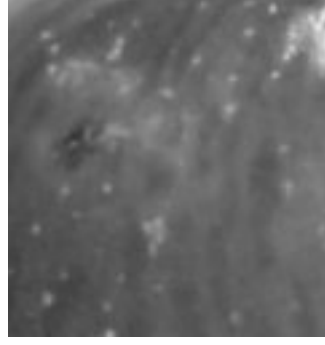
Şekil 3.11 Şablon ve arama ortamı resimleri üzerinde seçilen ve kırılan bölgeler

Şekil 3.11 değerlendirildiğinde benzerliğin Şekil 3.12'deki bölgede yakalanması beklenmektedir.

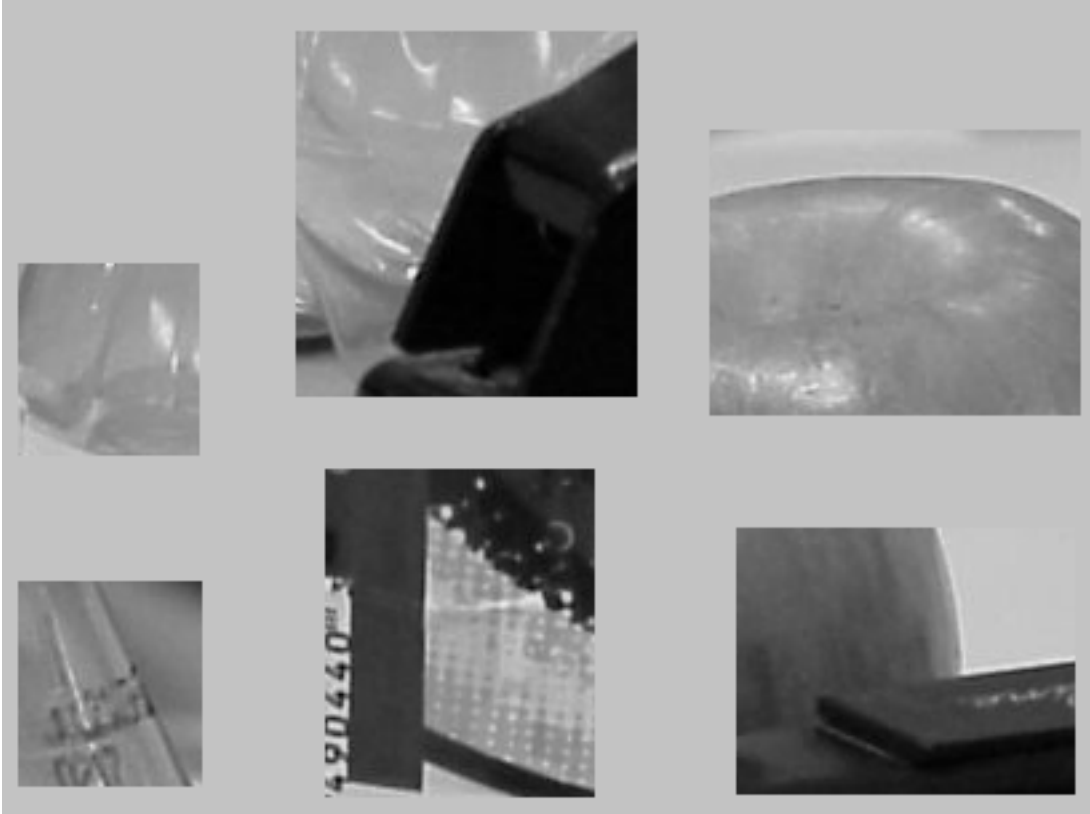


Şekil 3.12 Benzerliğinin en çok olması beklenen arama bölgesi

Şekil 3.11' de görünen, noktaların yoğunlaştığı bölgelerden alanların seçilmesi sonucu elde edilen, şablon (Şekil 3.13 (a)) ve yeni arama ortamı (Şekil 3.13 (b)) arasında bir ilişiklik araştırmasına gidilirse, Şekil 3.14'deki gibi 0-1 arasında değişen değerler elde edilir.

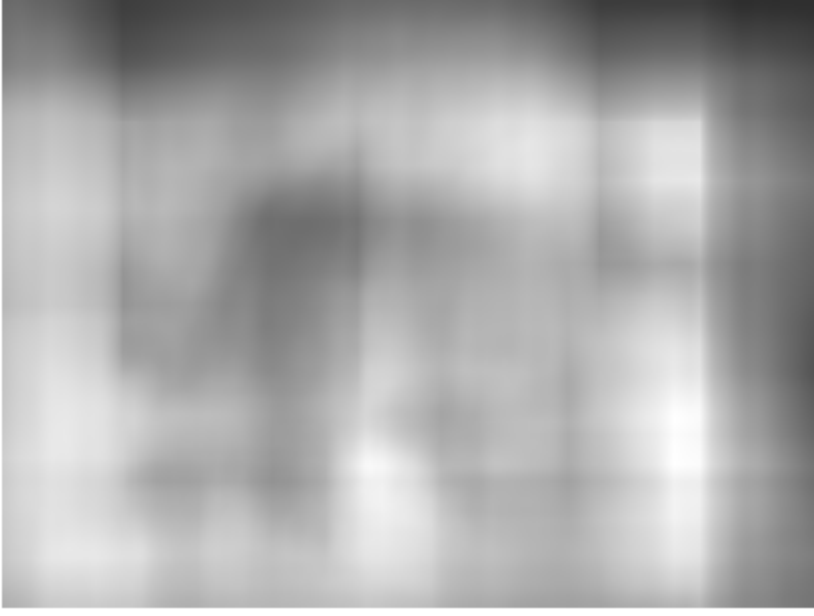


(a) Şablon resim



(b) Yeni arama ortamı

Şekil 3.13 Şablon eşleme için kullanılacak şablon ve arama ortamı



Şekil 3.14 Şablon eşlemede ilişikliğin frekans uzayında gösterilmesi

İlişikliğin 0 a yakın olması benzerliğin olmadığını, 1 e yaklaşması ise benzerliğin arttığını gösterir. Dolayısı ile parlak bölgeler ilişikliğin, yani benzerliğin fazla olduğu bölgelerdir (Şekil 3.14).

İlişikliğin bir eşik değerinden geçirilmesi ile istenen benzerlik oranı belirlenebilir. Eşik değeri olarak 1'e yakın değerlerden, 0.80, 0.90, 0.98 ve 0.99 seçerek benzerlik bölgelerinin değişimini Şekil 3.15'de gözlemleyebiliriz.



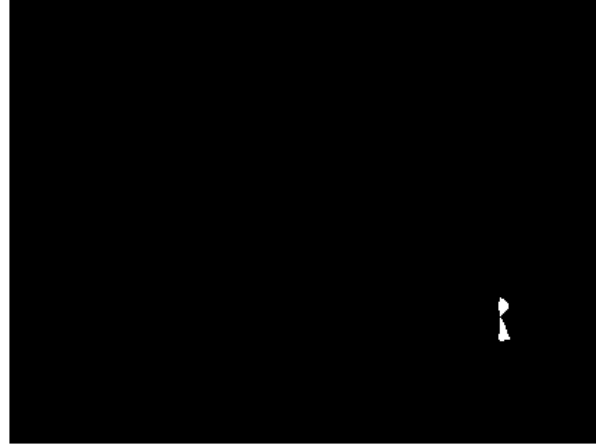
(a) Eşik değeri : 0.80



(b) Eşik değeri : 0.90



(c) Eşik değeri : 0.98



(d) Eşik değeri : 0.99

Şekil 3.15 İlişiklik değerlerinin bazı eşik değerlerinden geçirilmesi

Eşik değerinin yüksek tutulması ile benzerliğin en fazla olduğu bölgeler yakalanabilir. Eşik değeri 1'e en yakın değer olan 0.99 seçildiğinde veya sayısal değerler içinden en büyük değer eşik değeri olarak atandığında, belirlenen bölgeye ait orta nokta, orjinal arama ortamı üzerinde Şekil 3.16'daki gibi işaretlenerek gösterebilir.



Şekil 3.16 Seçilen model ile benzerliği en çok olan noktanın orjinal arama ortamı resminde gösterilmesi

Ek 5’de kullanılan verisetindeki diğer resimlere ait görsel sonuçlar verilmiştir.

3.2 Veriseti Üzerinde Elde Edilen Sonuçlar

Verisetinde bulunan toplam 8 nesnenin her biri için, bir bakış açısından görünen fotoğrafı model resim olarak alınmış (Şekil 3.17) ve 51 adet test resminde (Bkz. Ek 6) anahtar nokta eşleştirme ve sonrasında şablon eşleme işlemleri sırasıyla gerçekleştirilmiştir. İki işleme ait performans değerlendirmesi Çizelge 3.1 de verilen karışıklık matrisi (confusion matrix) ile yapılmıştır.



(a) 1. Nesne



(b) 2. Nesne



(c) 3. Nesne



(d) 4. Nesne

Şekil 3.17 Performans testlerinde kullanılan model resimler



(e) 5. Nesne



(f) 6. Nesne



(g) 7. Nesne



(h) 8. Nesne

Şekil 3.17 devam ediyor

Çizelge 3.1 Performans değerlendirmesi için kullanılan karışıklık matrisi

		Eşleme algoritması sonucu	
		var	yok
Gerçekte	var	TP	FN
	yok	FP	TN

Çizelge 3.1 de görülen kısaltmalar :

- Doğru pozitif , TP (true-positive): Gerçekte var olanın program tarafından da var olarak bulunması
- Yanlış negatif, FN (false-negative) : Gerçekte var iken, program tarafından yok olarak bulunması
- Yanlış pozitif, FP (false - positive): Gerçekte yok iken, program tarafından var olarak bulunması
- Doğru negatif, TN (true-negative): Gerçekte yok olanın, program tarafından da yok olarak bulunması

Sınıflandırma var-yok olarak yapılmaktadır.

Var: aranan nesnenin arama ortamında bulunduğu;

Yok: aranan nesnenin arama ortamında bulunmadığını göstermektedir.

Sistemin başarısı, doğru sınıfa ve yanlış sınıfa atanan örnek sayısı ile alakalıdır. Karışıklık matrisinde satırlar test kümesindeki örneklere ait gerçek sayısal değerleri, kolonlar ise modelin tahminlemesini gösterir.

Belirlenen bu çizelge değerleri ile programın genel başarı değerlendirmesi eşitlik (3.1), (3.2), (3.3) ve (3.4) 'deki, veri madenciliğinde kullanılan model başarımlarının ölçütlerinin hesaplanması ile yapılabilir.

$$\text{Doğruluk} = \frac{\text{Sistem tarafından yapılan doğru sınıflandırma sayısı}}{\text{Kullanılan toplam örnek sayısı}} \quad (3.1)$$
$$= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}$$

Doğruluk (accuracy), doğru sınıflandırılmış örnek sayısının, toplam örnek sayısına oranıdır. Hata oranı ise bu değer 1'e tamlanıdır [62].

$$\text{Duyarlılık} = \frac{\text{Dođru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif sınıflandırılmış örnek sayısı}} \quad (3.2)$$

$$= \frac{TP}{TP + FP}$$

Duyarlılık (precision), dođru sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranıdır. Rastgele seçilmiş geri kazanılmış bir dosyanın dođru olma olasılığıdır [62].

$$\text{Geriçađırım} = \frac{\text{Dođru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif örnek sayısı}} \quad (3.3)$$

$$= \frac{TP}{TP + FN}$$

Geriçađırım (recall), aynı zamanda TP oranıdır. Bu oran, bir arařtırmada, rastgele seçilen dođru bir örneđin, geri kazanımı olasılığıdır [62].

$$F - \text{ölçütü} = \frac{2 \times \text{Duyarlılık} \times \text{Geriçađırım}}{\text{Duyarlılık} + \text{Geriçađırım}} \quad (3.4)$$

F-Ölçütü (F-factor, F-measure) ile kesinlik ve duyarlılık ölçütleri bir arada deđerlendirilmiş olur. F-ölçütü, kesinlik ve duyarlılığın harmonik ortalamasıdır. Aritmetik ortalamaya A, geometrik ortalamaya G denirse, F ölçütü harmonik ortalama yani, $2G/A$ olur [62].

3.2.1 Anahtar nokta eşleşirme başarısı

Anahtar nokta eşleşirme işlemine ait algoritmanın performans deđerlendirmesi ařađıda verilmektedir (Çizelge 3.2 - Çizelge 3.10). Algoritmanın başarısı, detaylı resimlerde yüksekken, detayın azalması ile yakalanan eşleşmeler de azalmaktadır. Ayrıca, benzer detaylara sahip, farklı nesnelere ait resim

bölgelerinde eşleşmeler algoritmanın başarısını çok düşürmektedir. Dolayısı ile bir doğrulama kriteri kullanımı zorunludur.

Çizelge 3.2 Birinci nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	8	3	11
	yok	29	11	40
Σ		37	14	51

Çizelge 3.3 İkinci nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	11	0	11
	yok	40	0	40
Σ		51	0	51

Çizelge 3.4 Üçüncü nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	9	0	9
	yok	35	7	42
Σ		44	7	51

Çizelge 3.5 Dördüncü nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	10	0	10
	yok	41	0	41
Σ		51	0	51

Çizelge 3.6 Beşinci nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	9	0	9
	yok	42	0	42
Σ		51	0	51

Çizelge 3.7 Altıncı nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	4	0	4
	yok	35	12	47
Σ		39	12	51

Çizelge 3.8 Yedinci nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	12	0	12
	yok	39	0	39
Σ		51	0	51

Çizelge 3.9 Sekizinci nesne için, nokta eşleştirme performans değerlendirmesi

		Nokta eşleme algoritması sonucu		Σ
		var	yok	
Gerçekte	var	12	0	12
	yok	36	3	39
Σ		48	3	51

Çizelge 3.10 Tüm nesnelere için, nokta eşleştirme performans değerlendirmesi

NESNE	Doğruluk	Duyarlılık	Geriçağırım	F-ölçütü
1	0,3725	0,2162	0,7273	0,3333
2	0,2157	0,2157	1,0000	0,3548
3	0,3137	0,2045	1,0000	0,3396
4	0,1961	0,1961	1,0000	0,3279
5	0,1765	0,1765	1,0000	0,3000
6	0,2667	0,1026	0,3077	0,1538
7	0,2353	0,2353	1,0000	0,3810
8	0,2941	0,2500	1,0000	0,4000

Nokta eşleme yönteminin doğru noktaları yakalaması üzerine verilmiş bu performans değerleri, bir doğrulama sürecinin ne derece gerekli olduğunu göstermektedir. Doğrulama işleminin yöntemine dahil edilmesi ile bu değerler fazlasıyla değişmiştir ve yöntemin kesinliğinin arttığı gözlemlenmiştir.

3.2.2 B.N.T. yönteminin başarısı

Geliştirilen B.N.T. yönteminde, eşleşen noktalar etrafındaki komşu bölgelerde bir şablon eşleme yöntemi uygulanarak, yanlış eşleşmelerin elenmesi sağlanmıştır. Aynı veriseti üzerinde elde edilen sonuçlara ait, yöntemin performans değerlendirmesi aşağıda verilmektedir (Çizelge 3.11 - Çizelge 3.19).

Çizelge 3.11 Birinci nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	var	8	3	11
	yok	1	39	40
Σ		9	42	51

Çizelge 3.12 İkinci nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	var	11	0	11
	yok	0	40	40
Σ		11	40	51

Çizelge 3.13 Üçüncü nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	Var	9	0	9
	Yok	0	42	42
Σ		9	42	51

Çizelge 3.14 Dördüncü nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	Var	10	0	10
	Yok	2	39	41
Σ		12	39	51

Çizelge 3.15 Beşinci nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	Var	9	0	9
	Yok	1	41	42
Σ		10	41	51

Çizelge 3.16 Altıncı nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	Var	4	0	4
	Yok	0	47	47
Σ		4	47	51

Çizelge 3.17 Yedinci nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	Var	12	0	12
	Yok	0	39	39
Σ		12	39	51

Çizelge 3.18 Sekizinci nesne için, B.N.T. performans değerlendirmesi

		B.N.T. algoritması sonucu		Σ
		var	yok	
Gerçekte	Var	12	0	12
	Yok	0	39	39
Σ		12	39	51

Çizelge 3.19 Tüm nesnelere için, B.N.T. performans değerlendirilmesi

NESNE	Doğruluk	Duyarlılık	Geriçağırım	F-ölçütü
1	0.9216	0.8889	0.7273	0.8000
2	1.0000	1.0000	1.0000	1.0000
3	1.0000	1.0000	1.0000	1.0000
4	0.9608	0.8333	1.0000	0.9091
5	0.9804	0.9000	1.0000	0.9474
6	1.0000	1.0000	1.0000	1.0000
7	1.0000	1.0000	1.0000	1.0000
8	1.0000	1.0000	1.0000	1.0000

Veriseti üzerinde yapılan değerlendirmelere göre; yöntemin ara aşamalarındaki kullanıcıya bağlı değerlerdeki (alınan eşik değerleri) değişiklikler, sonucu doğrudan etkilemektedir. Kullanılan resim boyutları da işlem süresini doğrudan etkilemektedir. Resim boyutlarının artması, karışık resimlerde detayların yakalanmasını kolaylaştırırken, piksel piksel araştıran bir algoritma olduğu için de işlem süresini uzatmaktadır. Resim boyutlarının aşamalı değiştirildiği denemeler yapılmış ve sonuçta boyut indirgenmesi ile gelen veri kaybını engellemek için algoritmaya girdi olarak verilen resimler verisetindeki boyutları ile kullanılmıştır.

Anahtar noktaların hesaplanması, algoritma içindeki en karmaşık ve tüm işlem sürecinin yaklaşık %70'ini alan aşamadır. Anahtar noktaların fazla olması engellemeler gibi zorlayıcı koşullarda tanıma olasılığını artırırken, işlem süresini de uzatmaktadır. Öznitelik vektörleri arasında fark hesaplanması aşamasında, istenilen en küçük farkı belirleyen eşik değerinin yükseltilmesi ise, yüksek benzerlik arandığı anlamına gelmekte ve eşleşme şansını düşürmektedir. Bu eşik değerinin eşleşmenin yakalanmadığı durumlarda, sırasıyla önce %5 sonra %10 azaltılması ile elde edilen eşleşmeler gözlenmiştir. Başarım ölçütlerinde ise, kıyaslanma yapılabilmesi amacıyla tüm nesnelere için sabit bir eşik değeri ile sonuçlar hesaplanmıştır. Anahtar nokta eşleştirme aşamasında, benzerlik oranının azaltılması aynı nesne için FP sayılarının artmasına dolayısı ile eşleştirme işleminin başarısının düşmesine sebep olmuştur.

Kullanıcı bağımlı değerlerin değiştirilmesi ile yapılan gözlemlere göre; bu değerlerin belirlenmesi için, nesne tanıma işleminin hangi koşullar altında gerçekleştiği hakkında bir ön bilgi kullanılabilirse, algoritmanın başarısı yükseltilebilir. Önceden bilinmesinde fayda olacak ortam koşullarının başında, aranan nesnenin arama ortamında bulunup bulunmadığı bilgisi gelir. Arama ortamında varlığı önceden bilenen bir nesnenin araştırılmasında, öznitelik vektörleri arasında aranan en küçük fark değeri de nesne yakalana kadar azaltılabilir.

Yapılan değerlendirmeler ayrıca, öznitelik çıkarımında resimlerin detaylı (kenar ve kışelere sahip) bir yapısının bulunması ve bu yapıda kendi içinde bir düzenin var olmasının, doğru anahtar noktaları eşlemede büyük bir avantaj sağladığını göstermiştir. Detaylar açısından yetersiz olan resimlerde, karşılaştırma kriterlerinde esnemeler yapılması ile elde edilen muhtemel nesnenin bulunduğu konumlar üzerinde, belirlenen bir şablon ile benzerlik araması yapıldığında, bu sefer nesnenin doku bilgisinden faydalanılmış ve nesnelerin yakalanması sağlanmıştır.

Geliştirilen B.N.T. uygulaması ile, kullanılan yöntemlerin tüm avantajlarından faydalanarak, tanıma işleminin başarısı, kesinlik ve duyarlılık ölçütlerinin bir arada değerlendirilmesi olan, F-Ölçütüne göre, %95 seviyelerine kadar artırılmıştır.

4.SONUÇLAR

Bir görüntüdeki nesnenin başka bir arama ortamında oryantasyondan bağımsız olarak bulunması için, bölgesel ayırt edici özniteliklerden faydalanılmıştır. Ayırt edici öznitelikler olarak, resim üzerinde belirlenen anahtar nokta tanımlayıcıları kullanılmıştır.

Anahtar noktaların belirlenmesi, anahtar noktalara ait öznitelik vektörlerinin, hem model nesne hem de arama ortamı için çıkarılması, vektörler arasında uyumluluğun en küçük fark hesaplanması ile aranması ve uyumlu vektörlerin ait olduğu anahtar noktaların, model nesne ve arama ortamı resmi arasında karşılıklı olarak eşleştirilmesi yöntemine ait başarı değerlendirmesi sonucu yanlış eşleştirmelerin oranının yüksek olduğu görülmüştür. Algoritmanın başarısı, içinde ayrıntılar bulunan resimlerde yüksekken, ayrıntılar azaldıkça yakalanan eşleşmeler de azalmaktadır. Ayrıca, benzer ayrıntılara sahip, farklı nesnelere ait resim bölgelerinde eşleşmeler algoritmanın başarısını çok düşürmektedir. Dolayısı ile bir doğrulama kriteri kullanımı zorunludur.

Geliştirilen B.N.T. yönteminde bu nokta eşleştirme tekniği kullanılmış fakat eşleştirmelerden yanlış olanların elenmesini sağlayan bir doğrulama işlemi sonucunda yöntemin başarı oranının ne derece iyileştiği gözlenmiştir.

8 ayrı nesnenin, farklı bakış açılarından alınmış fotoğraflarından ve içinde birden çok nesnenin bulunduğu, 51 adet karışık test resimlerinden oluşan veriseti üzerinde yapılan denemeler sonucu, B.N.T. yönteminin, 0.95 başarı oranıyla aranan nesneyi bulduğu görülmüştür. Yöntemin başarısından ödün verilerek sistemin hızı da artırılabilir. Bulunan anahtar nokta sayısının değiştirilmesi ile, doğruluk ve hız arasında, sistemden beklenenlere göre bir denge kurulabilir.

Yang and Song, SIFT ve şablon eşleme yöntemlerini, doğal resimlerden oluşmayan verisetleri üzerinde kullanmış ve tüm eşleme sürecini hızlandırmaya çalışmışlardır [54]. Bulunan anahtar nokta sayısı arttıkça işlem süresinin de artmasını engellemek amacıyla, bir benzerlik sınıflandırmasını ara aşama olarak kullanmışlardır. Sınıflandırma yöntemi olarak K-ortalama kümelemesi (K-means

clustering) yerine, sınıf sayısının bilinmesini gerektirmeyen benzer uzaklık eşiklemesi (Similar Distance Threshold (SDT)) sınıflayıcısından yararlanmışlardır. Şablon konumlandırma aşamasında ise, döngülü en küçük kare yöntemi (iterative Least Square Method (ILSM)) uygulamışlardır. Yöntemde model resim ve arama ortamı arasında bir izdüşüm matrisi oluşturularak konumlandırma gerçekleştirmişlerdir.

Yang and Song daha sonraki çalışmalarında [55], hesaplama süresini bir önceki çalışmalarına [54] göre %20 kısaltmışlar ve kararlılığı da arttırmayı denemişlerdir. Hesaplama süresini SIFT yönteminin en çok zaman alan aşaması üzerinde değişikliğe giderek kısaltmışlardır. Tüm sürecin %30'unu alan uçdeğer hesaplama aşamasında, DoG resimlerden elde edilen uçdeğerlerin, sadece konum, ölçek ve oryantasyon bilgileri kullanılmış, vektör ifadeleri kullanılmamıştır. Eşleşen anahtar noktalar arasında ölçek, öteleme ve oryantasyon farklarını hesaplamışlardır. Şablon konumlandırma için yine, en küçük kare yönteminden faydalanmışlardır.

Yang and Song, çalışmalarında ([54]; [55]) karakter ve simgelerden oluşan resimler kullandıklarından, bu tez çalışmasında kullanılan gerçek resimler ile adil bir karşılaştırmaya sokulamazlar. Ayrıca kullandıkları resimler yakalanması kolay olan, dönüşümlerden fazla etkilenmemiş ve kısmi engellemeye de uğramamış resimlerdir. Bu koşullarda uygun düşen anahtar noktaları yakalamak ve nesnelere konumlandırmak da oldukça kolaydır.

SIFT yöntemini hızlandırmak için başka algoritmalar da geliştirilmiştir. Michael and Helmut [29], tümlev (integral) resim kullanarak SIFT yönteminin, ölçek uzayı oluşturma ve bu uzaydan uçdeğer çıkarımı işlemlerinden oluşan, ilk iki aşamasını hızlandırmaya çalışmışlardır. Geliştirdikleri Hızlı Yaklaşık SIFT (Fast Approximated SIFT) yönteminde, DoG resimler yerine DoM (Difference-of-Mean) resimler kullanmışlardır. Michael and Helmut [29], çalışmalarında aynı resmin aralarında oryantasyon farklılığı olan iki durumu için denemeler yapmışlardır ve bu resimler için bu çalışmada da faydalanılan Lowe [25]'un SIFT yöntemine göre, ortalama altı kat kadar hız elde ettiklerini ileri sürmüşlerdir.

Ke and Sukthankar ise, temel bileşenler analizi (Principal Components Analysis) ile tanımlayıcı boyutunu azaltarak yine SIFT yöntemini hızlandırmaya çalışmışlardır [19]. Çalışmalarında kullandıkları veri seti (INRIA Graffiti dataset) renkli resimlerden oluşmaktadır ve eşleştirme aranan resimlerden her ikisi de farklı nesnelere oluşan karışık ve engellemelerin de bulunduğu doğal ortam resimleridir. Buna rağmen, eşleştirme aranan ortamların karmaşıklığı ve engellemelerin varlığı nedeniyle bu tez çalışmasının öznel eşleştirme aşaması ile karşılaştırılabilir. 1.60GHz Intel Core i7 bilgisayarda elde edilen verilere göre, Ke and Sukthankar [19]'ın çalışmasında verilen karşılaştırma kriterleri değerlendirildiğinde, anahtar nokta belirleme işleminde SIFT, PCA-SIFT yönteminden %5 daha hızlı fakat bu noktalar üzerinden eşleştirme yapma aşamasında iki kat daha yavaştır.

Şablon eşleme yönteminin, şablon resim ve arama ortamı arasında pikseller arası gri seviye ve ilişiklik araştırmasında, şablonun arama ortamı üzerinde kaydırılması ve benzerlik derecesinin hesaplanması, basit fakat zaman alan bir işlemdir. Bu sürecin hızlandırılması için de yapılan bazı çalışmalar vardır. Gharavi and Alkhansari [10], J.P.Lewis [22], Schweitzer and Bell [48], hızlı şablon eşleme yöntemleri geliştirmişlerdir.

Bu tez çalışmasında, şablon resim, tüm arama ortamı üzerinde kaydırılmamış, sadece SIFT ile belirlenen birkaç anahtar nokta etrafındaki alanlar üzerinde gezdirilmiştir. Bu nedenle şablon eşlemenin dezavantajı olan işlemin uzun sürmesi, bu çalışma için geçerli bir durum değildir. Bu çalışmadaki şablon eşleme süresi, tamamen eşleşen anahtar nokta sayısı ile orantılıdır. Dolayısı ile yukarıda bahsedilen hızlı şablon eşleme yöntemleri ile karşılaştırmak doğru olmayacaktır.

Aoyagi and Hattori [1], iç mekanda bir robotun hareketi için, yol belirleme amacıyla, hareket alanını, tüm alanı daraltarak belirleyen bir yöntem geliştirmek amacıyla, SIFT ve şablon eşleme yöntemlerinden faydalanmışlardır. Bölgesel şablonlar ve zeminin özelliklerini kullanarak, mekan tanıma yapmışlardır. Sınıflandırma ile gidilebilecek yolu mekanın kalanından ayırmışlardır.

Görüldüğü üzere kullanılan yöntemler, başarı oranları yüksek olan geliştirilmeye uygun, değişkenlerin kontrolü ile isteğe cevap verebilecek ve farklı alanlarda kullanılabilir yöntemlerdir.

4.1 Gelecek Çalışmalar Planı

Bu tez çalışması sonucu elde edilen bulgulara ek olarak, aşağıdaki konuların da ele alınması, gelecekte konuyla ilgili yapılacak çalışmalarda yararlı olacaktır.

- Benzerlik aramaları için kullanılan nokta etrafındaki bölge alanının değişmesi ile çalışmadaki başarının nasıl değiştiği araştırılabilir.
- Eşleşme yakalanan noktalar etrafında, daha kararlı bölgeler seçilmesi ile şablon eşlemenin işlem maliyeti azaltılabilir.
- Yapılan farklı benzerlik aramaları ile bunlar arasında da bir performans değerlendirmesi yapılabilir.
- Nesnenin bulunduğu konum belirlendikten sonra snake algoritması gibi bir yöntem, bölütleme (segmentation) amacıyla kullanılabilir. Böylece yakalanan nesnenin tamamı da belirginleştirilebilir.
- Arama ortamı ile ilgili bazı ön bilgiler kullanarak yöntemin hızı artırılabilir.
- Yakalanan nesne üzerinde, nesneye ait genel özellikler, yani global öznitelikler kullanılabilir.
- Öznitelik vektör çıkarımı aşamasında kullanılan öznitelikler çeşitlendirilerek farklı sonuçlar elde edilebilir.
- Çıkarılan bir öznitelik aracılığıyla nesne sınıfının belirlenmesi ile çalışmaya başlayan bir algoritma geliştirilerek, ilgili nesneye ait farklı bakış açılarından çekilmiş diğer resimler için de öznitelik çıkarımlarının yapılmasıyla, elde edilen öznitelik vektör kümesi bir arama ortamında birlikte araştırılabilir.
- Farklı verisetleri üzerinde daha çok denemeler yapılabilir.

KAYNAKLAR LİSTESİ

- [1] Aoyagi, S., Hattori, N., Kohama, A., Komai, S., Suzuki, M., Takano, M., and Fukui, E., Object Detection and Recognition Using Template Matching with SIFT Features Assisted by Invisible Floor Marks, *Journal of Robotics and Mechatronics (JRM)*, Vol.21, No.6, 2009.
- [2] Baumberg, A., Reliable feature matching across widely separated views, In *Conference on Computer Vision and Pattern Recognition (CCVPR)*, Hilton Head, South Carolina, pp.774-781, 2000.
- [3] Basri, R., and Jacobs, D.W., Recognition using region correspondences, *International Journal of Computer Vision (IJCV)*, 25(2),145-166, 1997.
- [4] Bicego M., Lagorio A., Grosso E., and Tistarelli M., On the use of SIFT features for face authentication, In *IEEE International Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pp.35-35, June 2006.
- [5] Brown, M. and Lowe, D.G., Invariant features from interest point groups, In *British Machine Vision Conference (BMVC)*, Cardiff, Wales, pp.656-665, 2002.
- [6] Carneiro, G., and Jepson, A.D., Phase-based local features, In *European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, pp.282-296, 2002.
- [7] Cesar Costa, *Shape analysis and classification: theory and practice*, Boca Raton, 2001.
- [8] Cheung, W., Hamarneh G., n-SIFT: n-dimensional scale invariant feature transform, *IEEE Transactions on Image Processing (TIP)*, Vol.18, No.9, pp.2012-2021, September 2009.
- [9] Crowley, J. L. and Parker, A.C., A representation for shape based on peaks and ridges in the difference of low-pass transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6(2), pp.156-170, 1984.
- [10] Gharavi-Alkhansari, M., A fast globally optimal algorithm for template matching using low-resolution pruning, *IEEE Transactions on Image Processing (TIP)*, Vol.10, pp.526-33, 2001.
- [11] Goedemé, T., Tuytelaars, T., Vanacker, G., Nuttin, M. and Van Gool, L., Feature based omnidirectional sparse visual path following, In *IEEE/RSJ International Conference on Intelligent Robots and Systems (ICIRS)*, Edmonton, pp.1003–1008, 2005.
- [12] Gonzalez and Woods, *Digital Image Processing*, 3rd Ed., Prentice-Hall, 2008.
- [13] Goshtasby Ardeshir, *2-D and 3-D image registration for medical, remote sensing, and industrial applications*, Wiley-Interscience, 2005.

- [14] Graves, M., and Batchelor, B., Machine Vision for the Inspection of Natural Products, Springer Verlag, 2004.
- [15] Harris, C. and Stephens, M., A combined corner and edge detector, In Fourth Alvey Vision Conference (AVC), Manchester, UK, pp.147-151, 1988.
- [16] Harris, C., Geometry from visual motion. In Active Vision, A. Blake and A. Yuille, Eds., MIT Press, pp.263-284, 1992.
- [17] Jain Anil K., Fundamentals of digital image processing, Prentice Hall, 1989.
- [18] Jähne Bernd, Digital Image Processing, 6th Ed., Springer, 2005.
- [19] Ke, Y., and Sukthankar, R., Pca-sift: a more distinctive representation for local image descriptors. In Computer Vision and Pattern Recognition (CVPR), Proceedings of the IEEE Computer Society Conference, Vol.2, pp.506-513, 2004.
- [20] Klette Reinhard and Azriel Rosenfeld, Digital geometry: geometric methods for digital picture analysis, Elsevier, 2004.
- [21] Ledwich, L., and Williams, S., Reduced SIFT features for image retrieval and indoor localisation, In Australian Conference on Robotics and Automation (ACRA), 2004.
- [22] Lewis, J. P., Fast template matching, Vision Interface, pp.120–123, 1995.
- [23] Lindeberg, T., Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention, International Journal of Computer Vision (IJCV), 11(3), pp.283-318, 1993.
- [24] Lindeberg, T., Scale-space theory: A basic tool for analysing structures at different scales, Journal of Applied Statistics, 21(2), pp.224-270, 1994.
- [25] Lowe, David G., Object Recognition from Local Scale-Invariant Features, In International Conference on Computer Vision (ICCV), Corfu, Greece, pp.1150-1157, 1999.
- [26] Lowe, David G., Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision (IJCV), 2004.
- [27] Ma Yi et al., An invitation to 3-D vision: from images to geometric models, Springer Verlag, 2004.
- [28] Matas, J., Chum, O., Urban, M., and Pajdla, T., Robust wide baseline stereo from maximally stable extremal regions, In British Machine Vision Conference (BMVC), Cardiff, Wales, pp.384-393, 2002.
- [29] Michael, G., Helmut, G., and Horst, B., Fast approximated sift. Asian Conference on Computer Vision (ACCV), Hyderabad, India, pp.918–927, 2006.

- [30] Mikolajczyk, K., Detection of local features invariant to affine transformations, Ph.D. thesis, Institut National Polytechnique de Grenoble, France, 2002.
- [31] Mikolajczyk, K., Zisserman, A., and Schmid, C. Shape recognition with edge-based features, In Proceedings of the British Machine Vision Conference (BMVC), Norwich, U.K, 2003.
- [32] Mikolajczyk, K., and Schmid, C., A performance evaluation of local descriptors, In Proceedings of Computer Vision and Pattern Recognition (CVPR), June 2003.
- [33] Mingqiang, Y., Kidiyo K., and Joseph R., A survey of shape feature extraction techniques, in Pattern Recognition, chapter 3, IN-TECH, Vienna, Austria, 2008.
- [34] Moeslund, T. B., Hilton A. , Krüger V., A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding (CVIU), Vol.104, No.2, pp.90-126, November 2006.
- [35] Moravec, H., Rover visual obstacle avoidance, In International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, Canada, pp.785-790, 1981.
- [36] Murase, Hiroshi, and Shree K. Nayar, Visual learning and recognition of 3-D objects from appearance, International Journal of Computer Vision (IJCV), 14, 1, pp.5–24, 1995.
- [37] Nixon Mark S. and Aguado Alberto S., Feature Extraction and Image Processing, Newnes, 2002.
- [38] Nelson, R.C., and Selinger, A., Large-scale tests of a keyed, appearance-based 3-D object recognition system, Vision Research, 38(15), pp.2469-88, 1998.
- [39] Ohba, Kohtaro, and Katsushi Ikeuchi, Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 19, 9, pp.1043–48, 1997.
- [40] Pope, A.R., and Lowe, D.G., Probabilistic models of appearance for 3-D object recognition, International Journal of Computer Vision (IJCV), 40(2), pp.149-167, 2000.
- [41] Pratt William K., Digital image processing, Wiley-Interscience, 2001.
- [42] Rothganger F., Lazebnik S., Schmid C., and Ponce J., 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints, International Journal of Computer Vision (IJCV), Vol. 66, No.3, pp.231-259, March 2006.
- [43] Russ John C., The image processing handbook, CRC Press, 2002.

- [44] Schaffalitzky, F., and Zisserman, A., Multi-view matching for unordered image sets, or ‘How do I organize my holiday snaps?’” In European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp.414-431, 2002.
- [45] Schiele, B., and Crowley, J.L., Object recognition using multidimensional receptive field histograms, Fourth European Conference on Computer Vision (ECCV), Cambridge, UK, pp.610–619, 1996.
- [46] Schiele, B., and Crowley, J.L., Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision (IJCV)*, 36(1), pp.31-50, 2000.
- [47] Schmid, C., and R. Mohr, Local grayvalue invariants for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19, 5, pp.530–534, 1997.
- [48] Schweitzer, H., Bell, J. W., and Wu, F., Very fast template matching. *Proceedings of the 7th European Conference on Computer Vision(ECCV)*, Part IV, pp.358-372, 2002.
- [49] Shi, J., and Tomasi, C., Good features to track. TR 93-1399, Cornell U., 1993.
- [50] Shokoufandeh, A., Marsic, I., and Dickinson, S.J., View-based object recognition using saliency maps, *Image and Vision Computing*, 17, pp.445-460, 1999.
- [51] Swain, M., and D. Ballard, Color indexing, *International Journal of Computer Vision (IJCV)*, 7, 1, pp.11–32, 1991.
- [52] Torr, P., Motion Segmentation and Outlier Detection, Ph.D. Thesis, Dept. of Engineering Science, University of Oxford, UK, 1995.
- [53] Tuytelaars, T., and Van Gool, L., Wide baseline stereo based on local, affinity invariant regions, In *British Machine Vision Conference (BMVC)*, Bristol, UK, pp.412-422, 2000.
- [54] Yang, Y., Song, Y., Shaikh, M. A., and Wang, J., A high-precision template localization algorithm using SIFT keypoints, *International Symposium on Computer and Information Sciences (ISCIS)*, pp.1-6, 2008.
- [55] Yang, Y., Song, Y., Fan, Z., Meng, Y. and Wang, J., A high-precision localization algorithm by improved SIFT key-points, *International Congress on Image and Signal Processing (CISP)*, pp.1–6, 2009.
- [56] Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.T., A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artificial Intelligence*, 78, pp.87-119, 1995.
- [57] Zhou, H., Yuan, Y., Shi, C., Object tracking using SIFT features and mean shift, *Computer Vision and Image Understanding (CVIU)*, Vol.113, No.3, pp.345-352, March 2009.







- [58] <http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/>, Aġustos 2011.
- [59] http://en.wikipedia.org/wiki/Computer_vision, Aġustos 2011.
- [60] [http://en.wikipedia.org/wiki/Object_recognition_\(computer_vision\)](http://en.wikipedia.org/wiki/Object_recognition_(computer_vision)), Aġustos 2011.
- [61] <http://www.quantdec.com/GIS/affine.htm>, Aġustos 2011.
- [62] http://en.wikipedia.org/wiki/Precision_and_recall, Aġustos 2011.

EKLER

EK 1. İlgin Dönüşümler

İlgin dönüşümler, bir koordinat nokta setini, ötelemek, ölçeklemek, döndürmek veya eğme amacıyla kullanılırlar [61].

	Genel matris eşitlik ifadesi	Matris değerleri
İki boyutlu bir ilgin dönüşüm ifadesi.	$\begin{bmatrix} x \\ y \end{bmatrix} = A \times \begin{bmatrix} v \\ w \end{bmatrix} + B$	$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ $B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$
Üç boyutlu bir ilgin dönüşüm ifadesi	$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} \times T$	$T = \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$

		2B İlgin Dön. A matrisi	3B İlgin Dön. T matrisi
Birim (Identity)		$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Öteleme (Translate)		$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ B matrisi öteleme miktarını belirler	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$
Ölçekleme (Scale)		$\begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}$	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Döndürme (Rotate)		$\begin{bmatrix} \cos q & -\sin q \\ \sin q & \cos q \end{bmatrix}$	$\begin{bmatrix} \cos q & \sin q & 0 \\ -\sin q & \cos q & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Düşey eğme (Sheer)			$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Yatay eğme (Sheer)			$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Şekil ek.1 İlgin dönüşüm matrisleri ve etkileri

Bu matrislerin bir arada kullanılması ile, dönüşümlerin birlikte yapılması sağlanabilir.

EK 2. DoG ve LoG Fonksiyonlarının Benzerliđi

DoG ifadesi:

$$D(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma) \quad (\text{ek.1})$$

LoG ifadesi:

$$LoG = k' \cdot \sigma^2 \nabla^2 G \quad (\text{ek.2})$$

Ölçeđe göre normalize edilmiř LoG ifadesi:

$$LoG = \sigma^2 \nabla^2 G \quad (\text{ek.3})$$

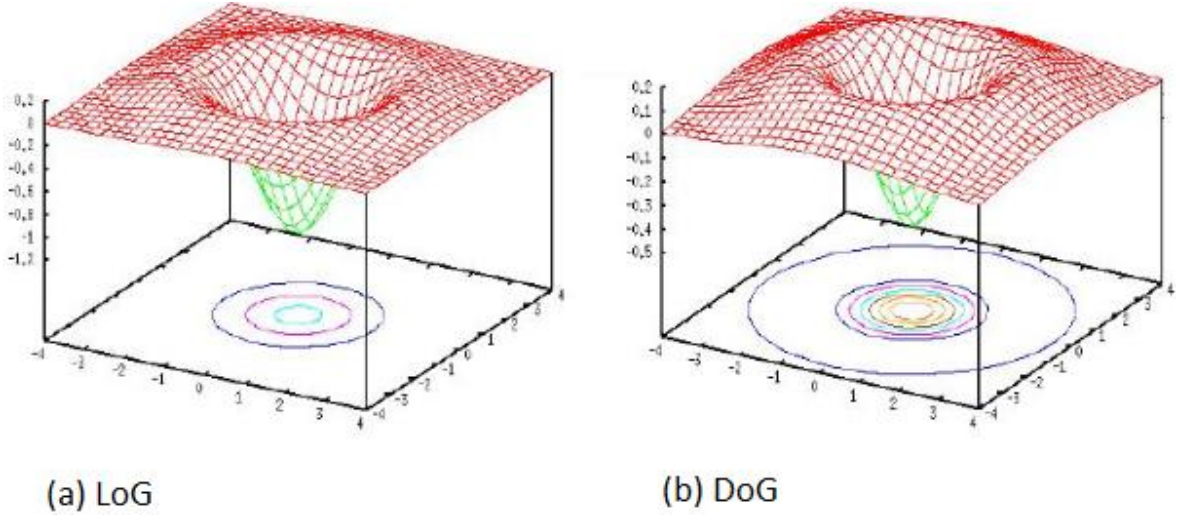
$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (\text{ek.4})$$

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (\text{ek.5})$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1) \sigma^2 \nabla^2 G \quad (\text{ek.6})$$

$(k - 1)$ tüm ölçekleri etkileyen bir sabit olduđundan, uçdeđerlerin konumlarını deđiřtirmeyecektir. Ayrıca DoG ifadesi, ölçekten bađımsız LoG için σ^2 ile normalize edilme gerekliliđinden de halihazırda bađımsızdır. Bu da fazladan bir avantajdır.

DoG ve LoG fonksiyonlarının bir resim üzerinde benzer sonuçlar vereceđi, iki fonksiyonun benzerliđinin Matlab ortamında gsterilmesi ile de anlařılmaktadır (řekil ek.2).

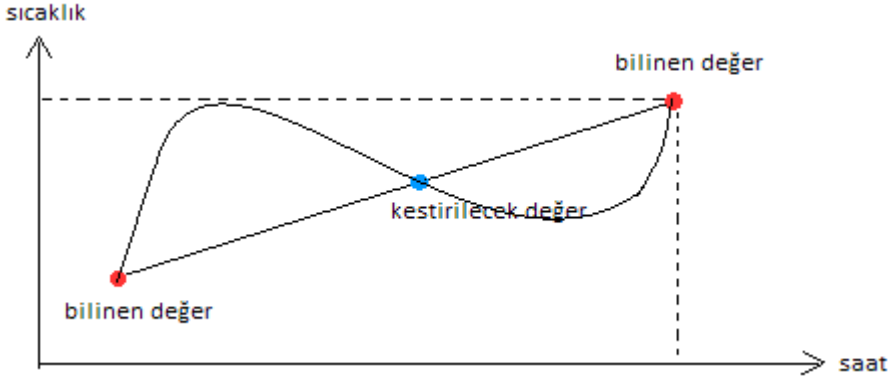


řekil ek.2 LoG ve DoG fonksiyonlarının Matlab ortamında benzerliđinin gsterilmesi

EK 3. Ara Değer Kestirimi

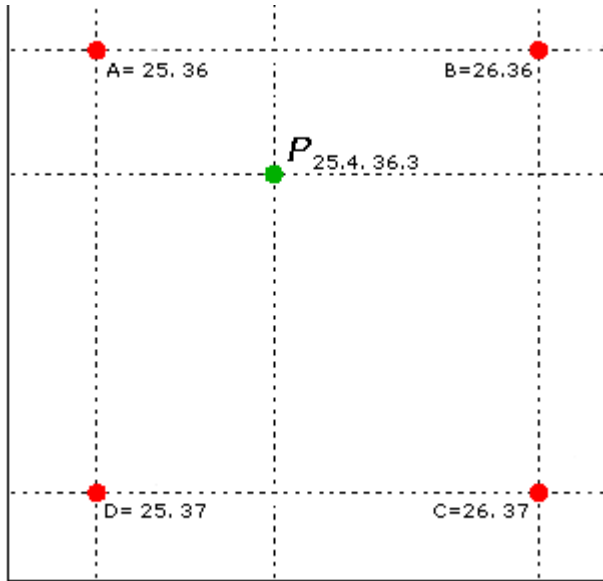
Bilinen değerleri kullanarak bilinmeyen değerleri elde etme işlemine ara değer kestirimi denilir.

Doğrusal (linear) bir ara değer kestirimi için Şekil ek.3'deki gibi bir grafik kullanılabilir.



Şekil ek.3 Doğrusal aradeğer kestirimi

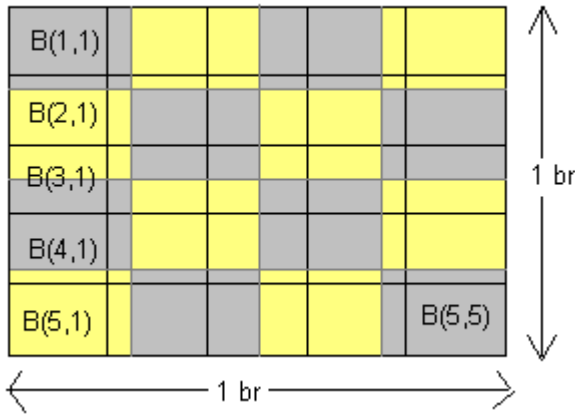
Çift doğrusal (bilinear) kestirim, doğrusal kestirimin iki değişkenli fonksiyonlar için genişletilmiş halidir. Buradaki ana fikir linear interpolasyonun önce bir yönde daha sonra diğer yönde yapılmasıdır. Resim üzerinde çift doğrusal kestirim yapılırken, ilk önce x yönünde, daha sonra da y yönünde doğrusal kestirim yapılır (Şekil ek.4).



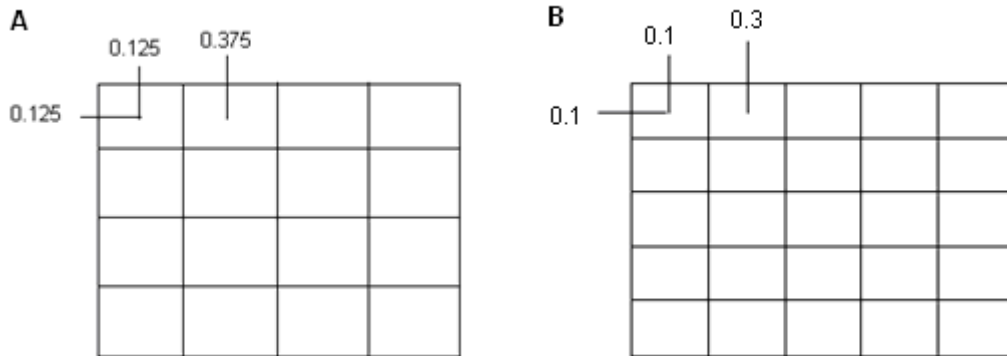
Şekil ek.4 Çift doğrusal aradeğer kestirimi

Buradaki P noktası hesaplamak için 3 defa doğrusal interpolasyon yapılır. Son kestirim, gri düzeyin belirlenmesi içindir.

Bir resmin yeniden ölçeklenmesi yapıldığında, ara değer kestirimleri yapılmalıdır. Örneğin; Şekil ek.5'de görüldüğü gibi 4x4 ölçülerinde bir resmi, 5x5 ölçülerine yeniden ölçeklediğimizi düşünelim. 4x4'lük resim A matrisi olarak, 5x5'lik resim ise, B matrisi olarak alınırsa; 5x5'lik bir ızgara A'nın üzerine oturtulmuş gibi düşünülmelidir. Her iki matrisi uzayda 1 br ebatlarına sahip ve bir köşeleri orjinde olarak düşünürsek;



Şekil ek.5 Resmin yeniden ölçeklenmesi sonucu oluşan ara değerler



Şekil ek.6 Resim pikselleri üzerinde ara değerlerin kestirilmesi

A(1,1)'deki pixelin değeri 1 dir ve bunun uzaydaki koordinatı 0.125, 0.125 br dir.

A(1,2)'deki pixelin değeri 2 dir ve bunun uzaydaki koordinatı 0,375, 0.125 br dir.

B(1,1)'in değerini bulmak istiyoruz ve bunun uzaydaki koordinatı 0.1, 0.1 br dir.

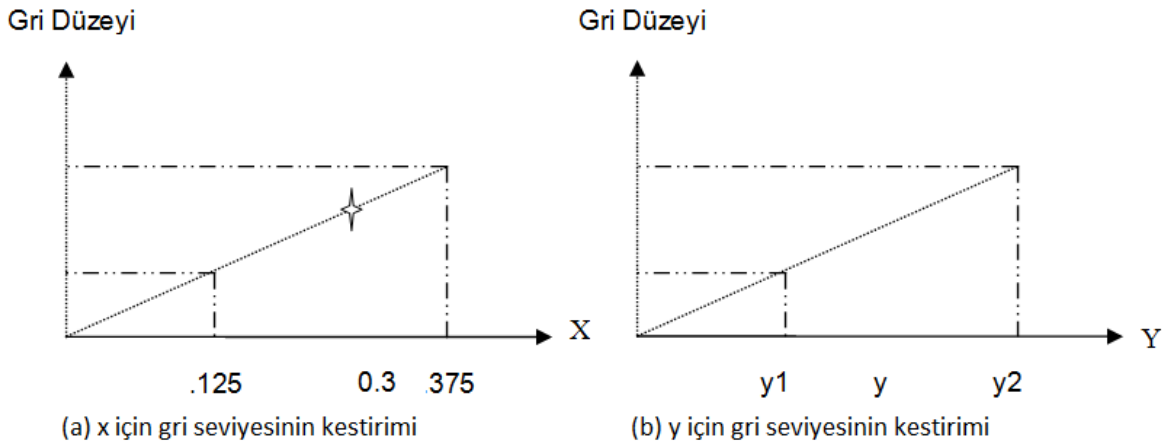
B(1,2)'in deęerini bulmak istiyoruz ve bunun uzaydaki koordinatı 0.3, 0.1 br dir.

.....

.....

Yapmak istediđimiz B(1,2) deki 0.3, 0.1 br deęerlerini kullanarak, B(1,2)'nin gri d¼zeyine x ve y yön¼ndeki etkiyi hesaplamak.

X için, A(1,1) ve A(1,2) nin x deęerleri 0.125 ve 0.375 kullanılarak bir doęru oluřturulur. Aynı iřlem benzer řekilde y için de yapılmalıdır.



řekil ek.7 x ve y için gri deęer kestirilmesi

EK 4. Pikseller Arası Komşuluk Araştırılması

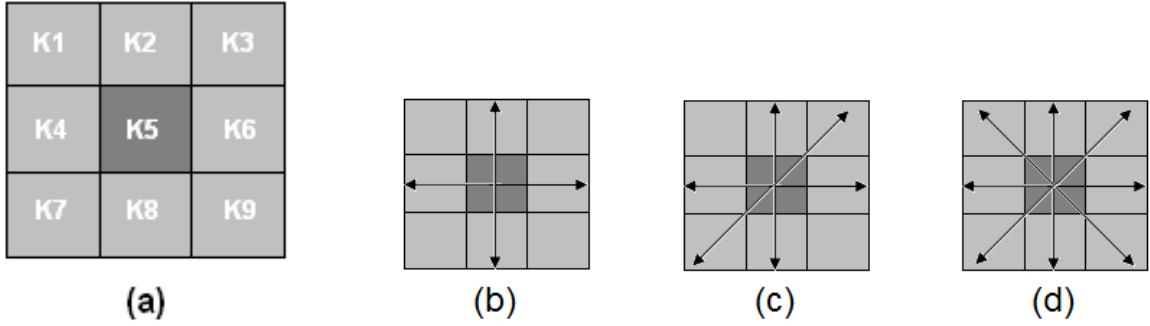
İki Boyutlu Komşuluk:

(x, y) koordinatında bulunan K5 pikseli için düşünüldüğünde (Şekil ek.8 (a)),

4-komşusu, ($x\pm 1, y$) ve ($x, y\pm 1$) koordinatlarında bulunan; K2, K4, K6 ve K8 (Şekil ek.8 (b))

6-komşusu, 4-komşulara ilave gelen ($x+1, y+1$) ve ($x-1, y-1$) koordinatlarında bulunan; K3 ve K7 (Şekil ek.8 (c))

8-komşusu, 4-komşulara ilave gelen ($x\pm 1, y\pm 1$) ve ($x\pm 1, y\mp 1$) koordinatlarında bulunan; K1, K3, K5, K7, ve K9 (Şekil ek.8 (d))

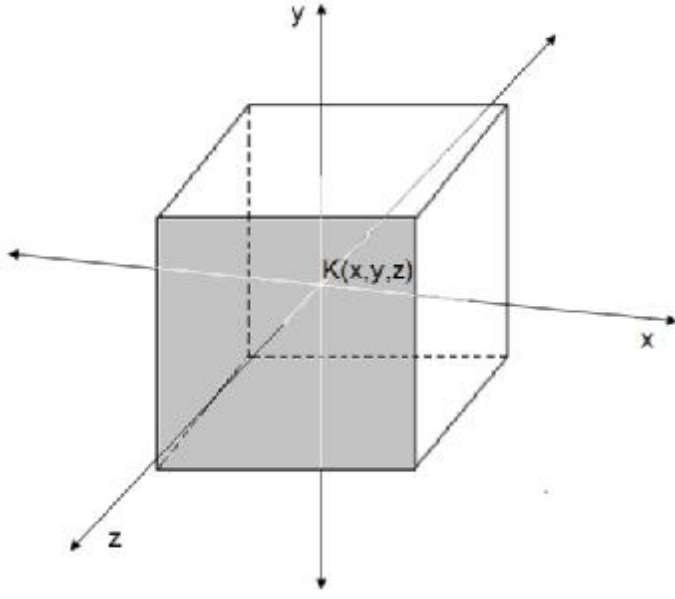


Şekil ek.8 K5 pikseline ait 4 ve 8 komşuluklar

Üç Boyutlu Komşuluk:

(x, y, z) koordinatında bulunan K pikseli için düşünüldüğünde (Şekil ek.9),

- 6-komşusu, ($x\pm 1, y, z$), ($x, y\pm 1, z$) ve ($x, y, z\pm 1$) koordinatlarında bulunan
- 18-komşusu, 6-komşulara ilave gelen ($x\pm 1, y\pm 1, z$), ($x\pm 1, y\mp 1, z$), ($x\pm 1, y, z\pm 1$), ($x\pm 1, y, z\mp 1$), ($x, y\pm 1, z\pm 1$), ve ($x, y\pm 1, z\mp 1$) koordinatlarında bulunan
- 26-komşusu, 18-komşulara ilave gelen ($x\pm 1, y\pm 1, z\pm 1$), ($x\pm 1, y\pm 1, z\mp 1$), ($x\pm 1, y\mp 1, z\pm 1$), ($x\pm 1, y\mp 1, z\mp 1$), ($x\mp 1, y\pm 1, z\pm 1$), ($x\mp 1, y\pm 1, z\mp 1$), ($x\mp 1, y\mp 1, z\pm 1$) ve ($x\mp 1, y\mp 1, z\mp 1$) koordinatlarında bulunan piksellerdir.

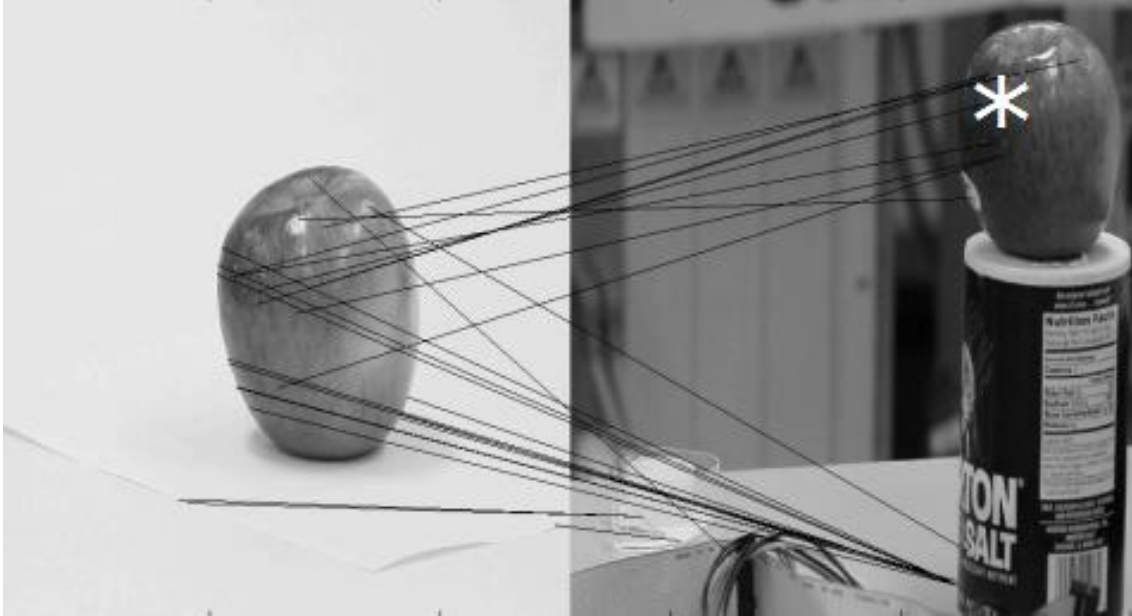


Şekil ek.9 K pikselinin konumu

EK 5. B.N.T. Yöntemine İlişkin Diğer Çıktı Örnekleri

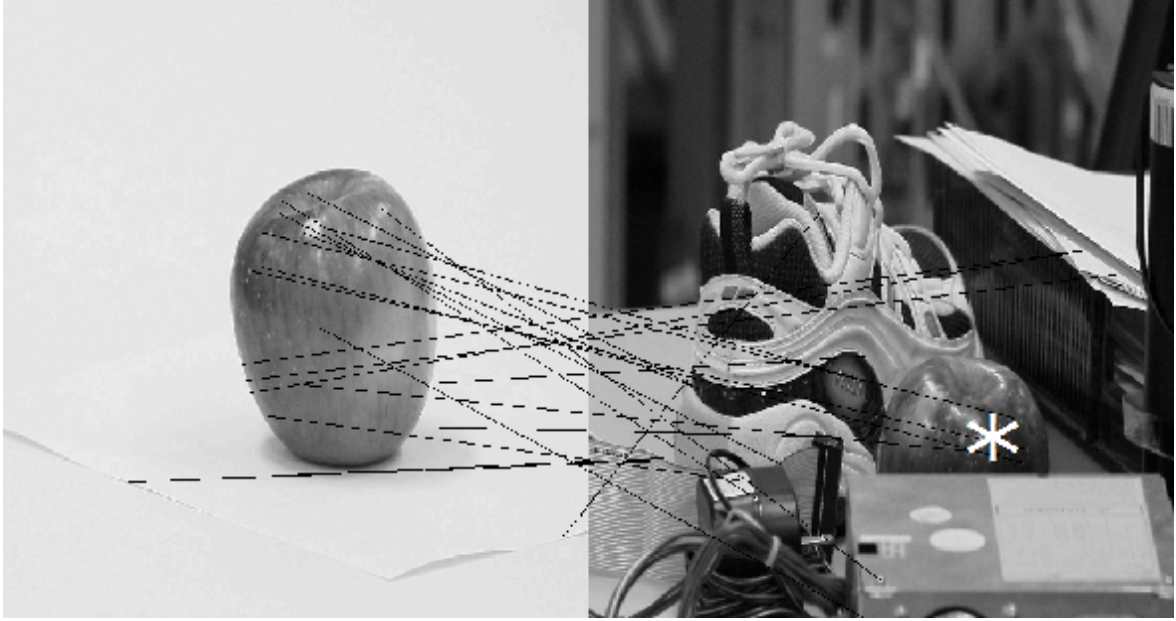
Verilen görsellerde (Şekil ek.10 - Şekil ek.75) B.N.T. algoritmasının tüm veriseti resimlerine uygulanması ile elde edilen sonuçlar gösterilmiştir. Sonuç resimlerinde anahtar noktaların eşleştirilmesi ve tespit edilen nesnenin bu görsel üzerinde işaretlenmesi yapılmıştır.

Herbir resim altında, toplam 8 farklı nesne ve 51 farklı arama ortamından oluşan verisetinin hangi elemanlarının kullanıldığı, model nesne (train) ve arama ortamı (test) ayrımı esas alınarak, verisetindeki ilgili numaraları ile belirtilmiştir.

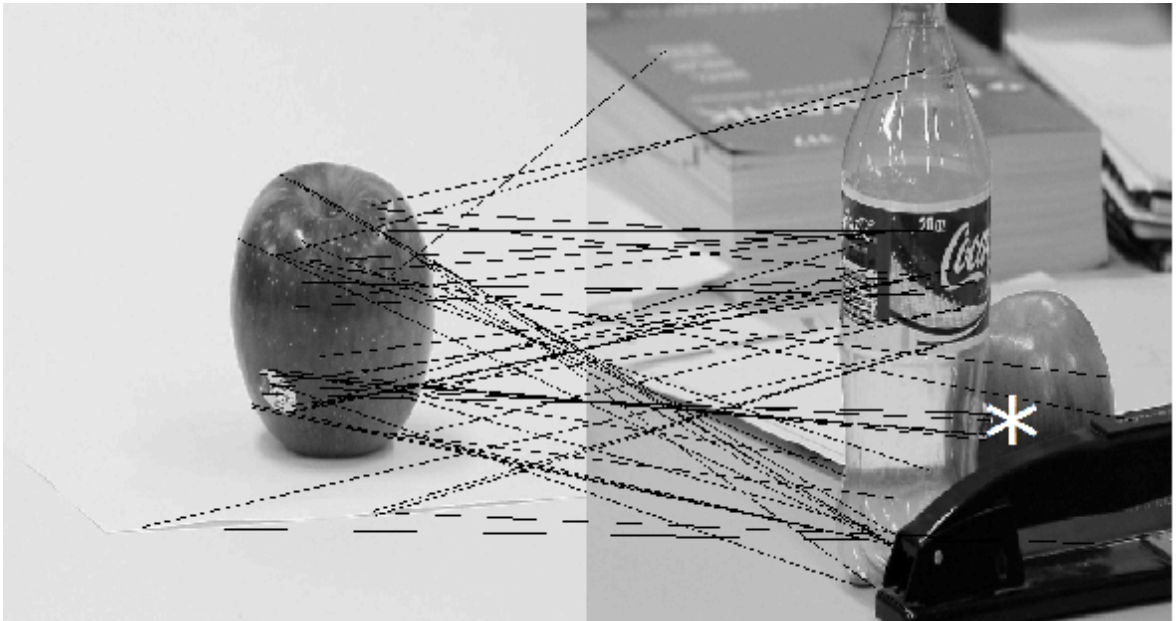


Şekil ek.10 Obj1-Train3 Test8

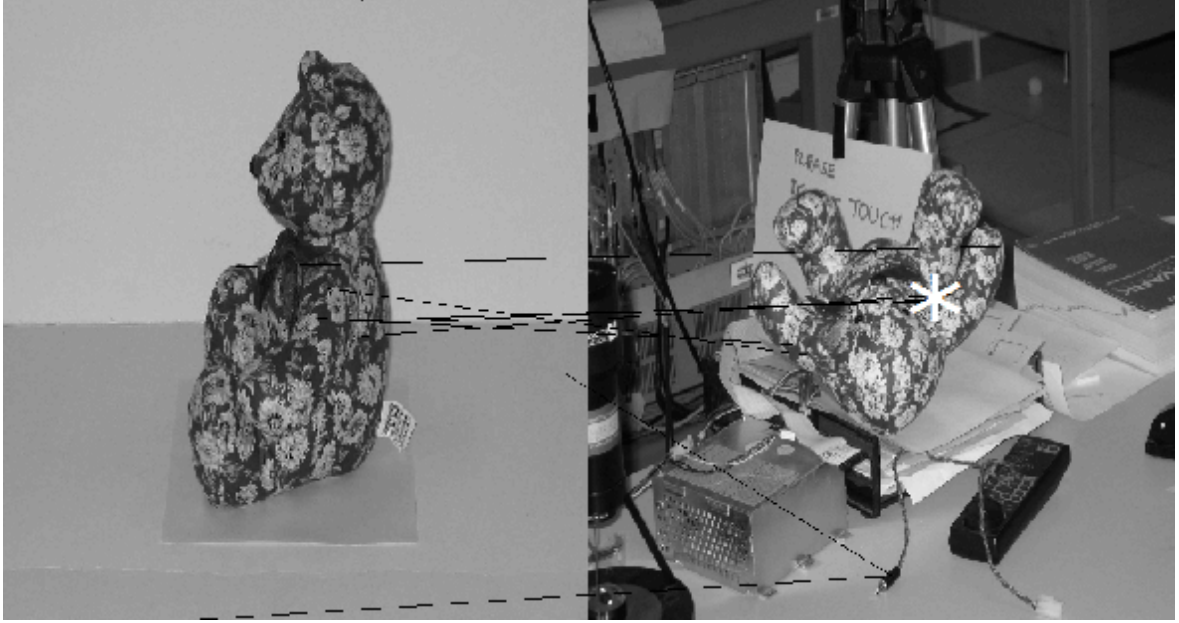
Şekil ek.10'da, birinci nesneye (Obj1) ait üçüncü eğitim resmi (Train3), sekiz numaralı test ortamında (Test8) araştırılmaktadır. Bu tanımlama yöntemi sonraki şekiller için de aynen uygulanmıştır.



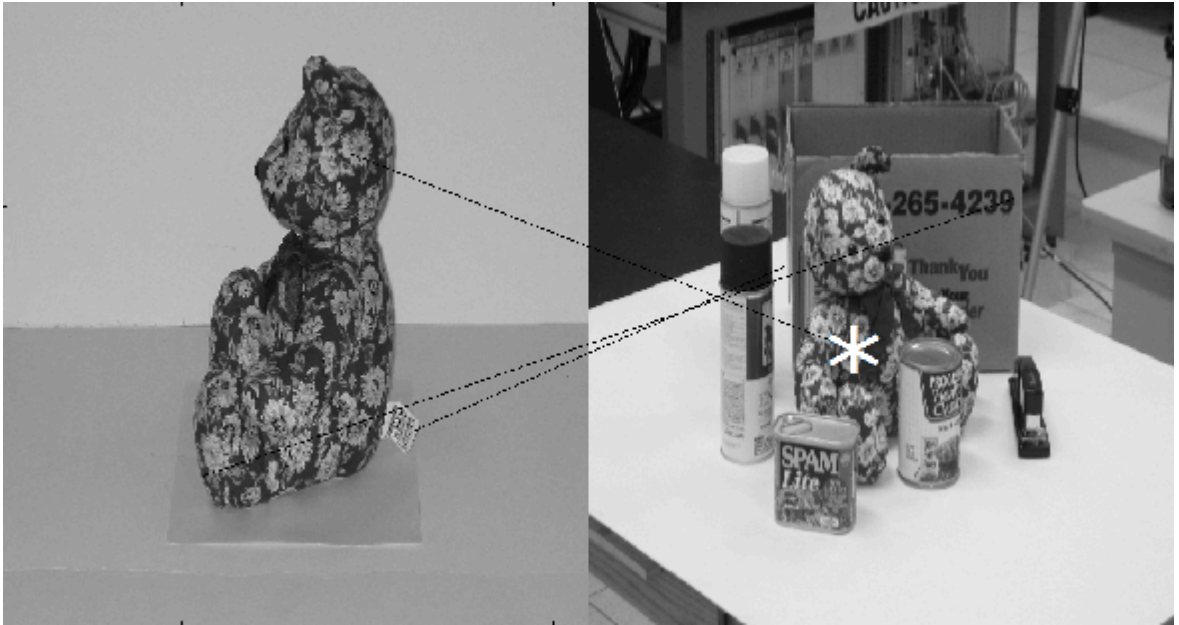
Şekil ek.11 Obj1-Train3 Test9



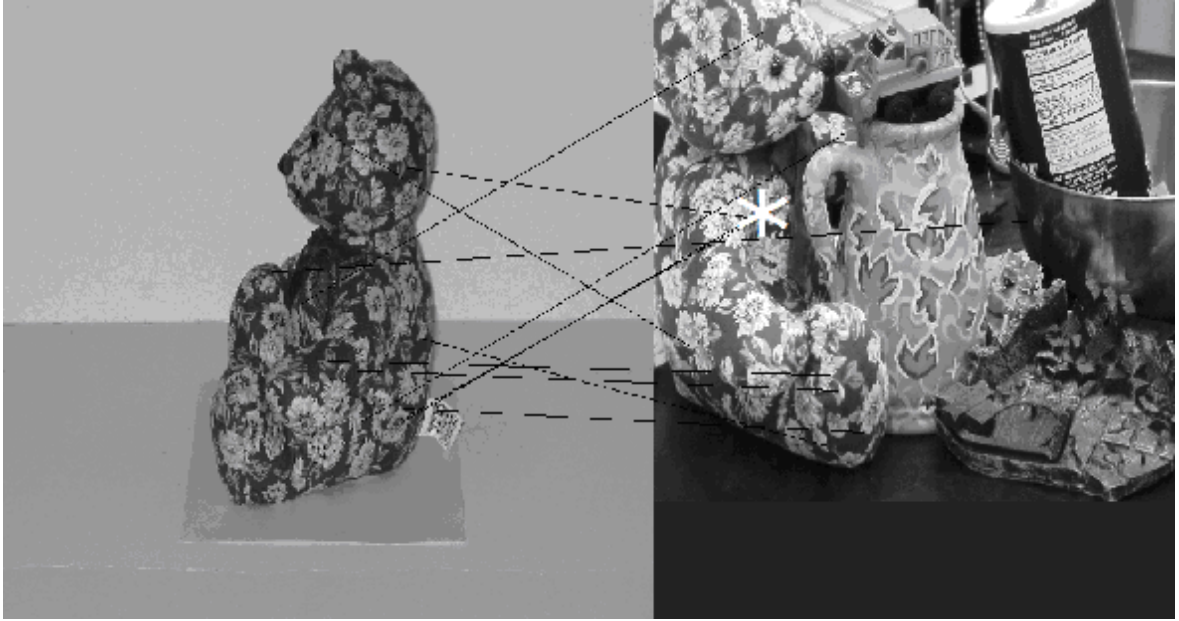
Şekil ek.12 Obj1-Train6 Test3



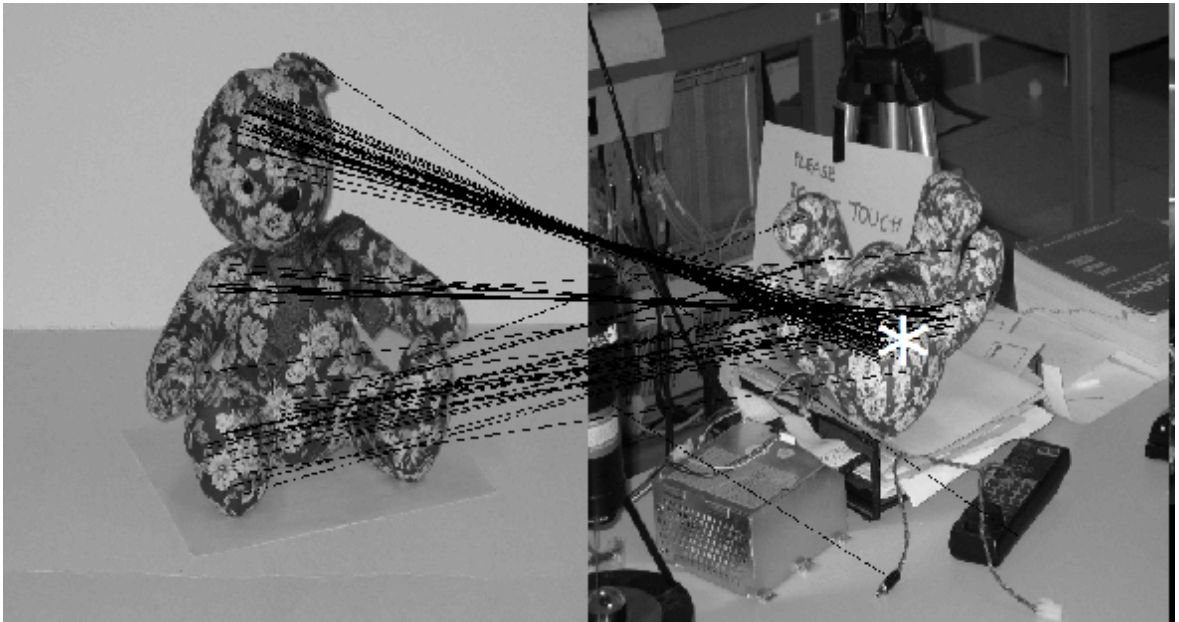
Şekil ek.13 Obj2-Train5 Test13



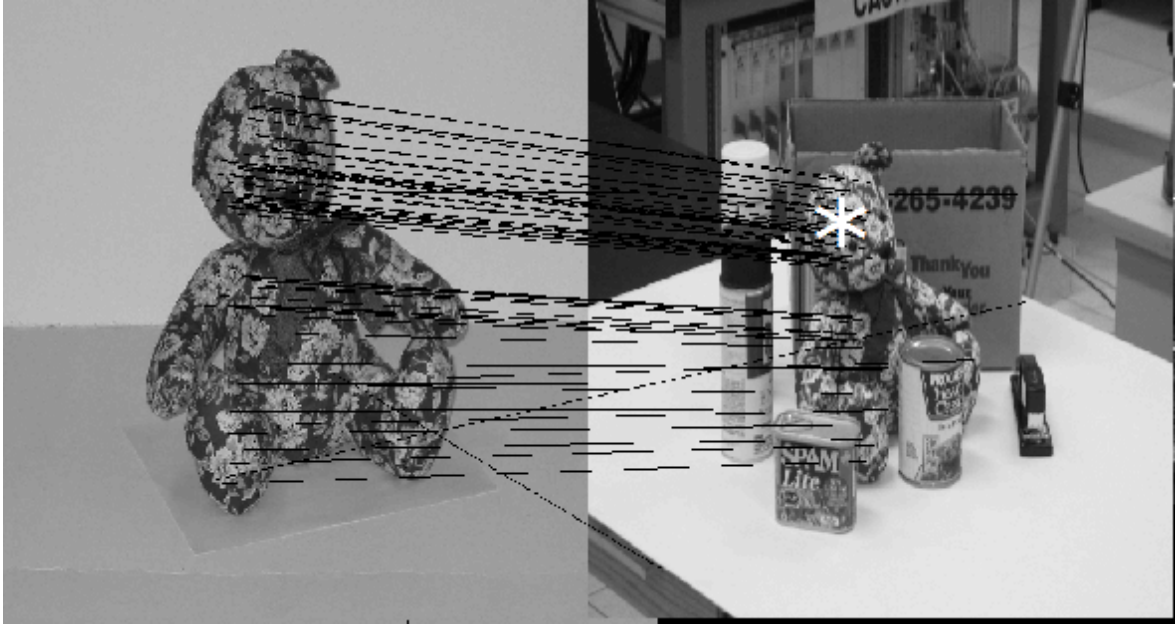
Şekil ek.14 Obj2-Train5 Test16



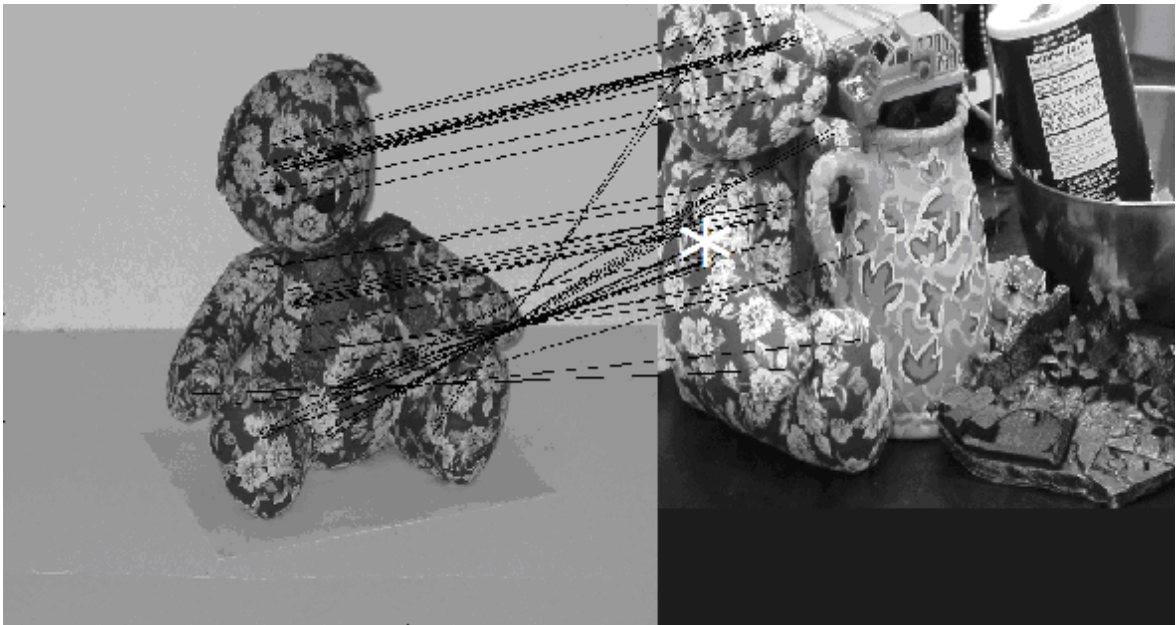
Şekil ek.15 Obj2-Train5 Test20



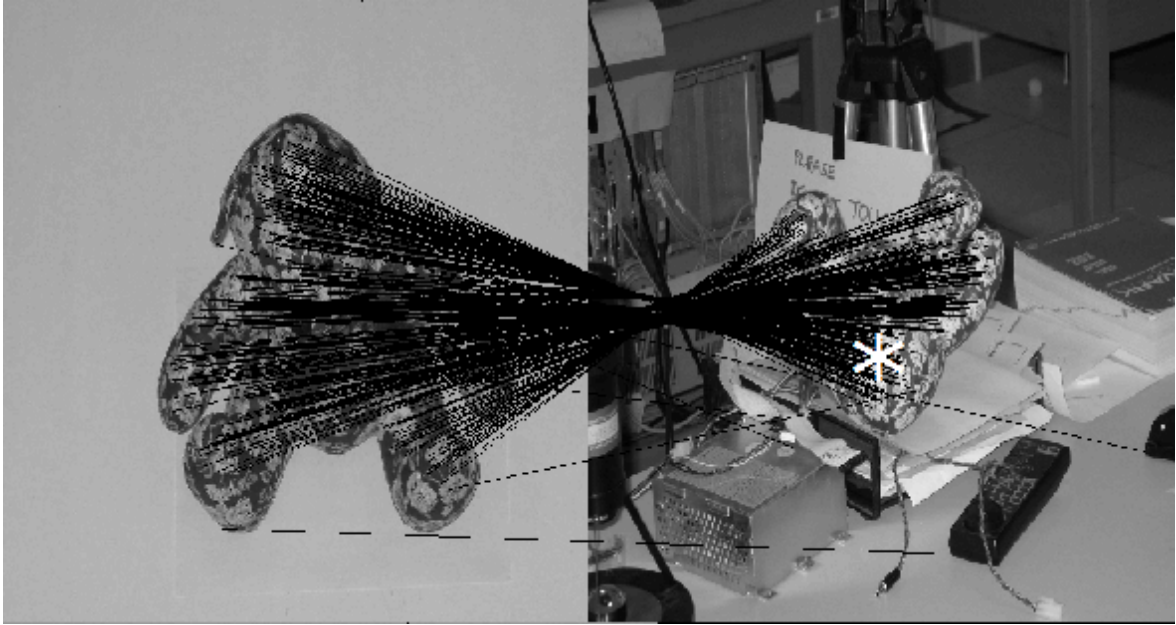
Şekil ek.16 Obj2-Train16 Test13



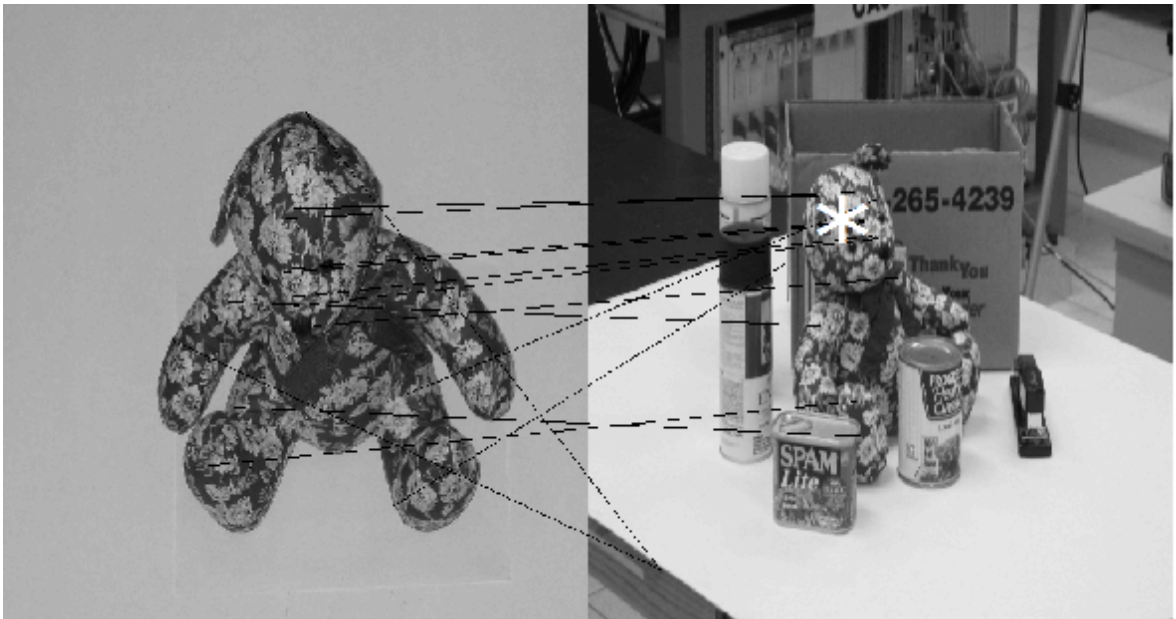
Şekil ek.17 Obj2-Train16 Test16



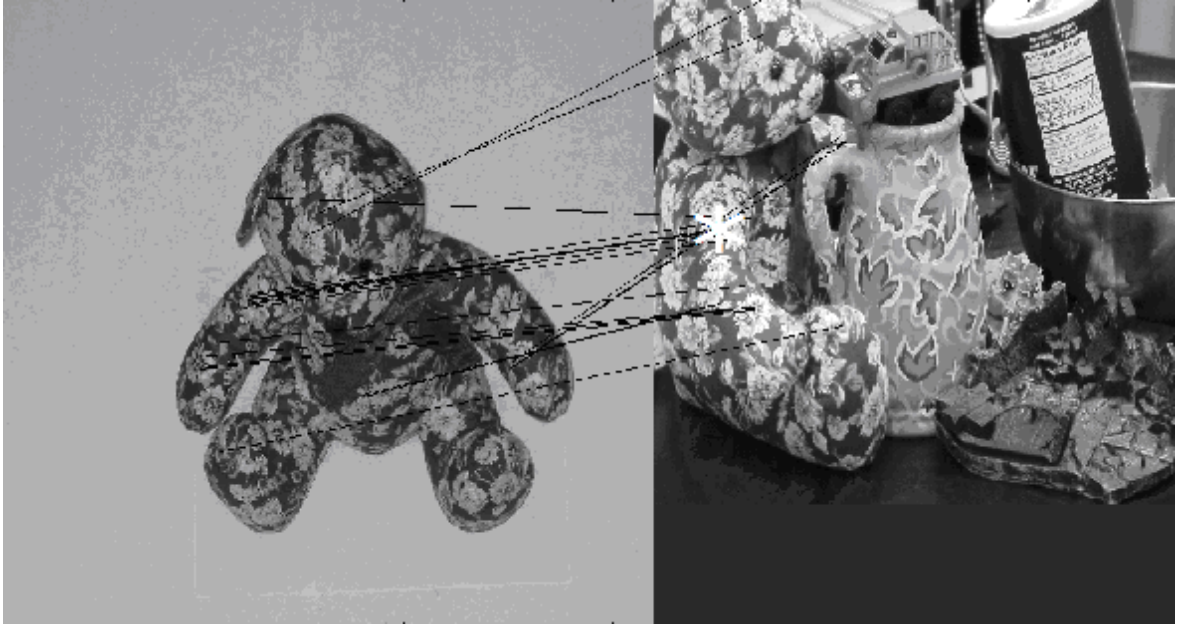
Şekil ek.18 Obj2-Train16 Test20



Şekil ek.19 Obj2-Train17 Test13



Şekil ek.20 Obj2-Train17 Test16



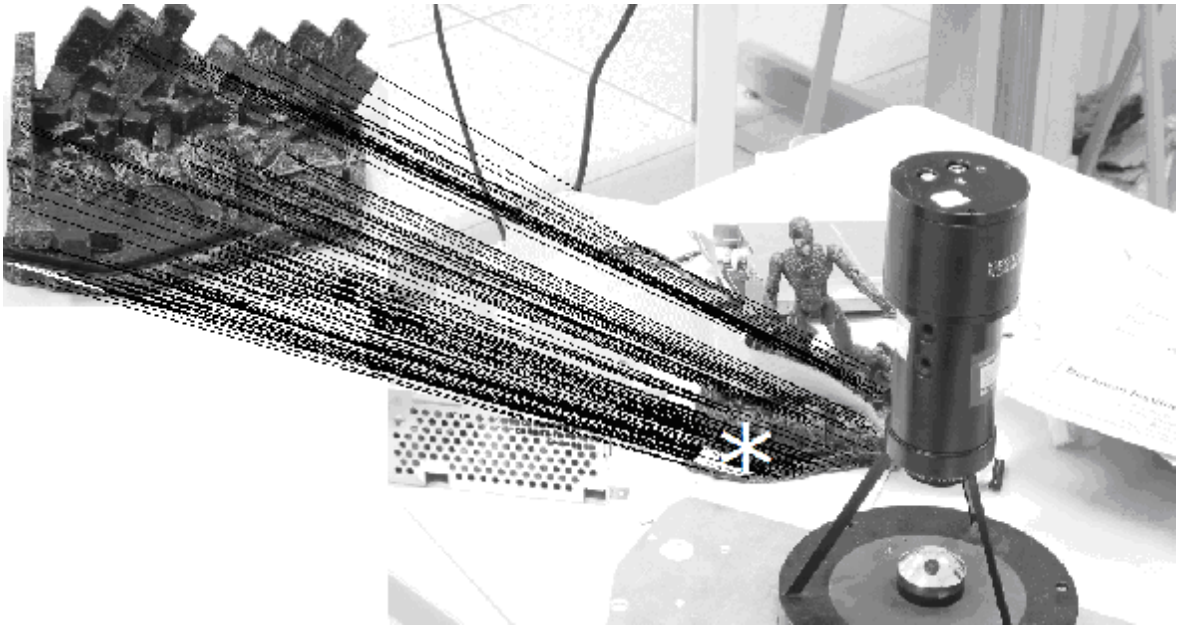
Şekil ek.21 Obj2-Train17 Test20



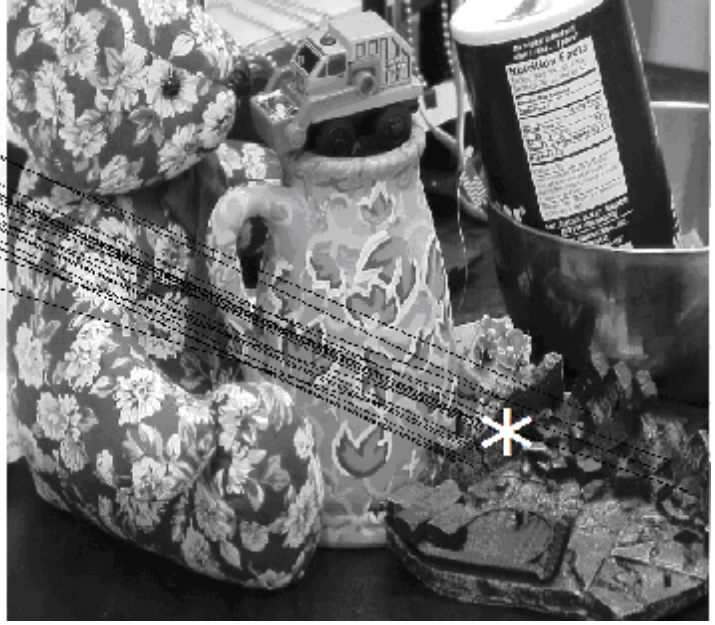
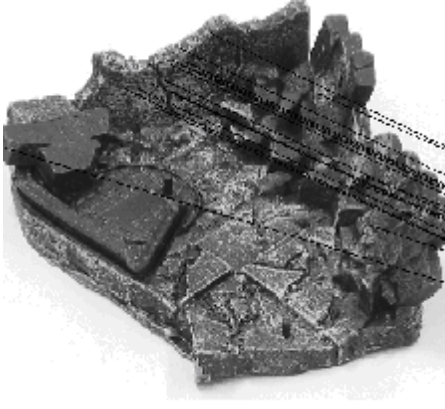
Şekil ek.22 Obj3-Train1 Test20



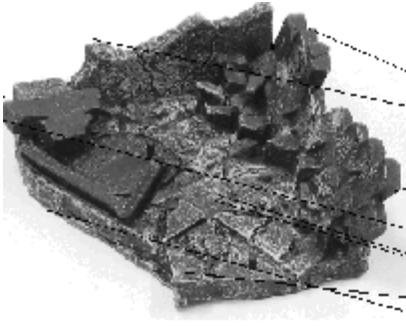
Şekil ek.23 Obj3-Train1 Test25



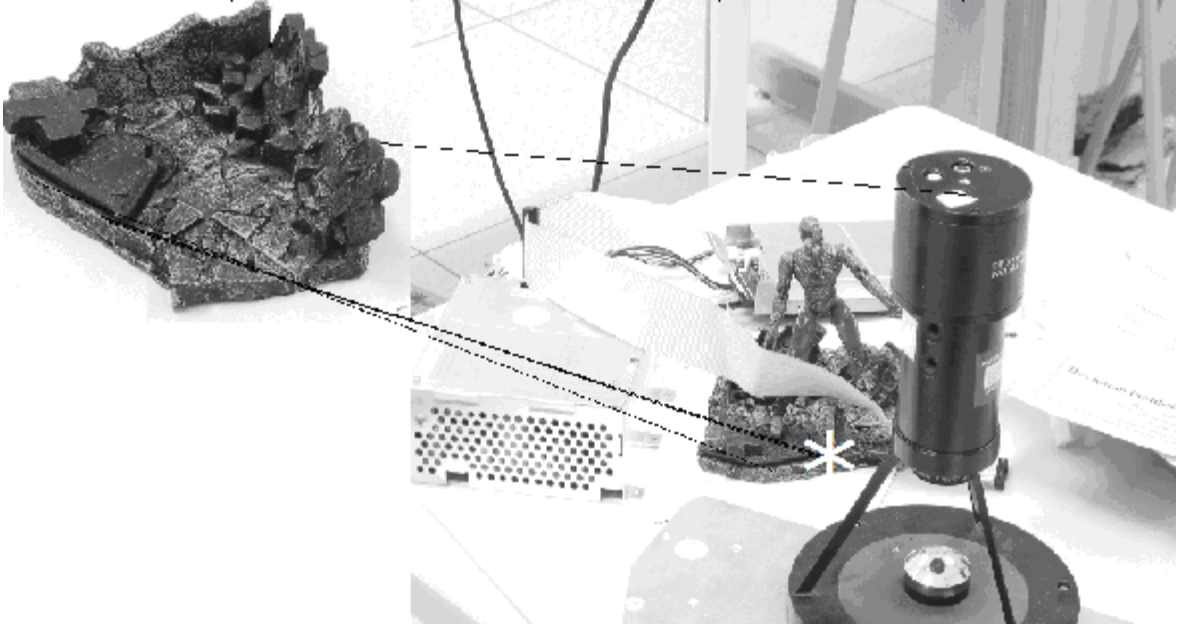
Şekil ek.24 Obj3-Train1 Test28



Şekil ek.25 Obj3-Train4 Test20



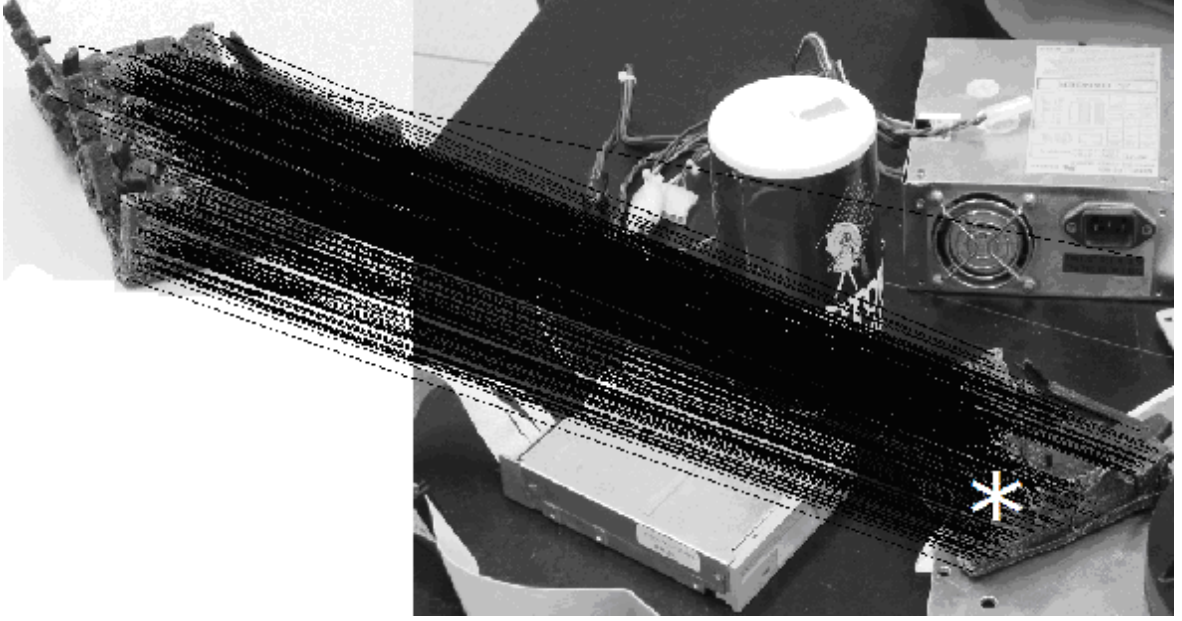
Şekil ek.26 Obj3-Train4 Test25



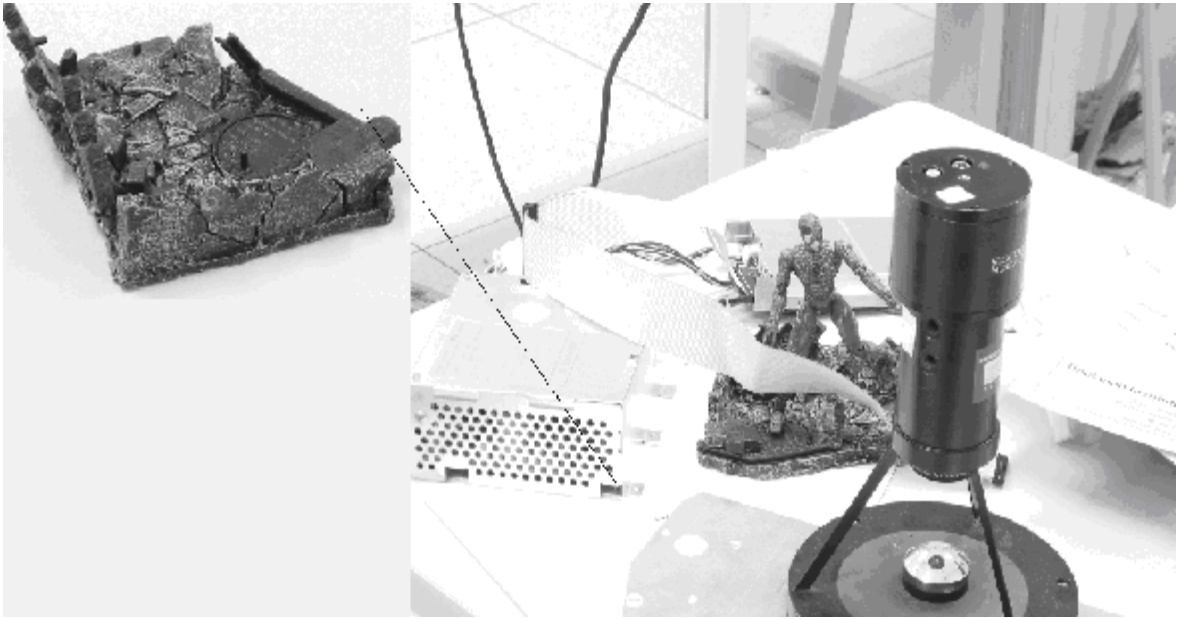
Şekil ek.27 Obj3-Train4 Test28



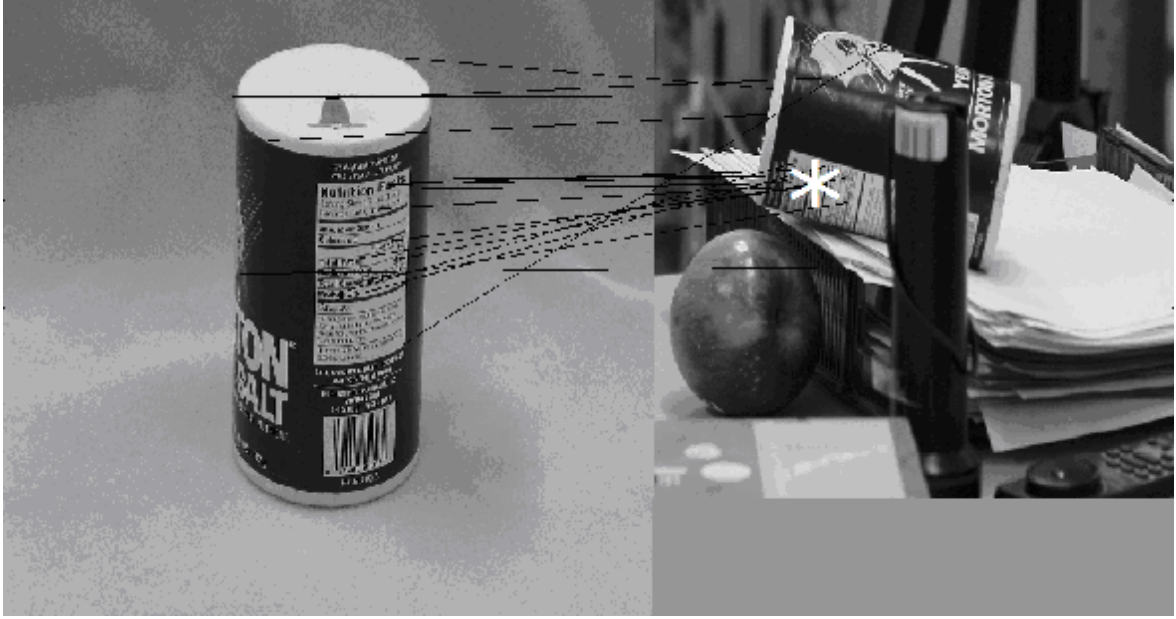
Şekil ek.28 Obj3-Train12 Test20



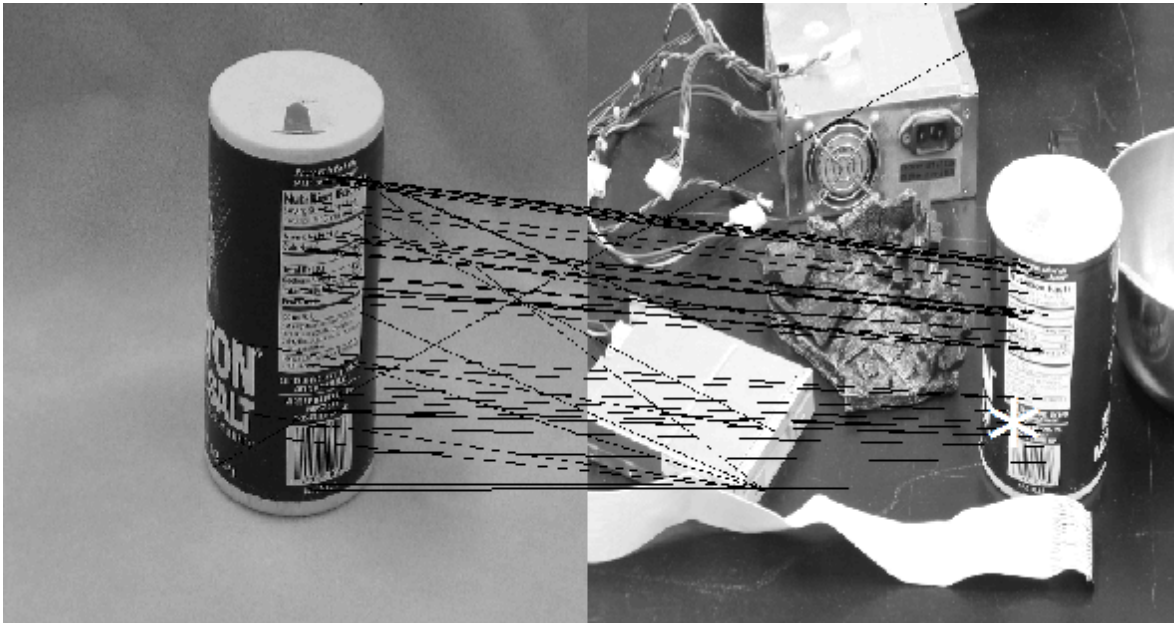
Şekil ek.29 Obj3-Train12 Test25



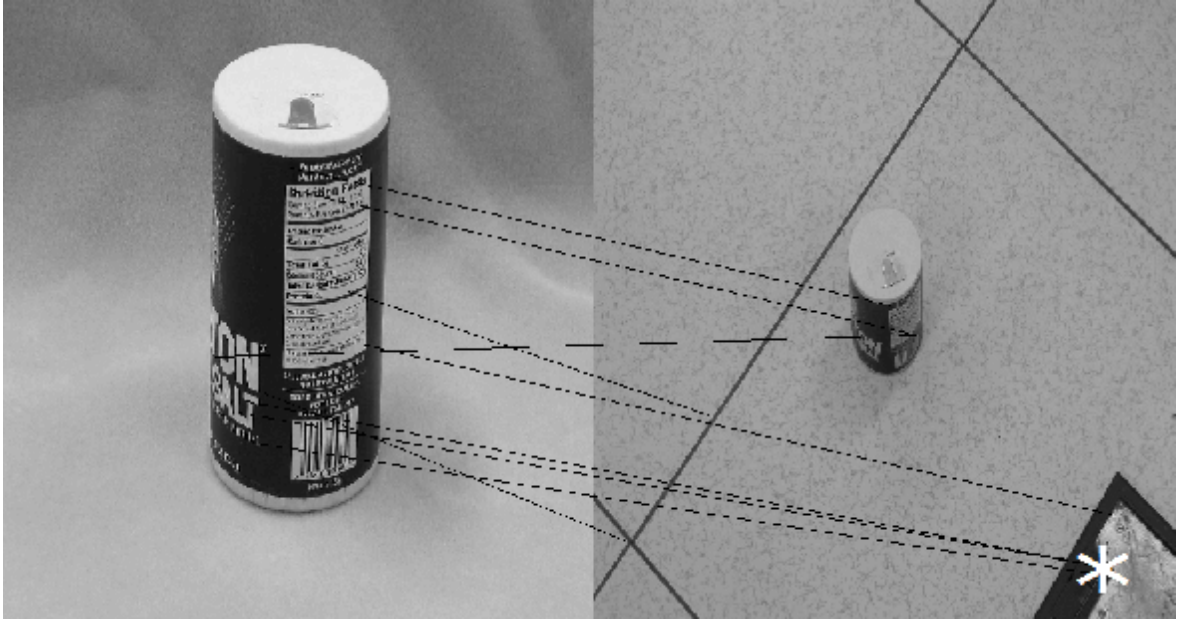
Şekil ek.30 Obj3-Train12 Test28



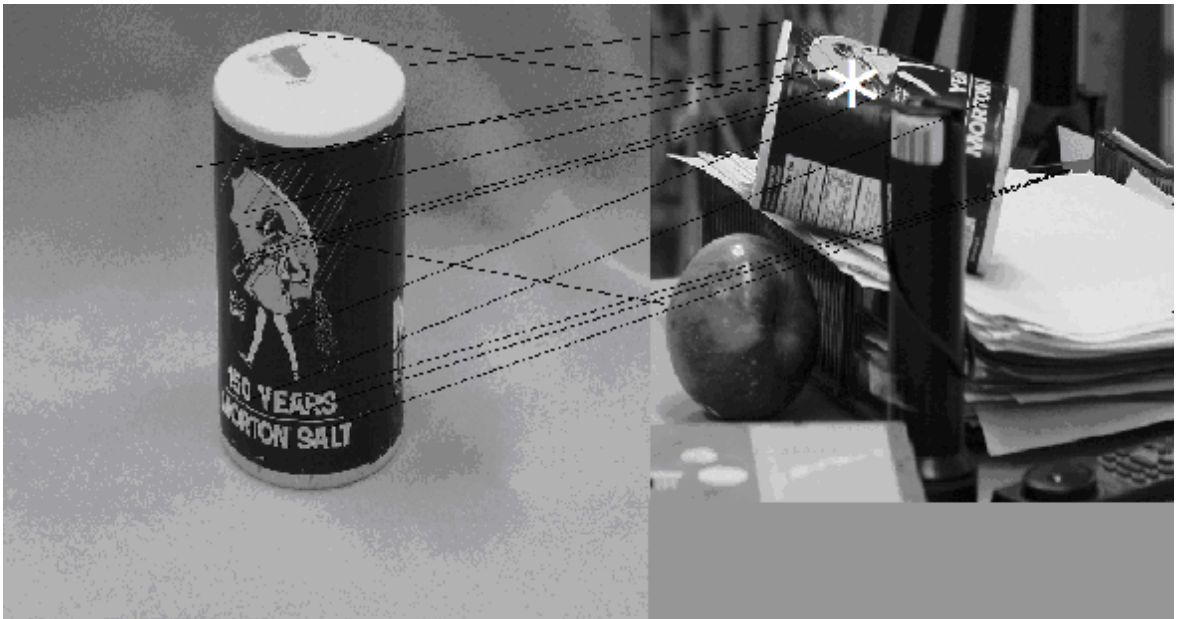
Şekil ek.31 Obj4-Train1 Test7



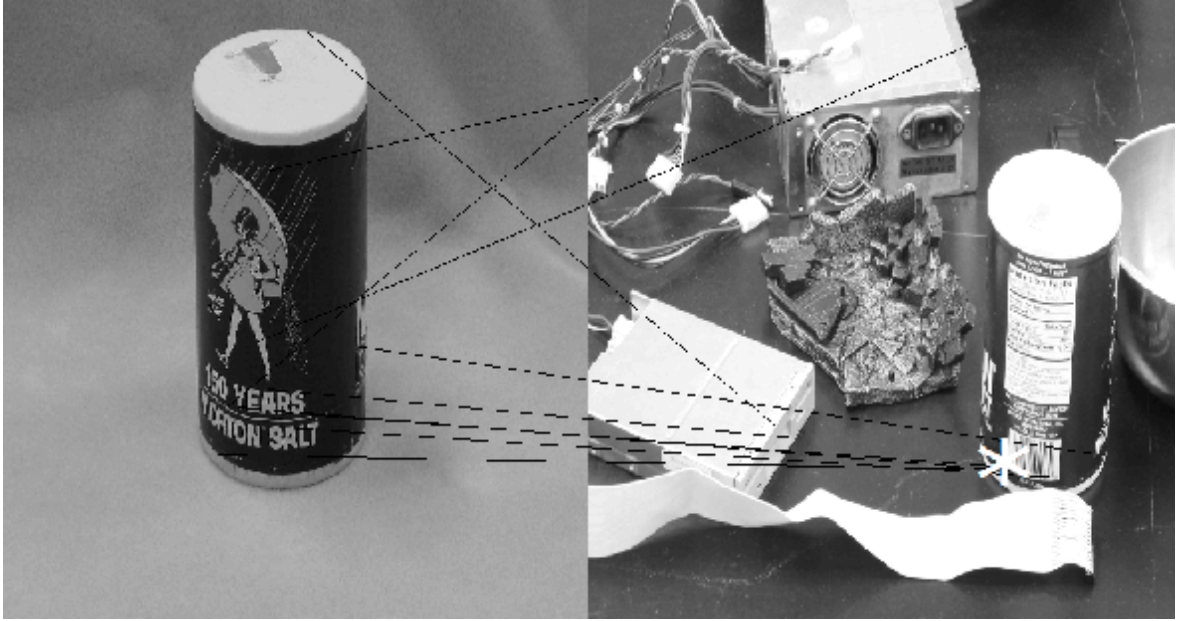
Şekil ek.32 Obj4-Train1 Test23



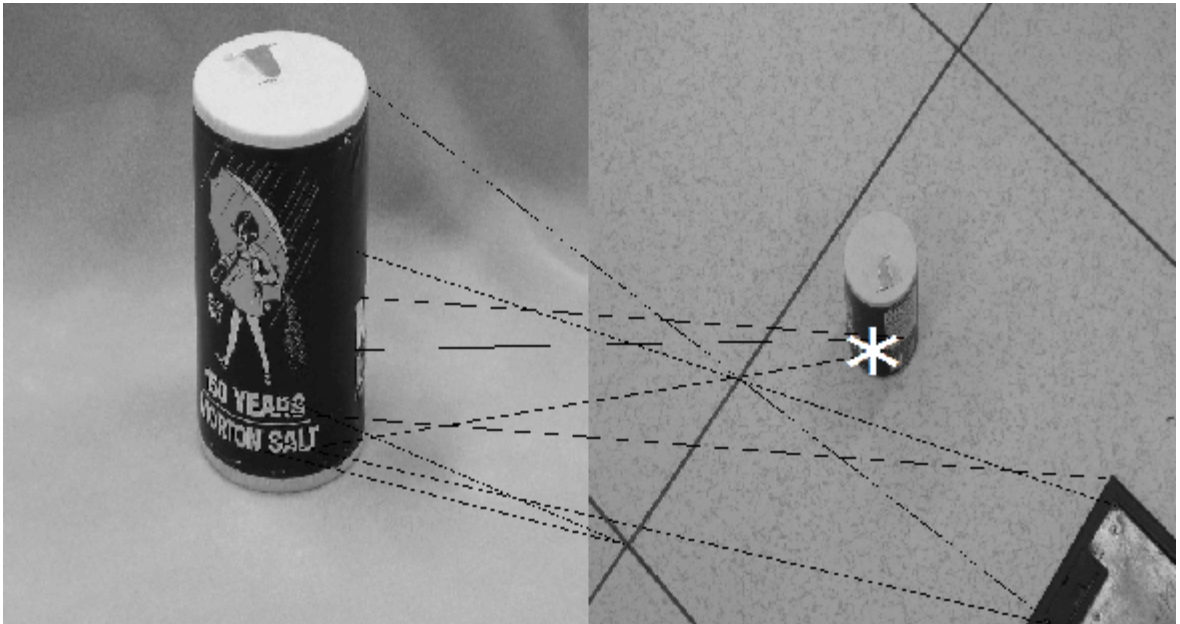
Şekil ek.33 Obj4-Train1 Test31



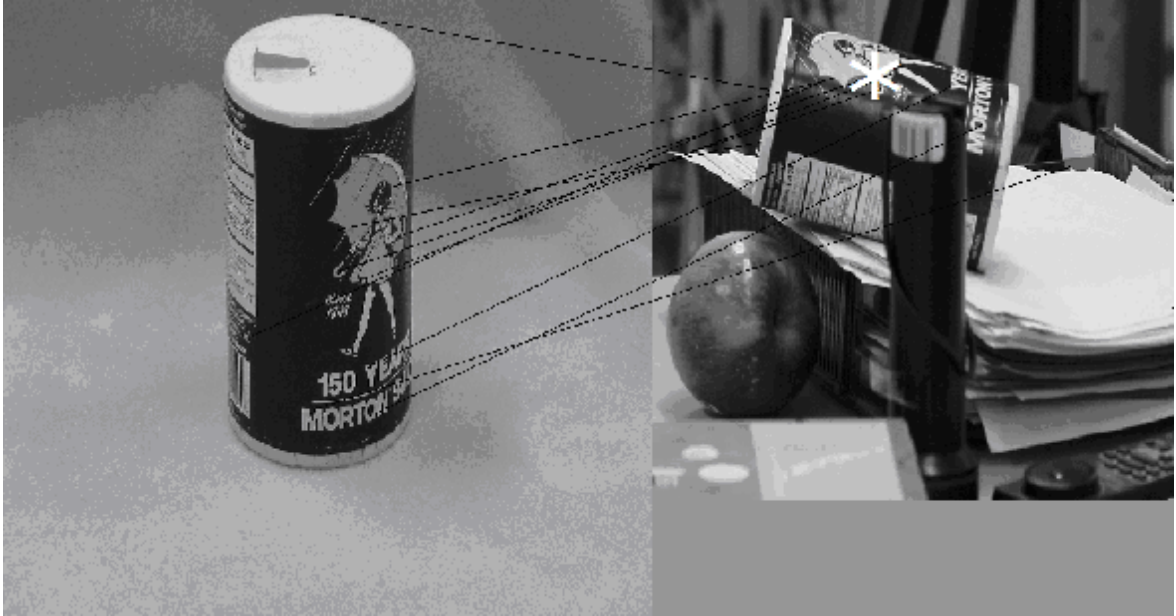
Şekil ek.34 Obj4-Train10 Test7



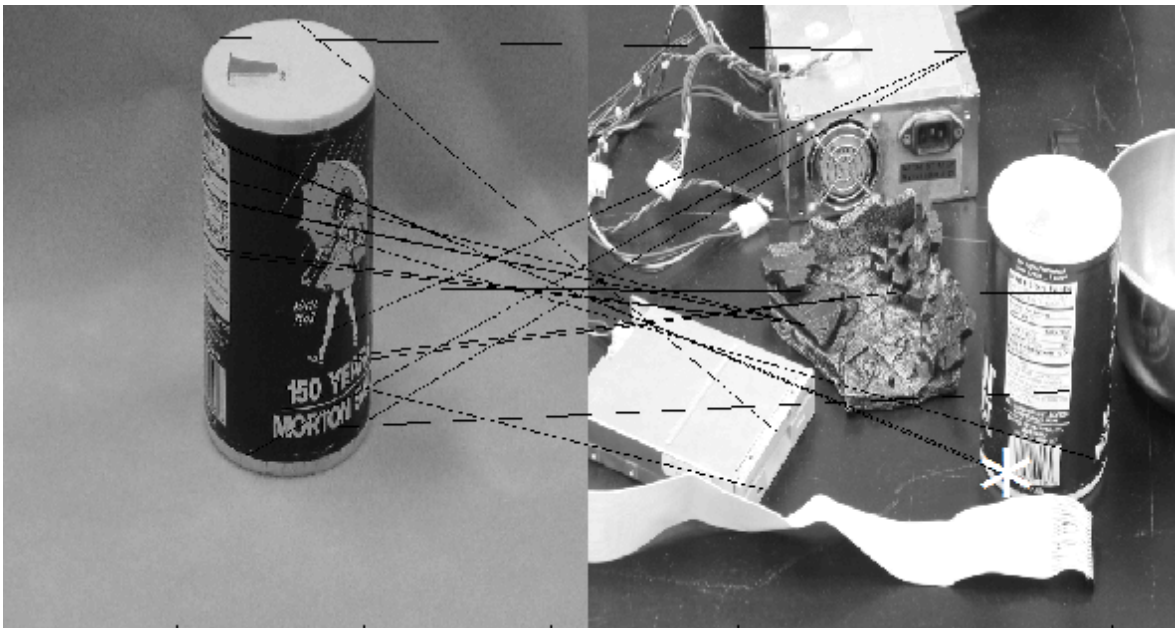
Şekil ek.35 Obj4-Train10 Test23



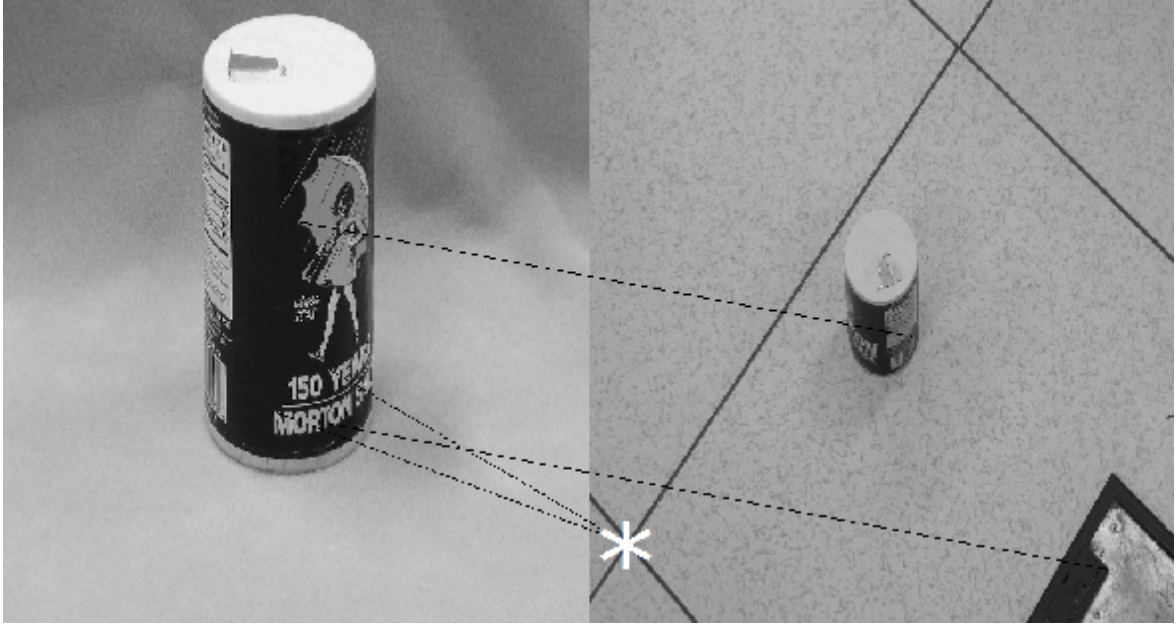
Şekil ek.36 Obj4-Train10 Test31



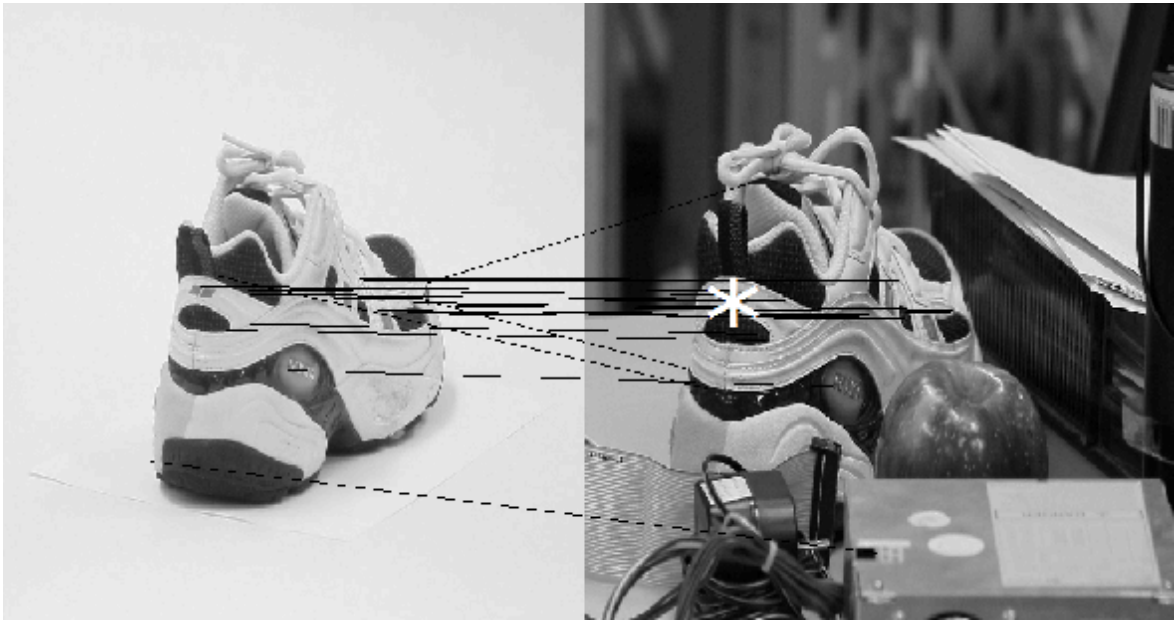
Şekil ek.37 Obj4-Train13 Test7



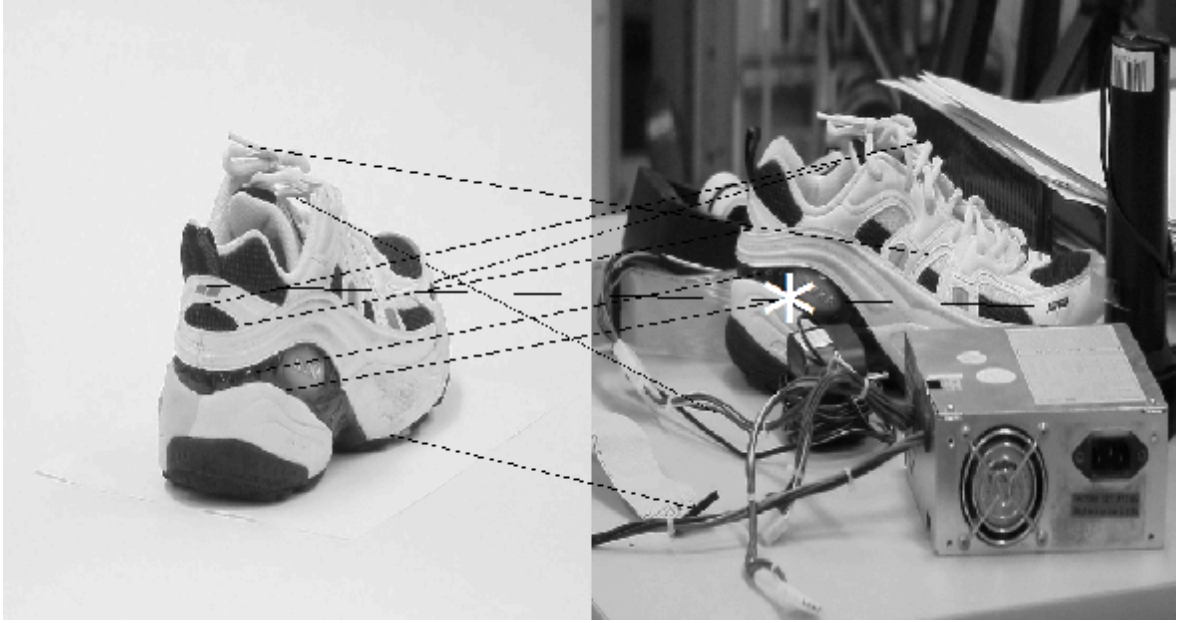
Şekil ek.38 Obj4-Train13 Test23



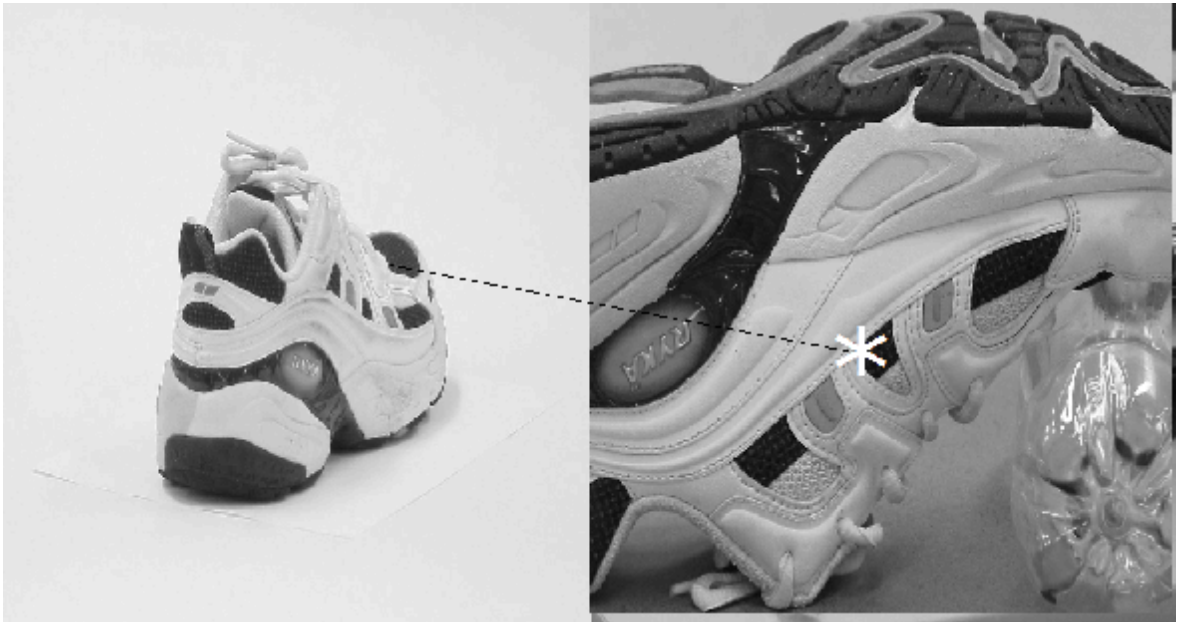
Şekil ek.39 Obj4-Train13 Test31



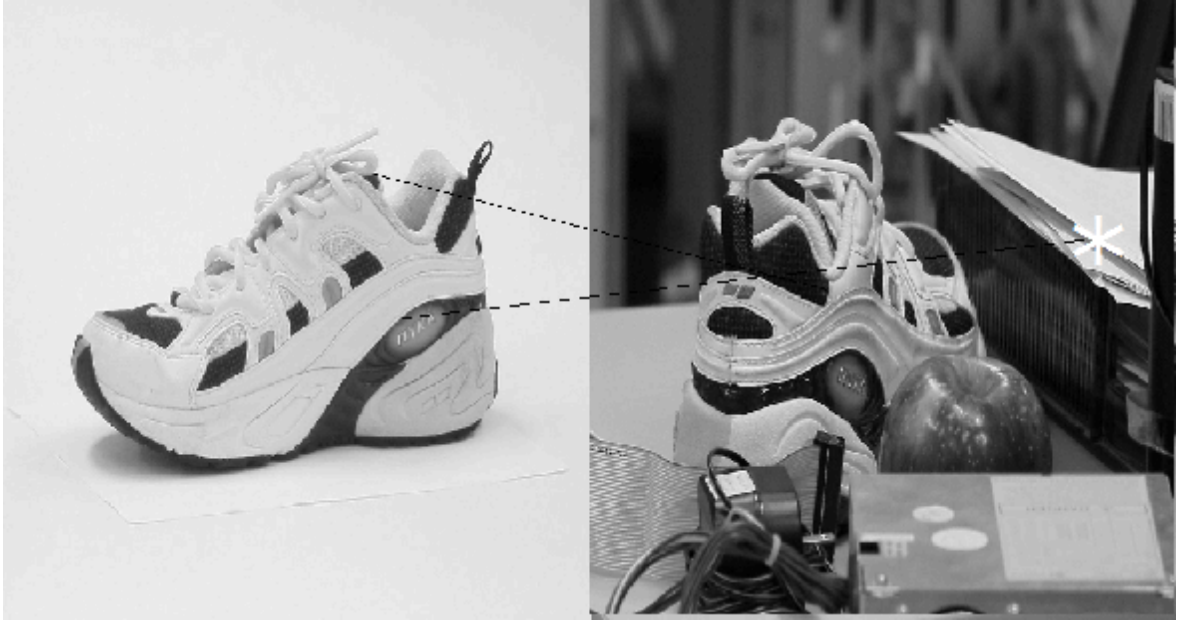
Şekil ek.40 Obj5-Train4 Test9



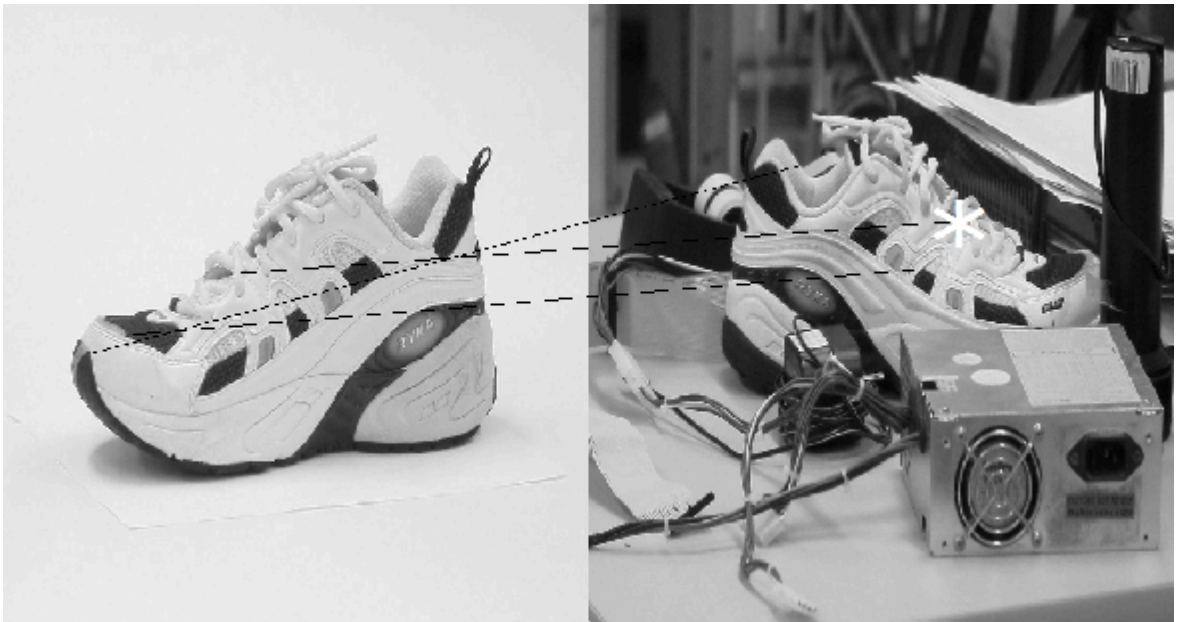
Şekil ek.41 Obj5-Train4 Test32



Şekil ek.42 Obj5-Train4 Test34



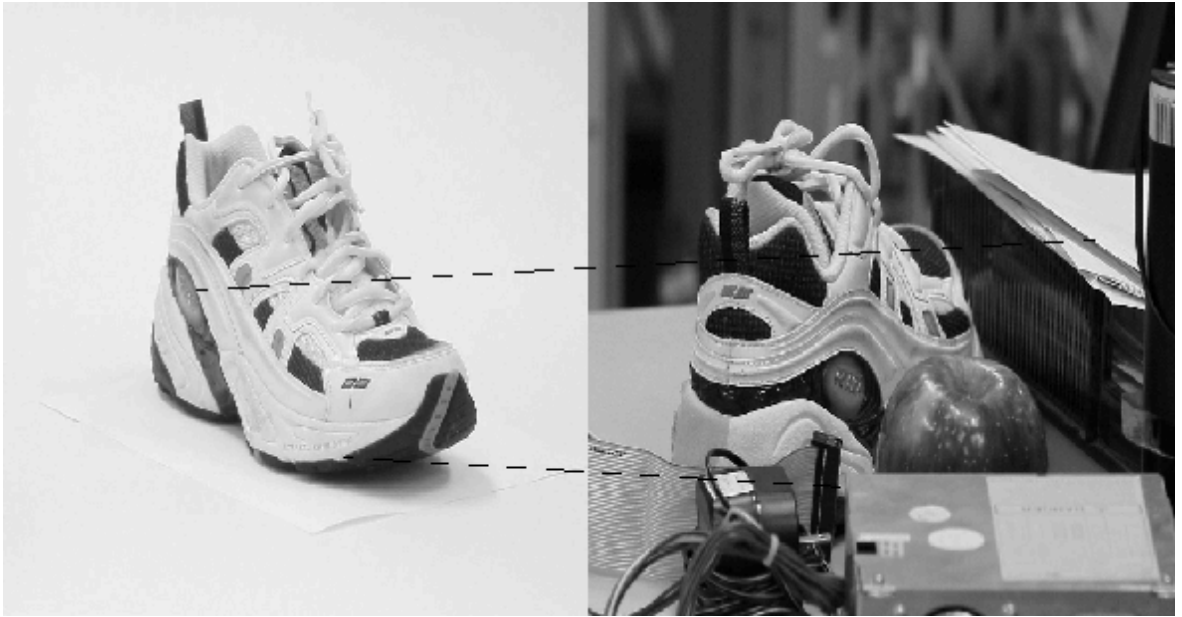
Şekil ek.43 Obj5-Train10 Test9



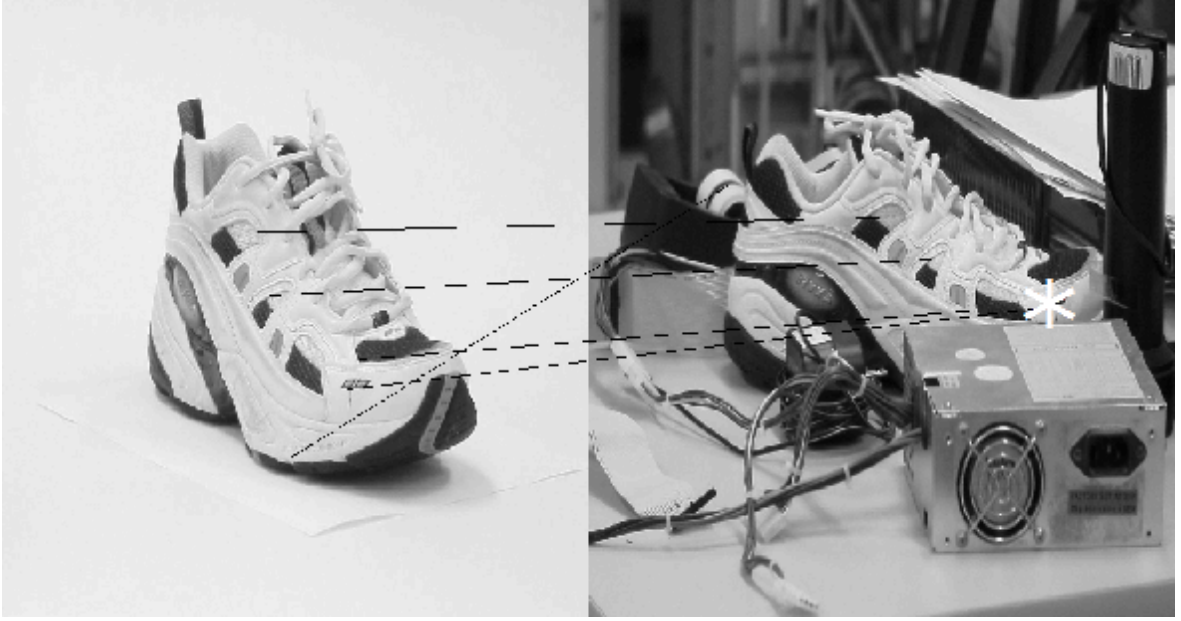
Şekil ek.44 Obj5-Train10 Test32



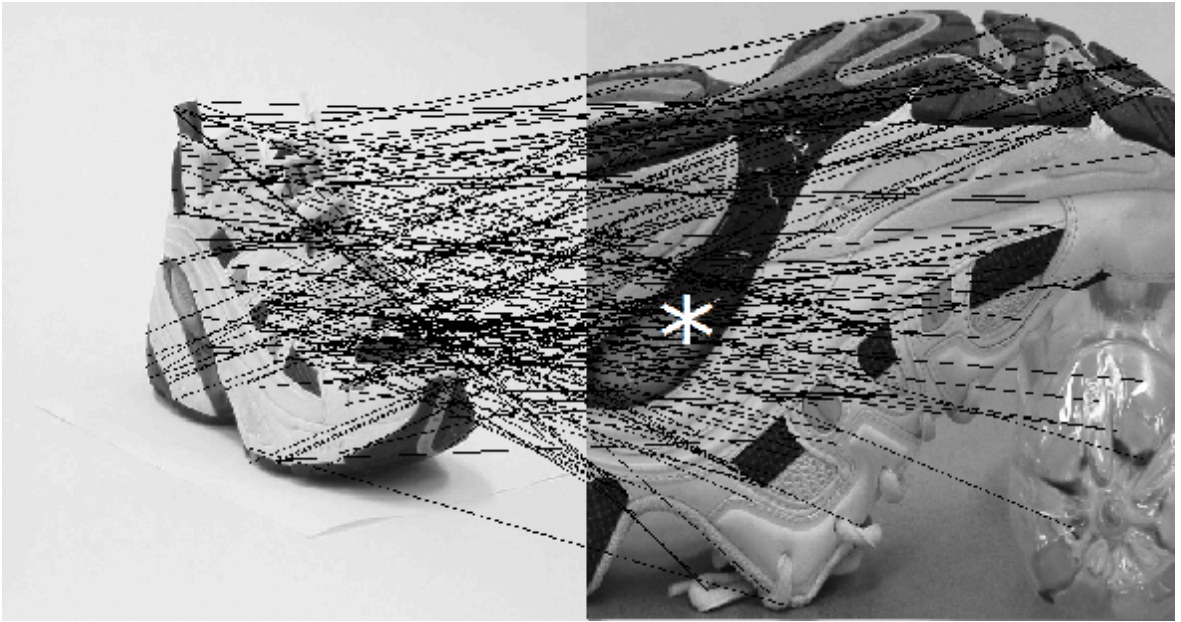
Şekil ek.45 Obj5-Train10 Test34



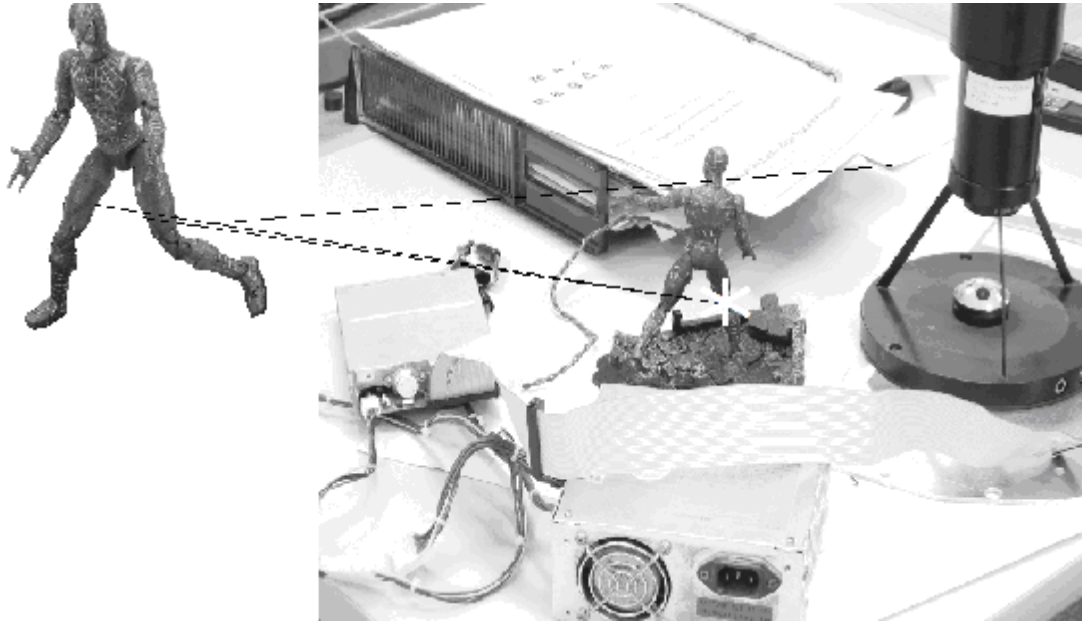
Şekil ek.46 Obj5-Train15 Test9



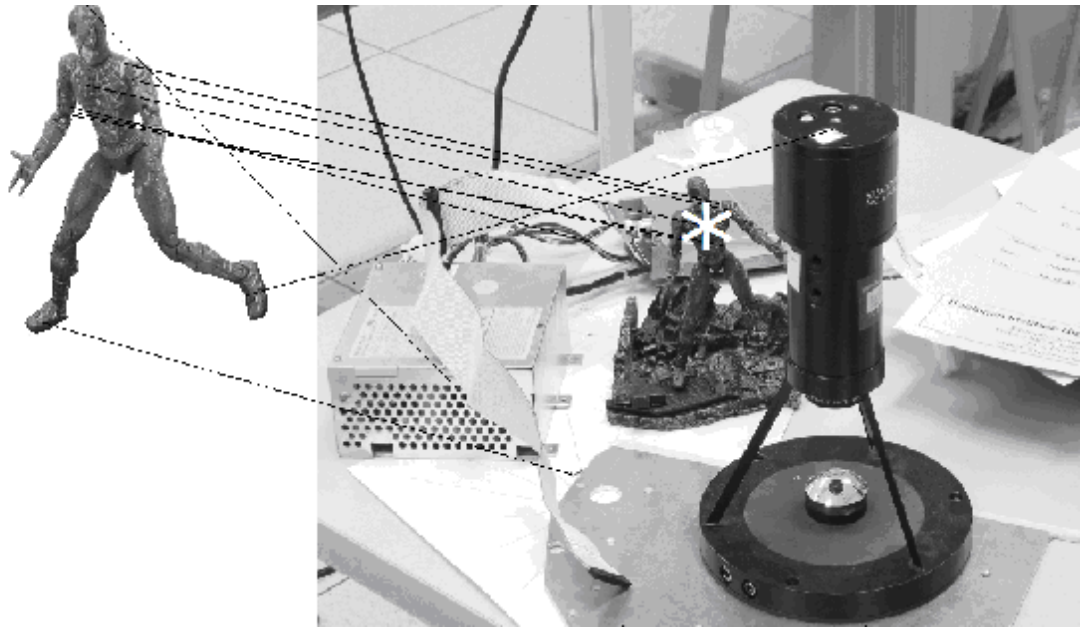
Şekil ek.47 Obj5-Train15 Test32



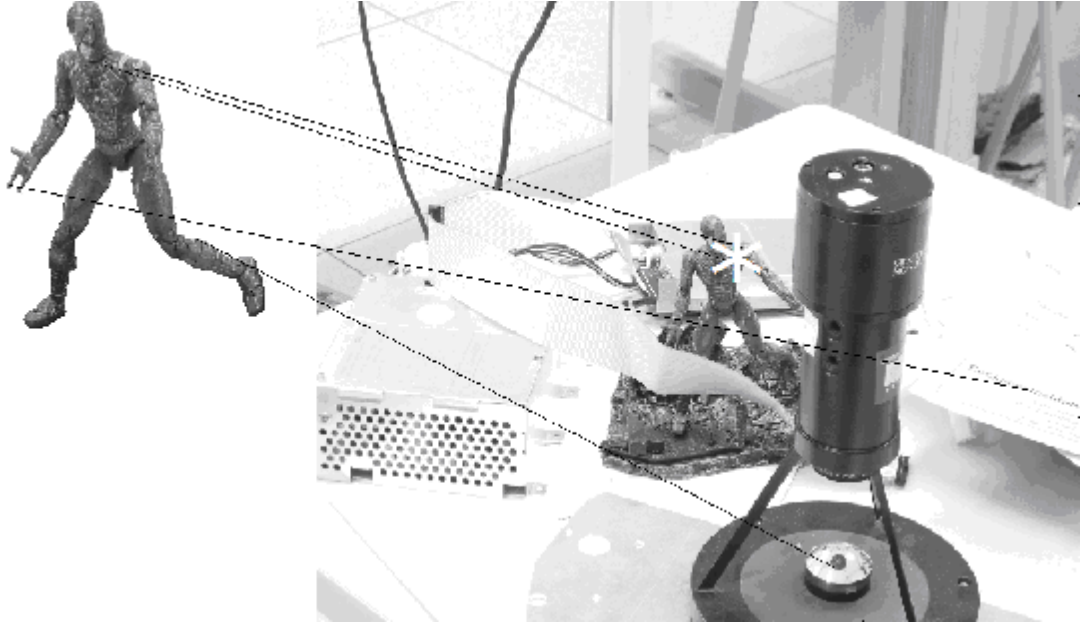
Şekil ek.48 Obj5-Train15 Test34 (Fark kriteri artırılmıştır)



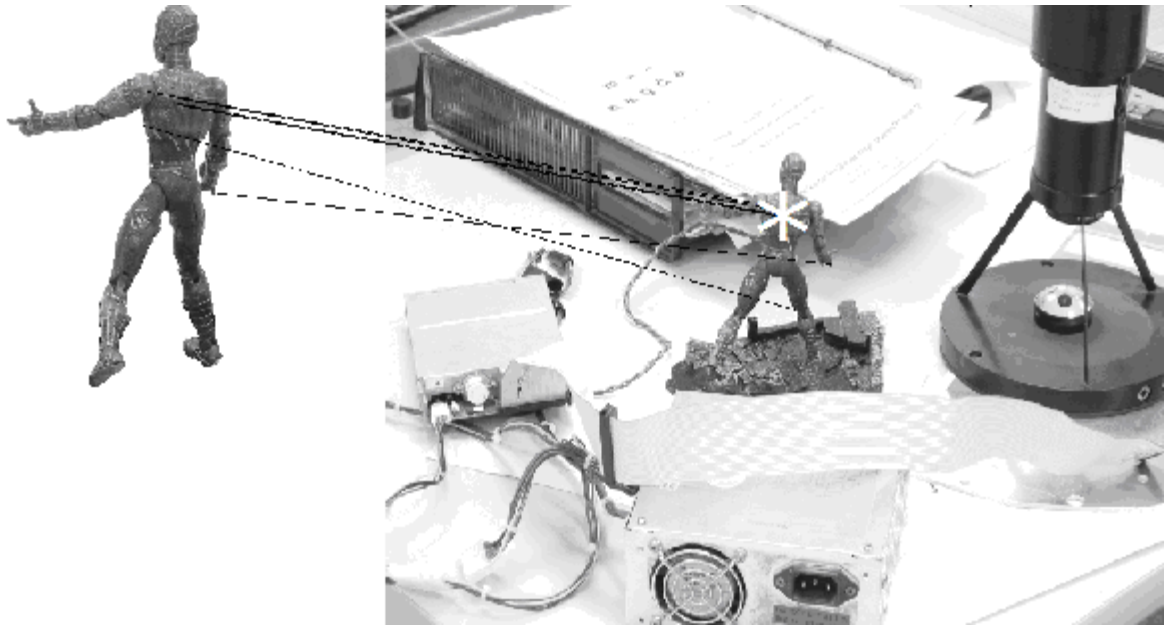
Şekil ek.49 Obj6-Train3 Test26



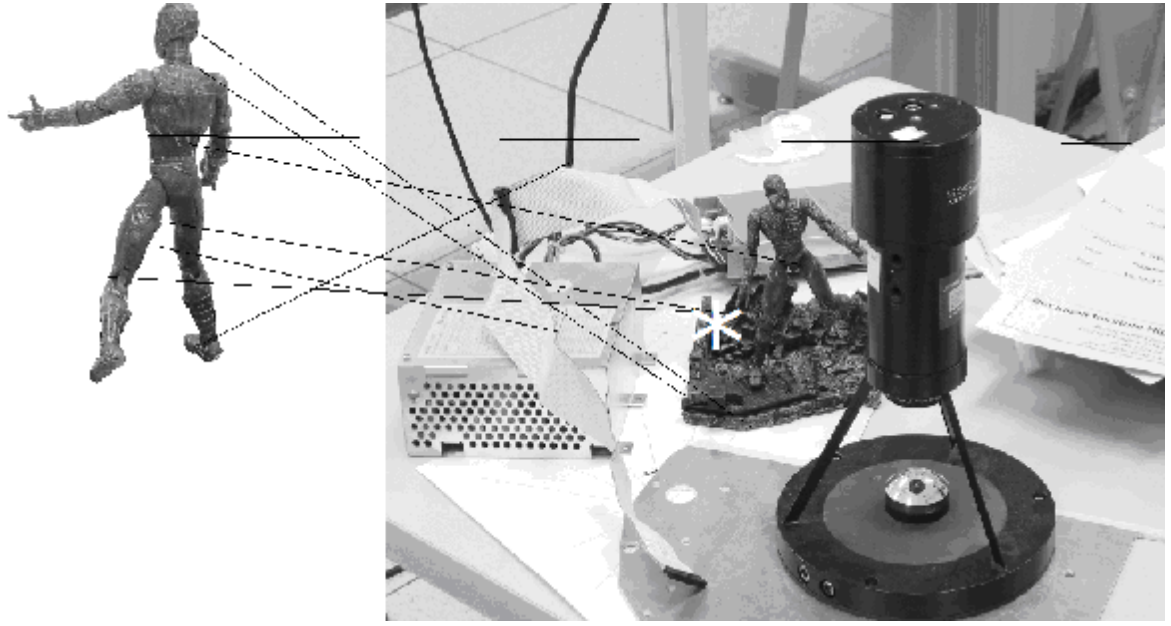
Şekil ek.50 Obj6-Train3 Test27



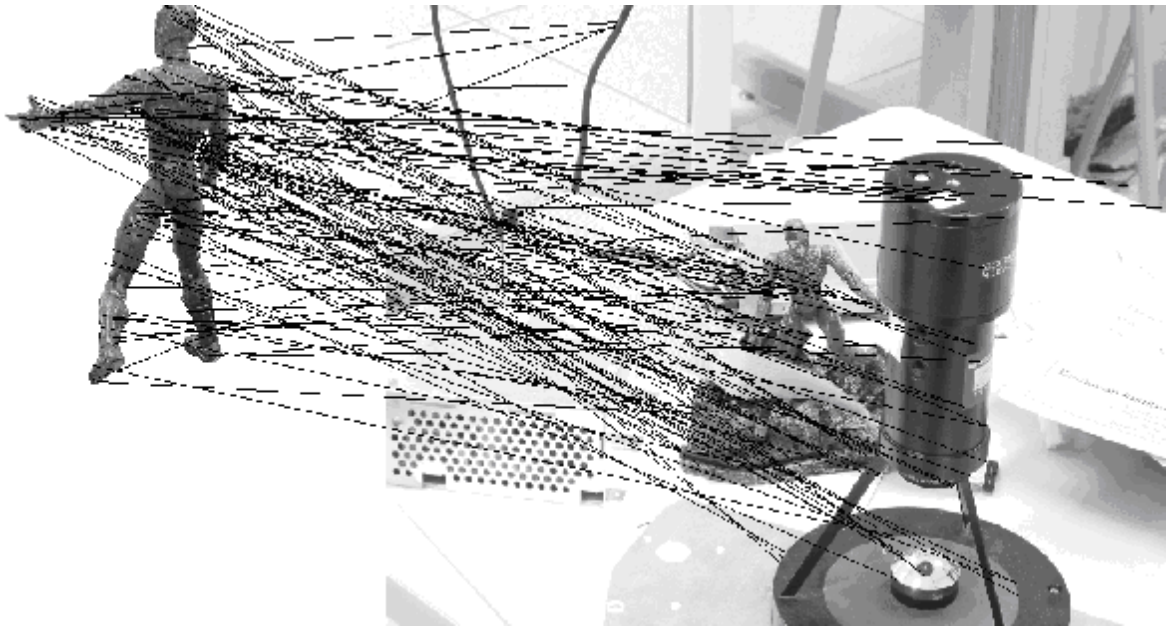
Şekil ek.51 Obj6-Train3 Test28



Şekil ek.52 Obj6-Train8 test26



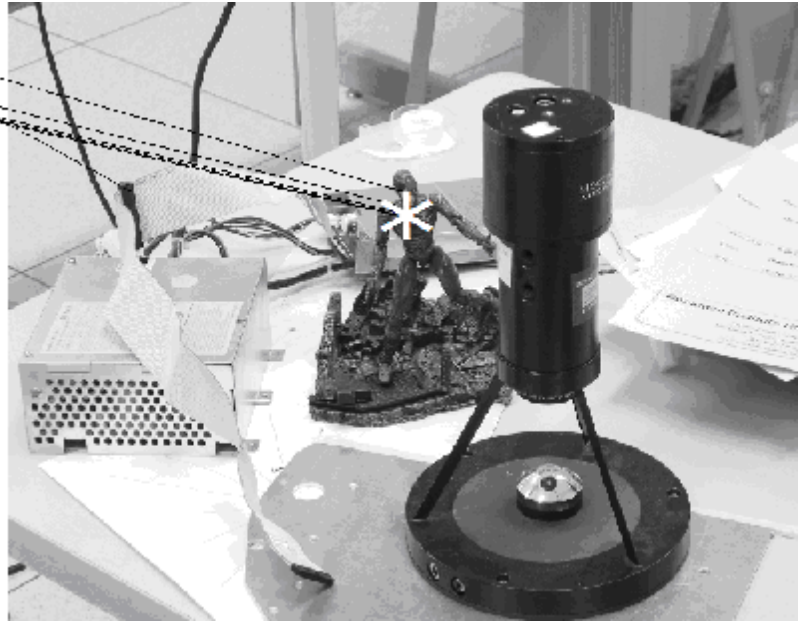
Şekil ek.53 Obj6-Train8 Test27



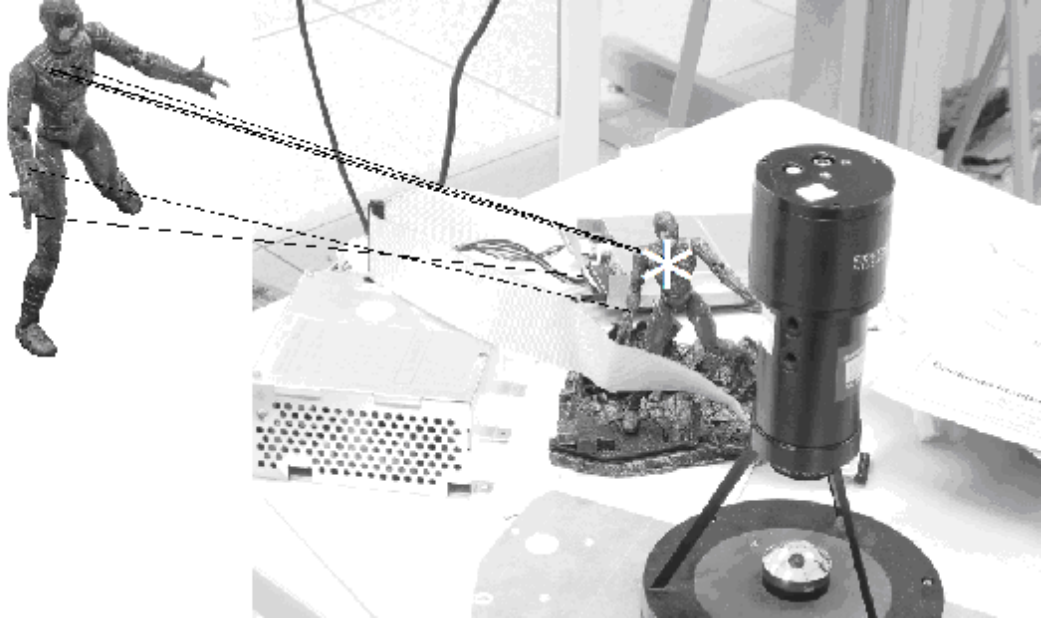
Şekil ek.54 Obj6-Train8 Test28 (Fark kriteri artırılmıştır)



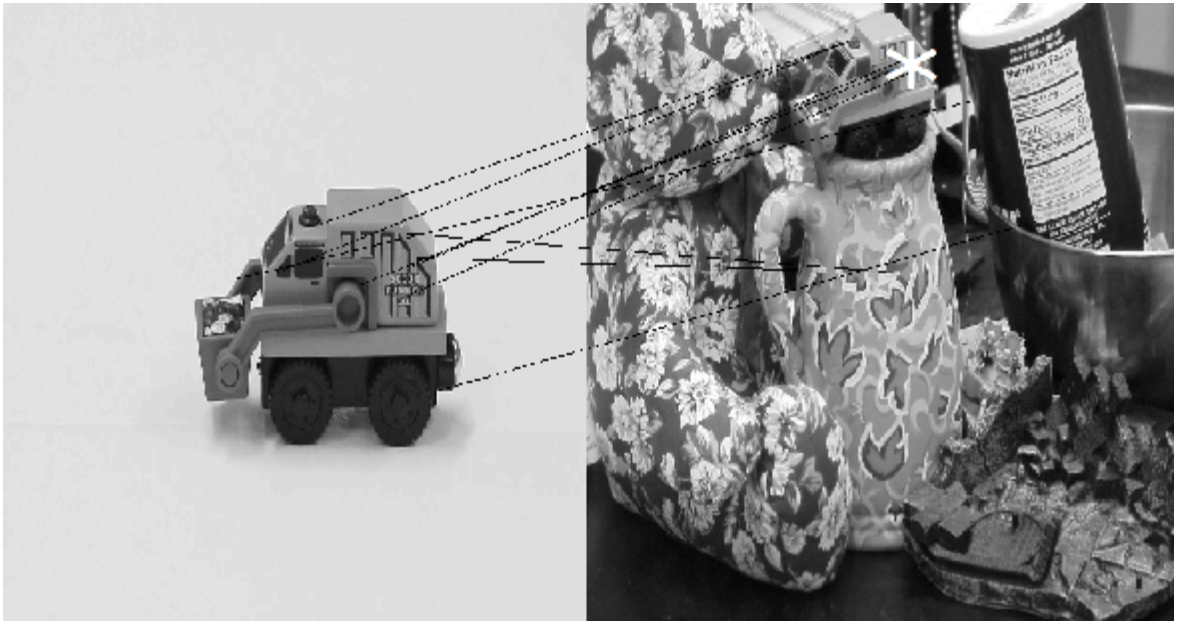
Şekil ek.55 Obj6-Train16 Test26



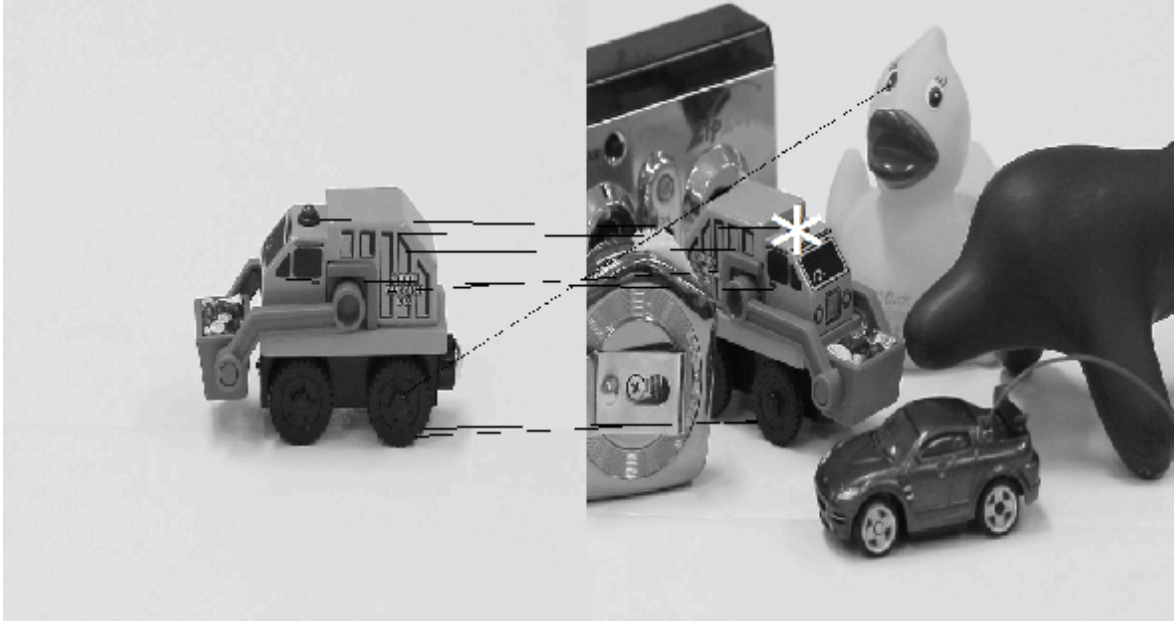
Şekil ek.56 Obj6-Train16 Test27



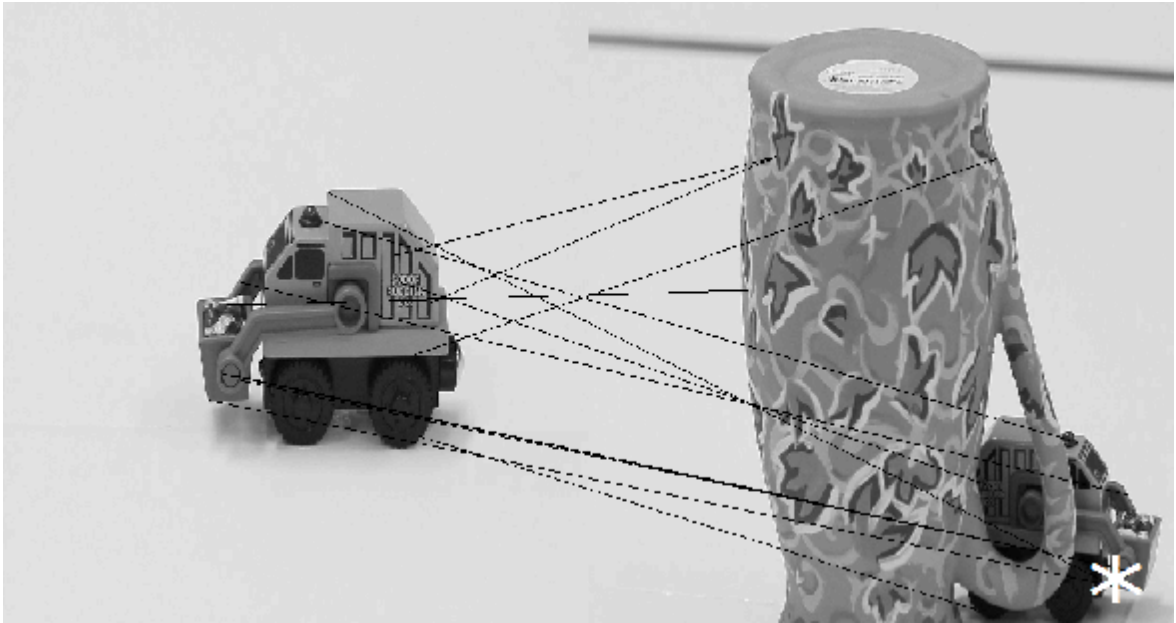
Şekil ek.57 Obj6-Train16 Test28



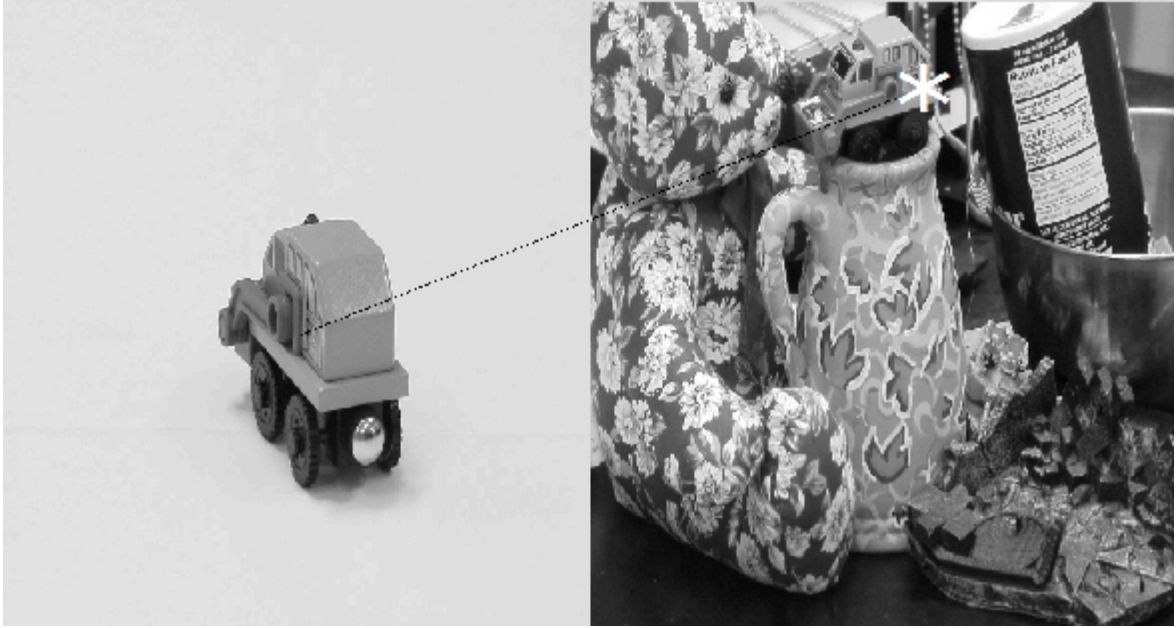
Şekil ek.58 Obj7-Train5 Test20



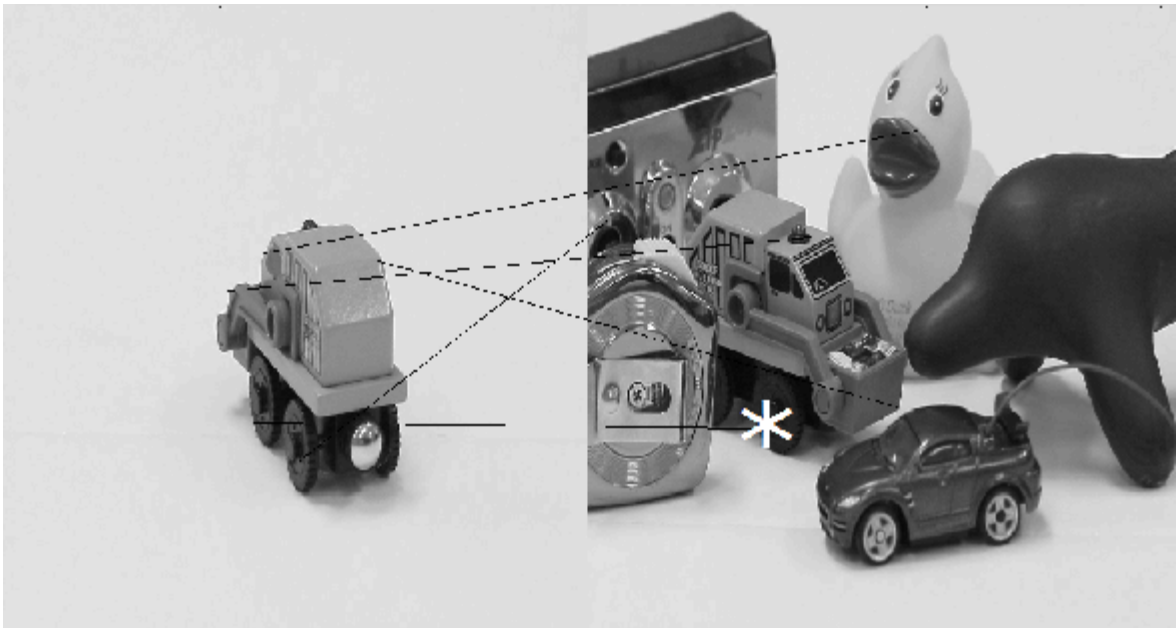
Şekil ek.59 Obj7-Train5 Test38



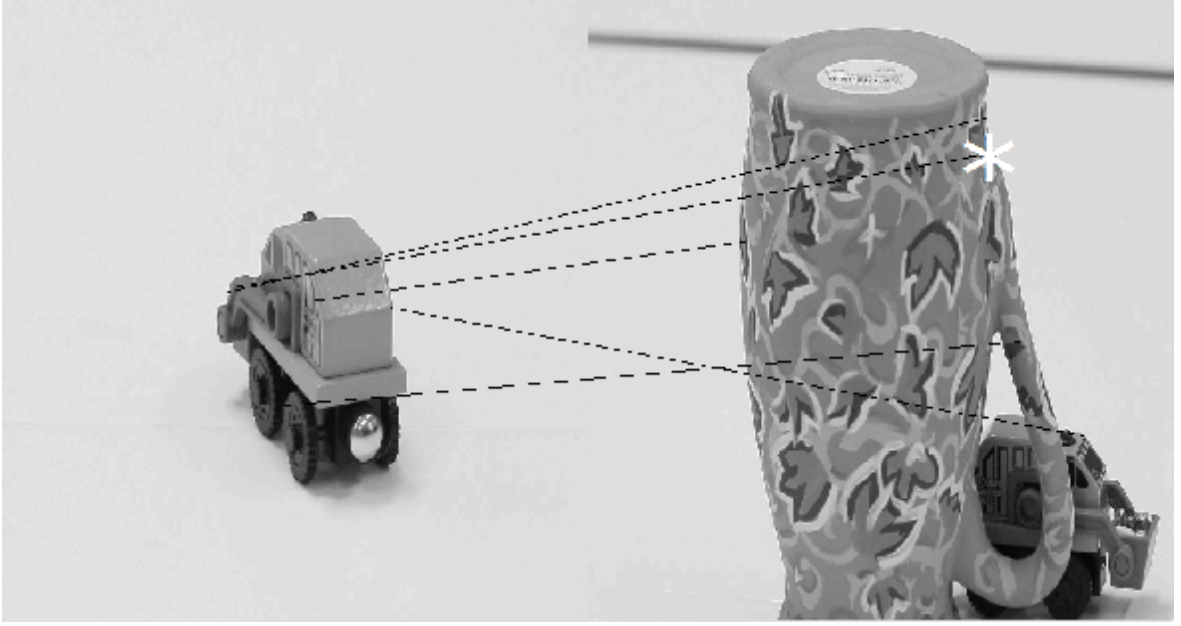
Şekil ek.60 Obj7-Train5 Test45



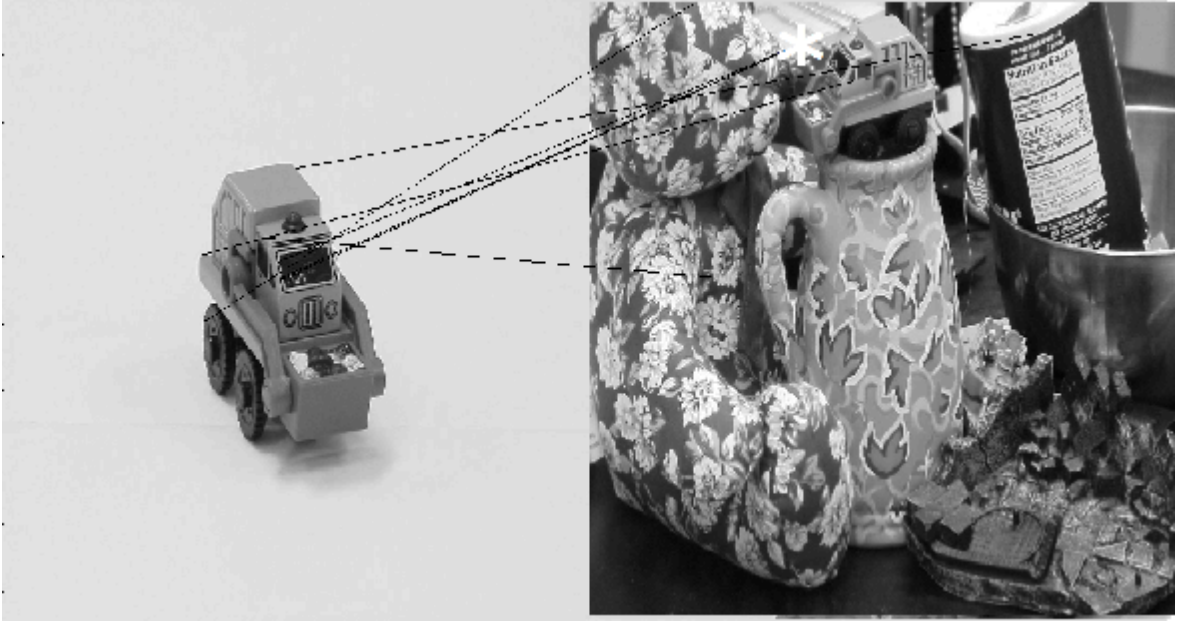
Şekil ek.61 Obj7-Train8 Test20



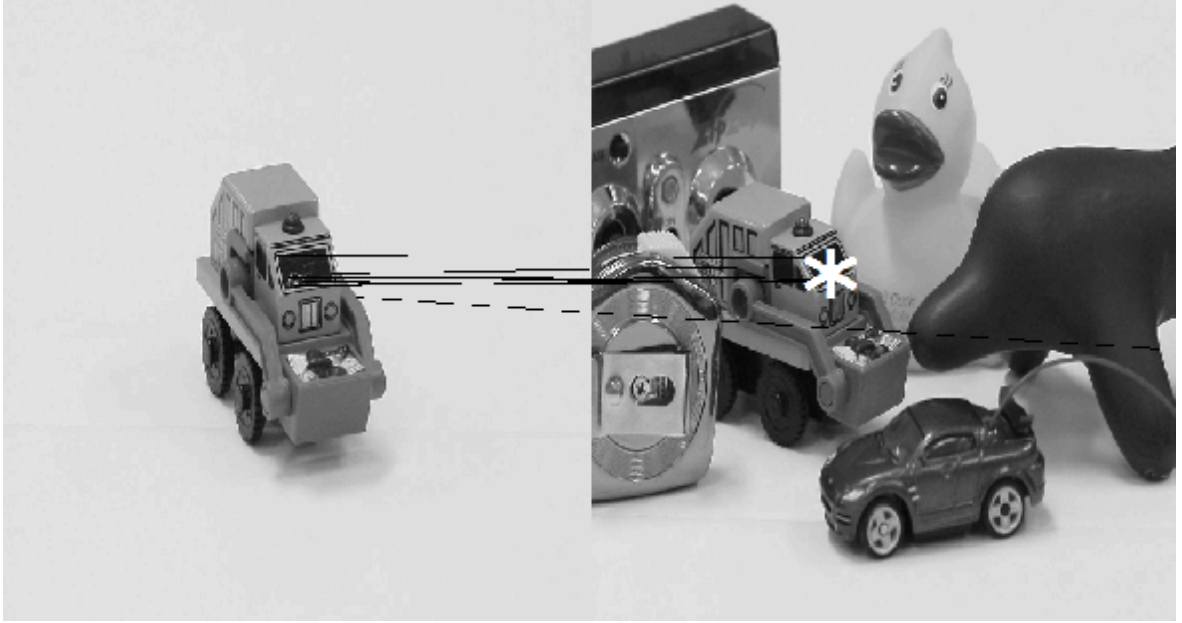
Şekil ek.62 Obj7-Train8 Test38



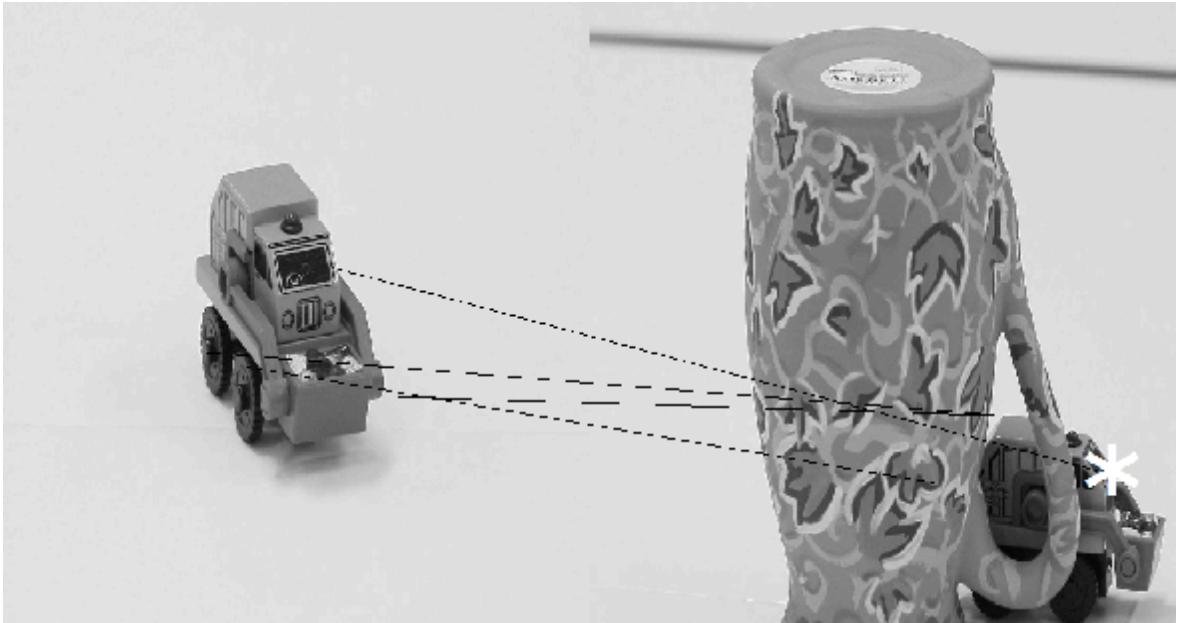
Şekil ek.63 Obj7-Train8 Test45



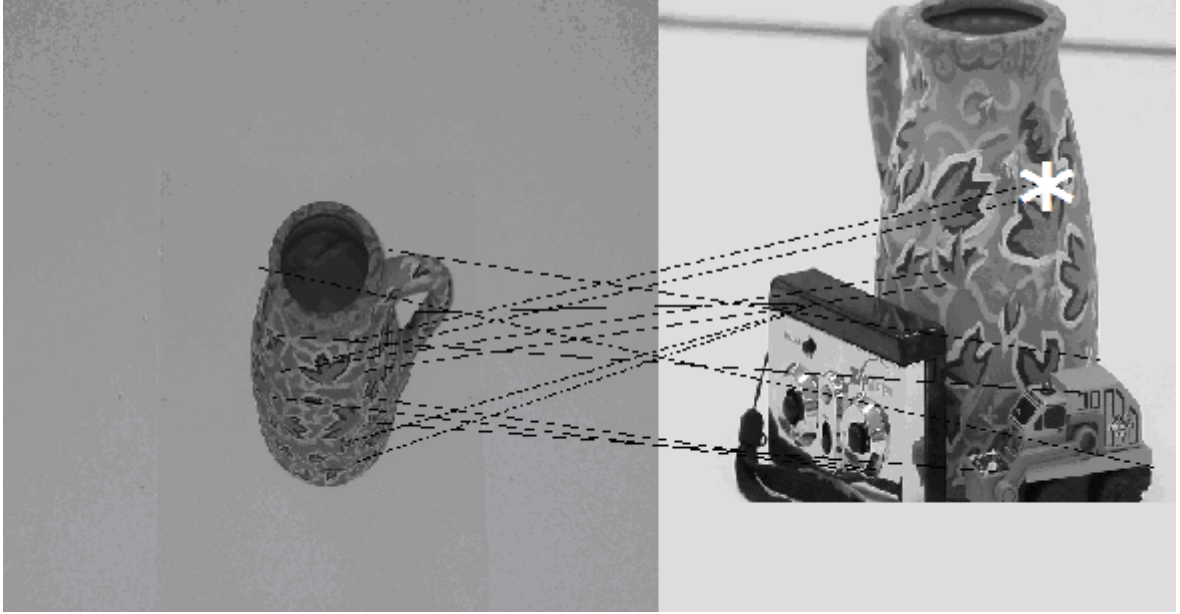
Şekil ek.64 Obj7-Train16 Test20



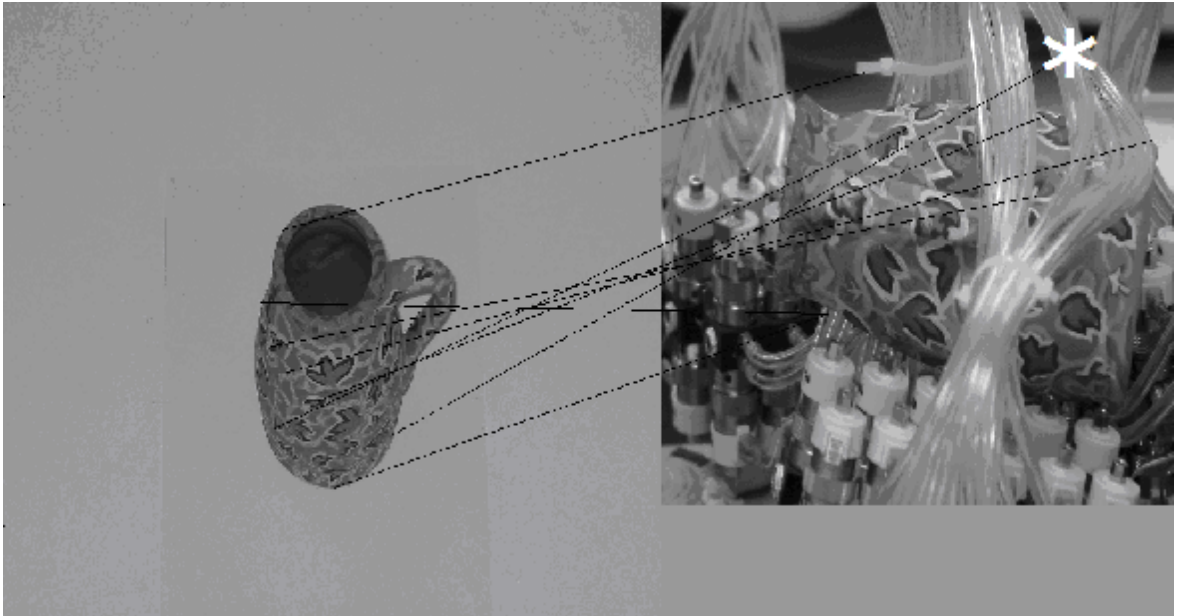
Şekil ek.65 Obj7-Train16 Test38



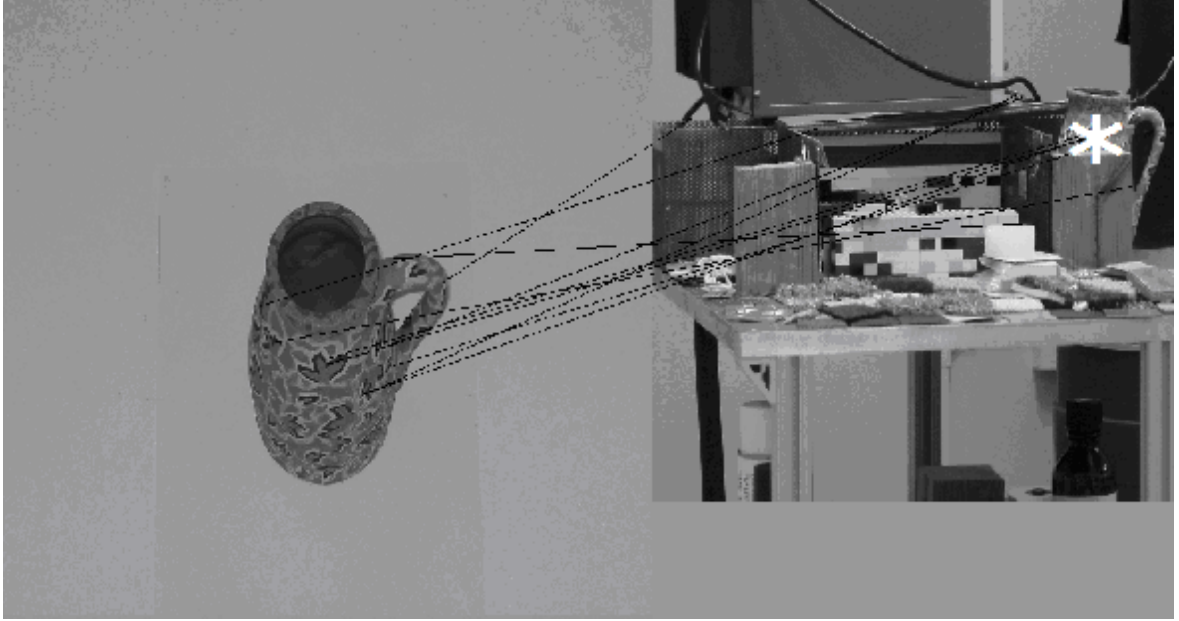
Şekil ek.66 Obj7-Train16 Test45



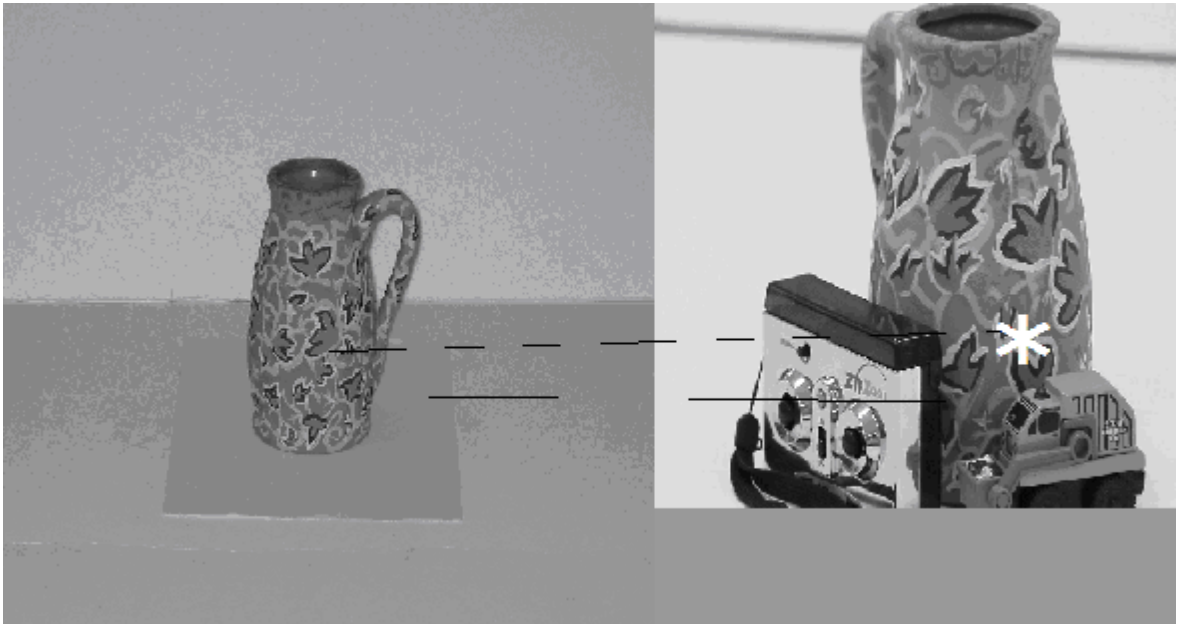
Şekil ek.67 Obj8-Train2 Test44



Şekil ek.68 Obj8-Train2 Test50



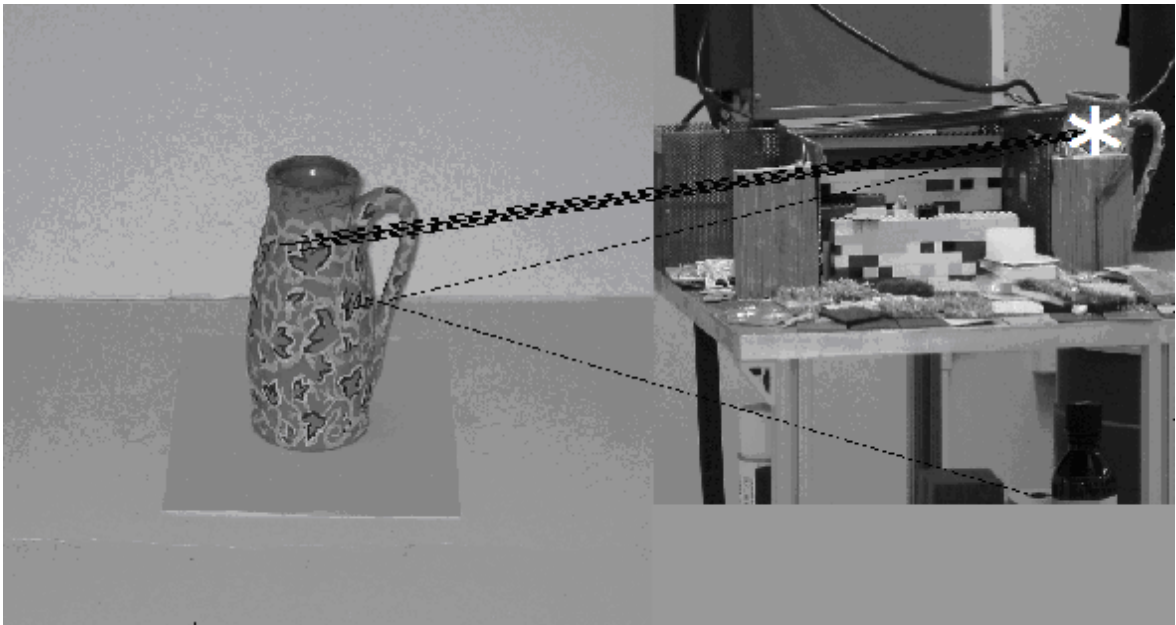
Şekil ek.69 Obj8-Train2 Test51



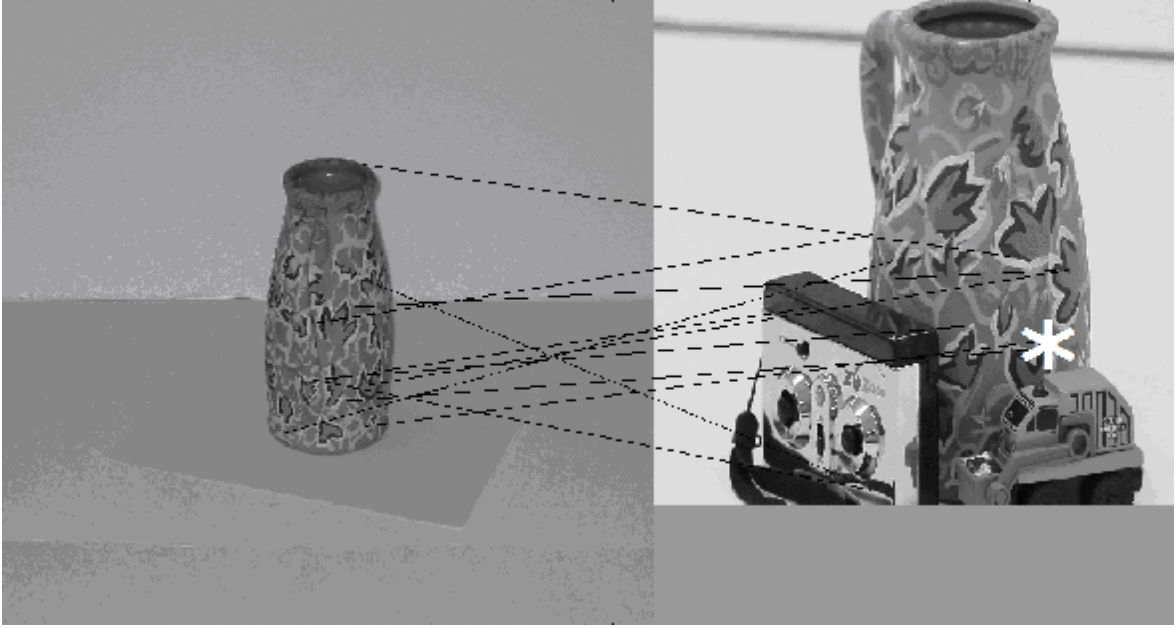
Şekil ek.70 Obj8-Train9 Test44



Şekil ek.71 Obj8-Train9 Test50



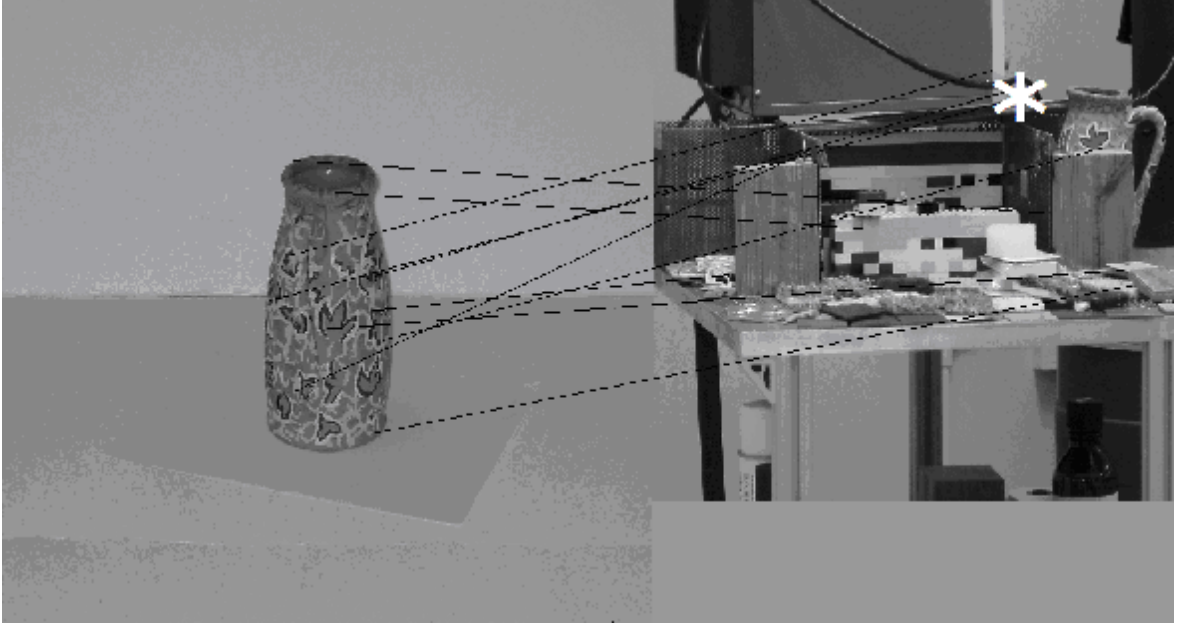
Şekil ek.72 Obj8-Train9 Test51



Şekil ek.73 Obj8-Train14 Test44



Şekil ek.74 Obj8-Train14 Test50



Şekil ek.75 Obj8-Train14 Test51

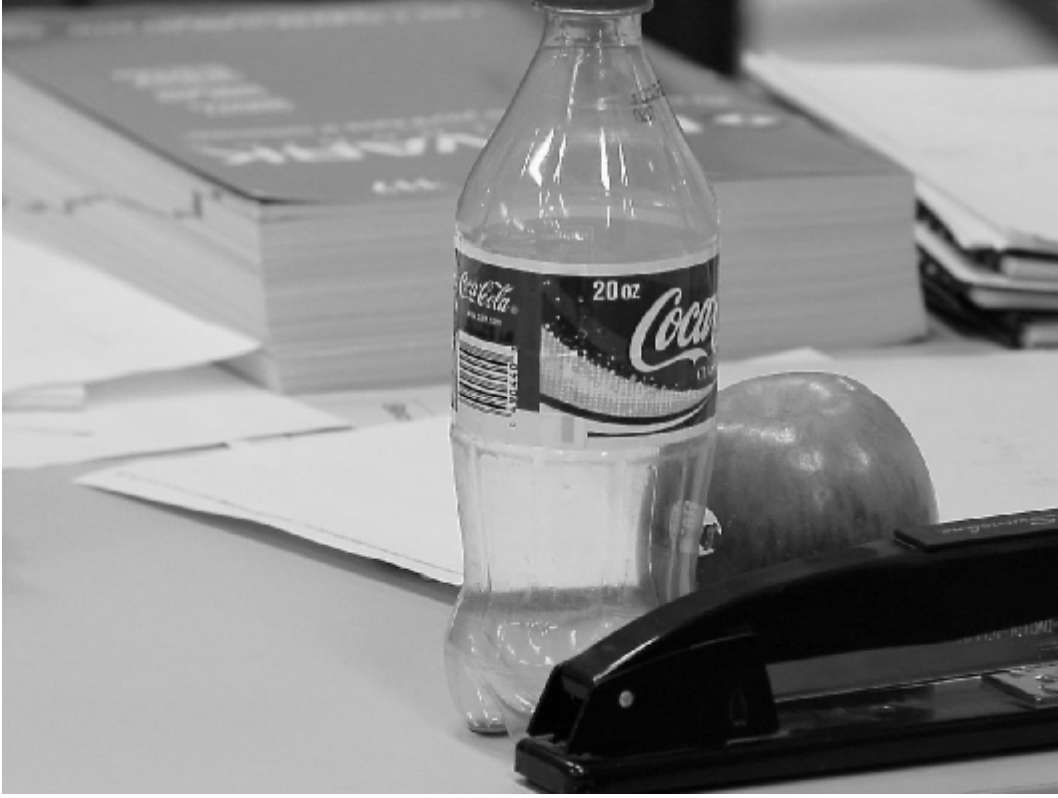
EK 6. Başarı Testlerinde Kullanılan Resimler



Şekil ek.76 1. test resmi



Şekil ek.76 2. test resmi



Şekil ek.76 3. test resmi



Şekil ek.76 4. test resmi



Şekil ek.76 5. test resmi



Şekil ek.76 6. test resmi



Şekil ek.76 7. test resmi



Şekil ek.76 8. test resmi



Şekil ek.76 9. test resmi



Şekil ek.76 10. test resmi



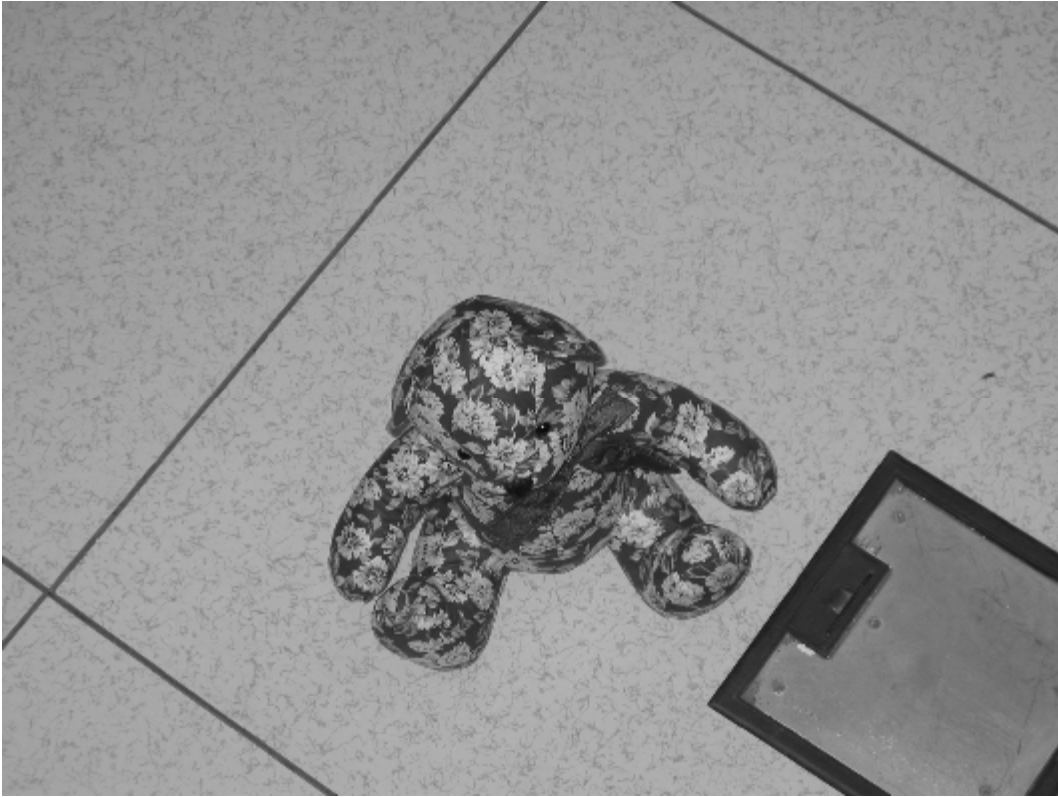
Şekil ek.76 11. test resmi



Şekil ek.76 12. test resmi



Şekil ek.76 13. test resmi



Şekil ek.76 14. test resmi



Şekil ek.76 15. test resmi



Şekil ek.76 16. test resmi



Şekil ek.76 17. test resmi



Şekil ek.76 18. test resmi



Şekil ek.76 19. test resmi



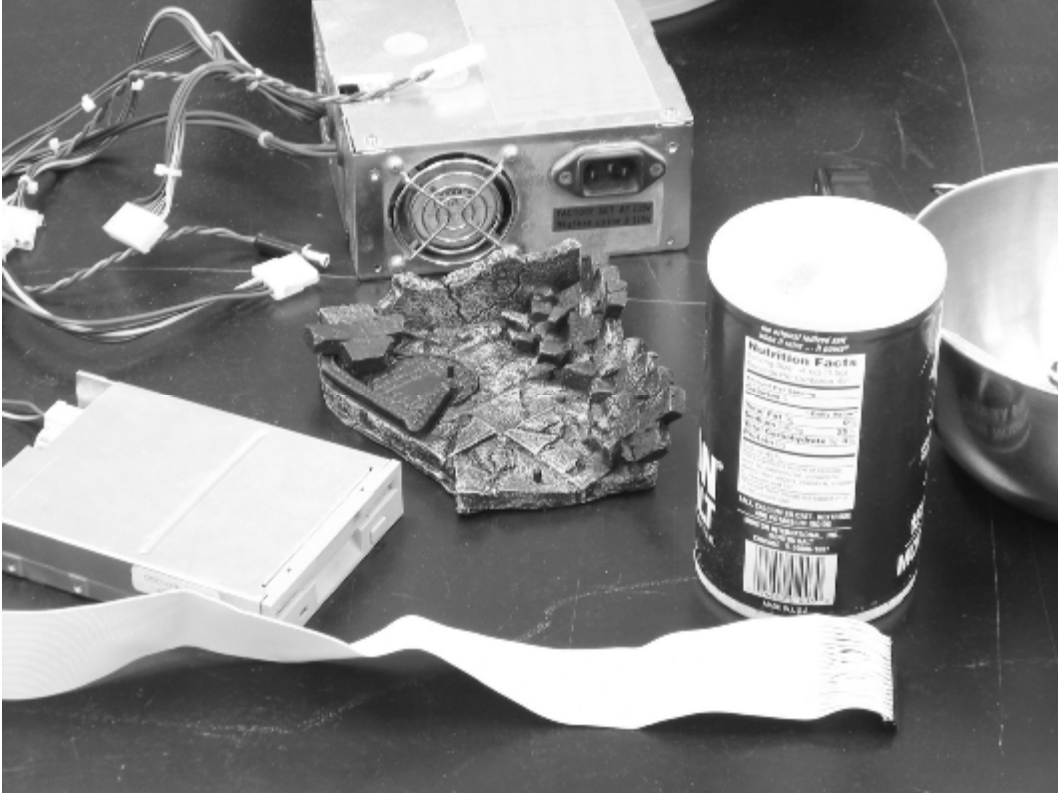
Şekil ek.76 20. test resmi



Şekil ek.76 21. test resmi



Şekil ek.76 22. test resmi



Şekil ek.76 23. test resmi



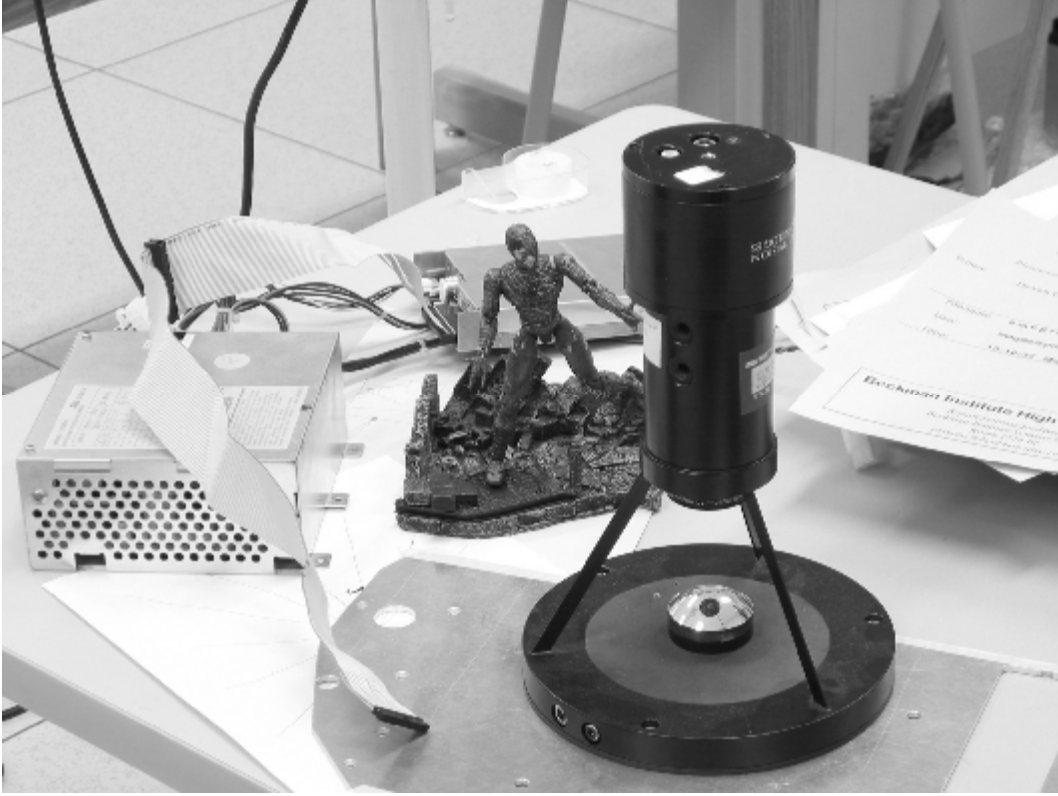
Şekil ek.76 24. test resmi



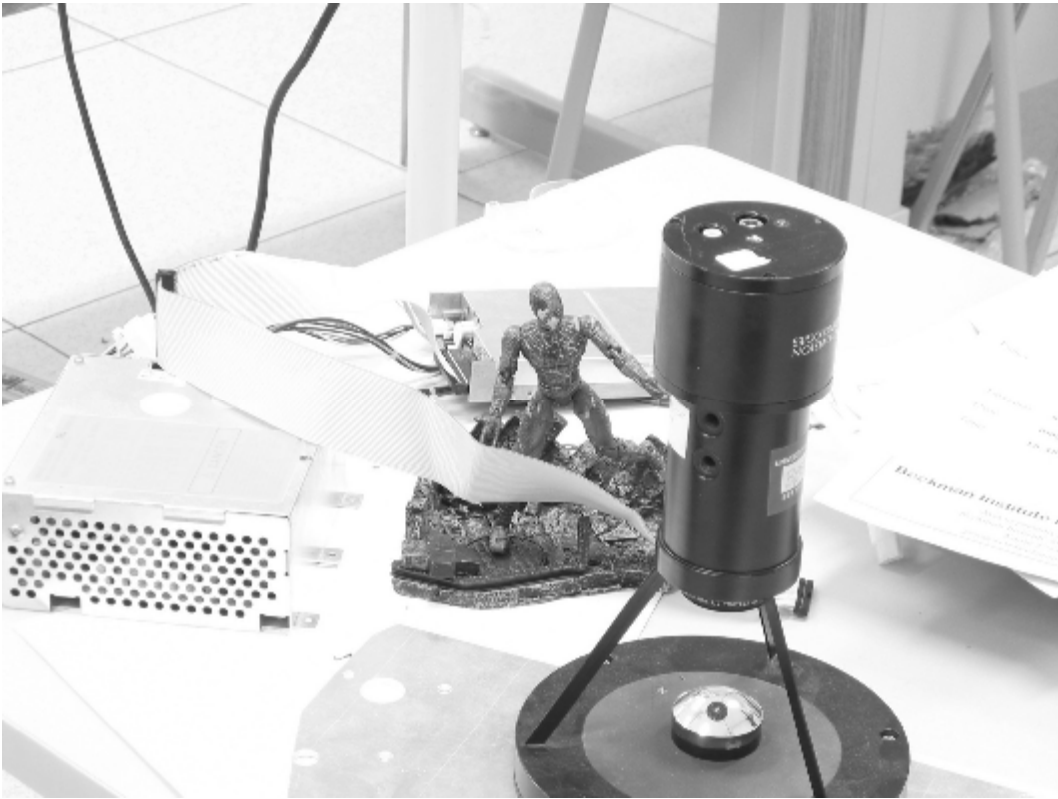
Şekil ek.76 25. test resmi



Şekil ek.76 26. test resmi



Şekil ek.76 27. test resmi



Şekil ek.76 28. test resmi



Şekil ek.76 29. test resmi



Şekil ek.76 30. test resmi



Şekil ek.76 31. test resmi



Şekil ek.76 32. test resmi



Şekil ek.76 33. test resmi



Şekil ek.76 34. test resmi



Şekil ek.76 35. test resmi



Şekil ek.76 36. test resmi



Şekil ek.76 37. test resmi



Şekil ek.76 38. test resmi



Şekil ek.76 39. test resmi



Şekil ek.76 40. test resmi



Şekil ek.76 41. test resmi



Şekil ek.76 42. test resmi



Şekil ek.76 43. test resmi



Şekil ek.76 44. test resmi



Şekil ek.76 45. test resmi



Şekil ek.76 46. test resmi



Şekil ek.76 47. test resmi



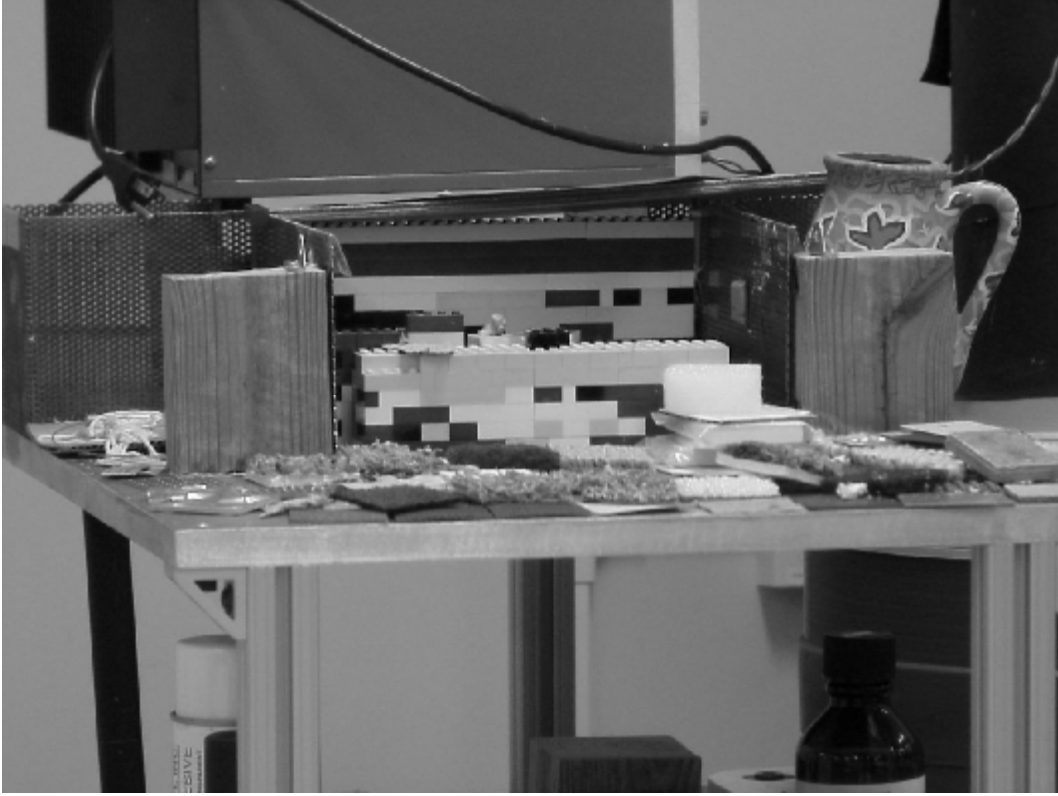
Şekil ek.76 48. test resmi



Şekil ek.76 49. test resmi



Şekil ek.76 50. test resmi



Şekil ek.76 51. test resmi

