



**T.C.  
İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**



**YÜKSEK LİSANS TEZİ**

**LAZER KESİM MAKİNELERİ İÇİN GÖRÜNTÜ İŞLEME  
UYGULAMALARI**

**İsmail PINAR**

**Elektrik Elektronik Mühendisliği Anabilim Dalı**

**Elektrik Elektronik Mühendisliği Programı**

**Danışman**

**Prof. Dr. Aydın AKAN**

**Mart , 2014**

**İSTANBUL**

Bu çalışma 05/03/2014 tarihinde aşağıdaki jüri tarafından Elektrik-Elektronik Mühendisliği Anabilim Dalı Elektrik-Elektronik Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

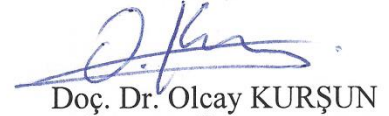
**Tez Jürisi:**

  
Prof. Dr. Aydın AKAN(Danışman)  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Prof. Dr. Mukden UĞUR  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Doç. Dr. Hakan DOĞAN  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Doç. Dr. Fırat KAÇAR  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Doç. Dr. Olcay KURŞUN  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Bu alıřma İstanbul Üniversitesi Bilimsel Arařtırma Projeleri Yürütücü Sekreterliđinin  
XXXXXXXXXX numaralı projesi ile desteklenmiřtir.

## ÖNSÖZ

Lazer Kesim Makineleri İçin Görüntü İşleme Uygulamaları isimli bu çalışma İ.Ü. Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans Programında yüksek lisans tezi olarak hazırlanmıştır.

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca destek ve yardımdan dolayı değerli hocam Prof. Dr. Aydın AKAN' a teşekkürlerimi sunarım.

Tezi yapabilmeme yardımcı olan ve her türlü gerekli desteği sağlayan başta ERMAKSAN A.Ş. Genel Müdürü Sayın Ahmet ÖZKAYAN 'a ve şeflerim Sayın Bekir KOCA ve Sayın Nizam MUZAFFEROĞLU' na Sayın Ünal TENKEKECİ' ye teşekkürü bir borç bilirim.

Mart, 2014

İsmail PINAR

# İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	i
İÇİNDEKİLER .....	ii
ŞEKİL LİSTESİ.....	iii
TABLO LİSTESİ .....	v
SİMGE VE KISALTMA LİSTESİ .....	vii
ÖZET.....	viii
SUMMARY .....	ix
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. GENEL KISIMLAR .....</b>	<b>3</b>
2.1 KÖŞE BULMA YÖNTEMLERİ .....	3
2.1.1 Moravec Yöntemi .....	4
2.1.2 Harris Yöntemi.....	6
2.1.3 Hough Dönüşümü Yöntemi .....	9
2.1.4 Yönlendirilebilir Filtre Yöntemi .....	12
2.1.4.1 Yönlendirilebilir Filtreler ile Görüntü İşleme.....	14
<b>3. MALZEME VE YÖNTEM .....</b>	<b>19</b>
<b>4. BULGULAR .....</b>	<b>22</b>
<b>5. TARTIŞMA VE SONUÇ .....</b>	<b>58</b>
<b>KAYNAKLAR .....</b>	<b>63</b>
<b>EKLER.....</b>	<b>65</b>
<b>ÖZGEÇMİŞ.....</b>	<b>87</b>

## ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Algoritmada kaymalar(solda) ve (1,1) yönündeki kayma(sağda)(Aydoğdu,2010).....	4
Şekil 2.2: Moravec algoritması için değişik test noktaları(D. Parks and J.P. Gravel).....	5
Şekil 2.3: 90° ve 30° döndürülmüş resimlerde Moravec uygulaması(D. Parks and J.P. Gravel)....	6
Şekil 2.4: Özdeğerlerin büyüklüğüne göre pikselin sınıflandırılması(Aydoğdu,2010).....	8
Şekil 2.5: 90 derecelik ve 30 derecelik Resimler Harris Algoritması Uygulanması. ....	9
Şekil 2.6: a) Düz bir doğrunun $\rho$ ve $\theta$ gösterimi, b) Görüntü uzayı, c) Hough uzayı.....	11
Şekil 2.7: Hough Dönüşüm Testi. ....	11
Şekil 2.8: Yönlendirilebilir Filtre Blok Şeması(Albora ve Diğerleri,2003). ....	12
Şekil 2.9: $g^0(x, y)$ için yönlendirmeli süzgeç çıktıları.....	15
Şekil 2.10: $g^{90}(x, y)$ için yönlendirmeli süzgeç çıktıları.....	15
Şekil 2.11: $g^{45}(x, y)$ için yönlendirmeli süzgeç çıktıları. ....	16
Şekil 2.12: $g^{135}(x, y)$ için yönlendirmeli süzgeç çıktıları. ....	16
Şekil 2.13: Kare Deneme Şekli.....	17
Şekil 2.14: 0° ve 90° Filtrelenmiş Kare Deneme Şekli.....	17
Şekil 2.15: 45° ve 135° Filtrelenmiş Kare Deneme Şekli.....	17
Şekil 2.16: Daire Deneme Şekli. ....	18
Şekil 2.17: 0° ve 90° Filtrelenmiş Daire Deneme Şekli. ....	18
Şekil 2.18: 45° ve 135° Filtrelenmiş Daire Deneme Şekli. ....	18
Şekil 3.1: Makinenin Referans için Çekilmiş Boş Resmi.....	19
Şekil 3.2: Kullanılan Deney Sacı ile Çekilmiş Resim .....	19
Şekil 3.3: Program Akış Diyagramı. ....	20
Şekil 4.1: Uygulanacak İlk 6 İmge. ....	22
Şekil 4.2: Uygulanacak Sonraki 8 İmge(İmge7- İmge14).....	23
Şekil 4.3: Uygulanacak Sonraki 8 İmge(İmge15- İmge22).....	24
Şekil 4.4: Uygulanacak Sonraki 8 İmge(İmge23- İmge30).....	25

<b>Şekil 4.5:</b> İlk 8 İmge için Harris Yöntemi Uygulaması(İmge1- İmge8).....	26
<b>Şekil 4.6:</b> Sonraki 8 İmge için Harris Yöntemi Uygulaması(İmge9- İmge16).....	27
<b>Şekil 4.7:</b> Sonraki 8 İmge için Harris Yöntemi Uygulaması(İmge17- İmge24).....	28
<b>Şekil 4.8:</b> Sonraki 6 İmge için Harris Yöntemi Uygulaması(İmge25- İmge30).....	29
<b>Şekil 4.9:</b> İlk 8 İmge için Hough Yöntemi Uygulaması(İmge1 - İmge8).....	34
<b>Şekil 4.10:</b> Sonraki 8 İmge için Hough Yöntemi Uygulaması(İmge9- İmge16).....	35
<b>Şekil 4.11:</b> Sonraki 8 İmge için Hough Yöntemi Uygulaması(İmge17- İmge24).....	36
<b>Şekil 4.12:</b> Sonraki 6 İmge için Hough Yöntemi Uygulaması(İmge25- İmge30).....	37
<b>Şekil 4.13:</b> İlk 6 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge1- İmge6).....	42
<b>Şekil 4.14:</b> Sonraki 8 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge7- İmge14)....	43
<b>Şekil 4.15:</b> Sonraki 8 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge15-İmge22)....	44
<b>Şekil 4.16:</b> Sonraki 8 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge23-İmge30)....	45
<b>Şekil 4.17:</b> İlk 6 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge1-İmge6).....	50
<b>Şekil 4.18:</b> Sonraki 8 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge7-İmge14)....	51
<b>Şekil 4.19:</b> Sonraki 8 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge15-İmge22)...	52
<b>Şekil 4.20:</b> Sonraki 8 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge23-İmge30)...	53

## TABLO LİSTESİ

Sayfa No

<b>Tablo 4.1:</b> Tek Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	30
<b>Tablo 4.2:</b> İki Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	31
<b>Tablo 4.3:</b> Üç Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	32
<b>Tablo 4.4:</b> Dört Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	33
<b>Tablo 4.5:</b> Tek Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	38
<b>Tablo 4.6:</b> İki Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	39
<b>Tablo 4.7:</b> Üç Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	40
<b>Tablo 4. 8:</b> Dört Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	41
<b>Tablo 4.9:</b> Tek Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	46
<b>Tablo 4.10:</b> İki Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	47
<b>Tablo 4.11:</b> Üç Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	48
<b>Tablo 4.12:</b> Dört Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.....	49
<b>Tablo 4.13:</b> Tek Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	54
<b>Tablo 4.14:</b> İki Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	55
<b>Tablo 4.15:</b> Üç Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	56
<b>Tablo 4.16:</b> Dört Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.....	57
<b>Tablo 5.1:</b> 1 mm tolerans payı dikkate alınmış tek köşeli resimler için başarı yüzdeleri .....	58



<b>Tablo 5.2:</b> 1 mm tolerans payı dikkate alınmış iki köşeli resimler için başarı yüzdeleri .....	58
<b>Tablo 5.3:</b> 1 mm tolerans payı dikkate alınmış üç köşeli resimler için başarı yüzdeleri .....	58
<b>Tablo 5.4:</b> 1 mm tolerans payı dikkate alınmış dört köşeli resimler için başarı yüzdeleri .....	58
<b>Tablo 5.5:</b> 1 mm tolerans payı dikkate alınmış tüm resimler için başarı yüzdesi .....	59
<b>Tablo 5.6:</b> 2 mm tolerans payı dikkate alınmış tek köşeli resimler için başarı yüzdeleri .....	59
<b>Tablo 5.7:</b> 2 mm tolerans payı dikkate alınmış iki köşeli resimler için başarı yüzdeleri .....	59
<b>Tablo 5.8:</b> 2 mm tolerans payı dikkate alınmış üç köşeli resimler için başarı yüzdeleri .....	60
<b>Tablo 5.9:</b> 2 mm tolerans payı dikkate alınmış dört köşeli resimler için başarı yüzdeleri .....	60
<b>Tablo 5.10:</b> 2 mm tolerans payı dikkate alınmış tüm resimler için başarı yüzdesi .....	60
<b>Tablo 5.11:</b> Uygulanan yöntemlerin saniye cinsinden uygulama süreleri .....	61

## SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
°	: Derece
$\Sigma$	: Toplama işleci
$\Delta u, \Delta v$	: Birim yönlerdeki kayma
$I(x,y)$	: Resim matrisi
$W$	: Görüntü üzerinde uygulanacak pencere( 3x3, vb)
$\det(M)$	: M matrisinin determinanı
$Tr(M)$	: M matrisinin transpozu
$(\lambda_1, \lambda_2)$	: Matrisin Özdeğerleri
$\delta$	: İmpuls tepki fonksiyonu
$\theta$	: Giriş görüntüsünün dönme açısı
$\rho$	: Giriş görüntüsünün diyagonal uzaklığı
%	: Yüzde işareti
$h^{\theta'}(x, y)$	: $\theta^i$ değerinde dönmüş durumda filtrenin dürtü cevabı
*	: Konvülüsyon
$\frac{\partial}{\partial x}$	: Türev işleci
$\alpha$	: Denklem kesişim açısı

Kısaltmalar	Açıklama
DET	: Baudet Köşe Belirleme İşleci
SUSAN	: En küçük Tek değerli Bölüm Asimile edilmiş Çekirdek
KOÖ	: Köşe Olma Ölçüsü
KKOÖ	: Köşe veya Kenar Olma Ölçüsü
BDS	: Bulunan Denklem Sayısı

## ÖZET

### YÜKSEK LİSANS TEZİ

#### LAZER KESİM MAKİNELERİ İÇİN GÖRÜNTÜ İŞLEME UYGULAMALARI

İsmail PINAR

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Aydın AKAN

Lazer teknolojisindeki son gelişmeler uygulama alanlarını arttırmıştır. Üretimde kaynak, kesme ve delme işlemlerinde en yaygın uygulama alanına sahiptir. Lazerin üretim süreçlerinde kullanılması otomasyonunun kolay entegre edilmesini sağlamakta ve üretim hatalarını azaltmaktadır. Lazer kullanan tezgâhlarla 24 saat üretim yapılabilmekte ve seri üretimi hızlandırdığından maliyet de azalmaktadır. Bu çalışmada seri üretimin daha da hızlandırılması için zaman kayıplarını en aza indirilmesi amaçlanmaktadır.

Makinelerin kesim işlemine başlamasından önce sacın köşe noktasını referans olarak tanıtmak gerekir. Bu referans noktasını şimdiki kullanılan sistemlerde iki farklı şekilde bulunmaktadır. İlk olarak otomatik sistem kullanılır. Bu sistem makinenin lazer kafasının bütün makinenin etrafını dolaşarak kenarları bulmasından ve daha sonra köşe noktasının hesaplamasından meydana gelmektedir. İkinci olarak manuel sistem kullanılır. Bu sistemde ise operatör kesim kafasını düşük hızda el ile köşeye doğru hareket ettirir. Bu yöntemler bir hayli zaman alıcı yöntemlerdir. Bir kesimde yaklaşık 50-60 saniye kayıp meydana gelmektedir. Bu da günde 1200 kesim yapan bir makinenin yaklaşık 2 saat zaman kaybına neden olmaktadır. Bu soruna görüntü işleme uygulamaları ile bir çözüm getirmek hedeflenmiştir.

Tezde kesilecek metalin köşesini bulabilmek için Hough Tekniği, Harris Tekniği ve Yönlendirilebilir Filtre Tekniği olmak üzere üç farklı algoritma geliştirilmiştir. Bu algoritmalarından elde edilen sonuçlar neticesinde Hough teoreminin Harris Teoremine oranla daha başarılı olduğu, en kötü sonuçların Yönlendirilebilir Filtre – Harris yönteminde alındığı belirlenmiştir. Çalışmada birçok bozucu etkiye rastlanmıştır. Işığın fotoğraf çekimlerini etkilemesinden dolayı, görüntülerde kısmen bozulma meydana gelmiştir. Bu yüzden Harris Tekniği kullanılarak birçok resimde köşe doğru bulunamamıştır. Fakat Yönlendirilebilir Filtre – Hough dönüşümü testlerinin tamamı başarılıdır. Elde edilen bulgular sonuçlar kısmında detayları ile anlatılacaktır.

Mart 2014, 96 Sayfa.

**Anahtar Kelimeler:** Görüntü İşleme, Köşe Bulma, Hough Dönüşümü, Harris Dönüşümü, Yönlendirilebilir Filtreler

## **SUMMARY**

**M. Sc. THESIS**

**IMAGE PROCESSING APPLICATIONS FOR LASER CUTTING MACHINES**

**İsmail PINAR**

**İstanbul University**

**Graduate School of Science and Engineering**

**Department of Electrical Electronics Engineering**

**Supervisor: Prof. Dr. Aydın AKAN**

Recent developments in laser technology increased the number of application fields. In manufacturing welding, cutting and hole cutting have the most common application areas. Using laser in manufacturing processes makes the automation to integrate easy and reduces the flaws. Machines that uses laser can produce 24 hours a day and speeds up the mass production that leads to lower costs. In this study, it is intended to speed up the mass production by reducing the time losses.

Before starting to cut a metal sheet, the corner of it should be defined as a reference point. This reference point is defined with two different methods by conventional systems. The first method is automatic. In this method the cutting head moves along the machine sides to find the corners to calculate the reference point. The second one is manual. In this method machine operator moves the head in low speed manually to the corner. This methods are quite time consuming. In one cutting process almost 50-60 seconds are lost. This leads to 2 hours of lost for a machine that cuts 1200 pieces per a day. It is intended to solve this problem with the help of image processing applications.

In this study three algorithms are applied to find the corner. These are Hough Technique, Haaris Technique and Steerable Filter Technique. Results show that Hough Technique is more successful than Harris Technique and Steerable Filter – Harris Technique is the worst to find the corner. In this study there are many disturbance effects are found. Because of light effecting the photo shoots, images are distorted partially. Therefore, using Harris Technique the corner can not be found from most of the images. However, using Steerable Filter – Hough Tachnique all of the corner points are found accurately. Details are explained in the results section.

March 2014, 96 Pages.

**Keywords:** Image Processing, Corner Detection, Hough Transform, Harris Transform, Steerable Filters

## 1. GİRİŞ

Günümüzde lazer teknolojisinin gelişmesi ile metal işleme sektöründe lazer kesim tezgâhları üretilmektedir. Bu tezgâhlarda sac işleme oldukça hızlı ve hassas olarak yapılabilmektedir. Bu tezgâhlar aralıksız olarak çalışmakta günde sayısız işleme yapabilmektedir. Her yeni sac için makine referans almak zorunda ve bu da bir hayli zaman kaybına neden olmaktadır. Referans aldırması manuel ve otomatik olarak iki şekilde yapılmaktadır. Otomatik şekilde referans aldırma esnasında makine ilk önce sac üzerinde kenar bulabilmek için makinenin genişliği kadar gider lazer ışını ve bir sensör sayesinde saca çarpma olup olmadığını tespit etmektedir. Kenar noktaya gelindiğinde sensör geri bildirim alamadığı için ilk kenarı tespit etmekte, aynı şekilde ikinci kenarı aramaya başlamakta ve iki kenarı bulduktan sonra köşeyi bir doğru denklemi sayesinde hesaplamaktadır. Bu hesaplama sonucunda ortaya çıkan noktaya giderek o bölgeyi referans almaktadır. İkinci olarak el ile kontrol şeklinde referans aldırılabilir. Bu yöntemde ise makine düşük hıza alınarak bir yönetme kolu yardımı ile istenilen konuma getirilebilir Işın görünümü yaklaşık 2 mm yarıçapında kesim hassasiyeti ise 50 mikron mertebesindedir. Bu yüzden bu yöntemler hassasiyet bakımından da sorunludurlar. Görüntü işleme teknikleri ile bu soruna çözüm getirilmek istenmiştir.

Görüntülerde köşe bulma, nesnelerin belirgin noktalarının bulunmasından ibarettir. Kenarlar ve köşeler nesnelerin belirlenmesi ve tanınması işlerinde ilk adımlardan biridir. Köşe, iki kenarın kesişimi olarak düşünülebildiği gibi kesintisiz bir kenar üzerindeki değişimi (gradient) değerinin, dar bir komşuluk içinde ani değişim gösterdiği yer olarak da tanımlanabilir. Ayrıca köşeler frekans uzayında belirli frekans bileşenlerinde değişikliklere neden oldukları da bilinmektedir. Diğer bir yaklaşım, görüntü üzerinde çeşitli açı ve piksel değeri açısından derinlikteki köşe şablonlarını aramaya dayanmaktadır. Bu yaklaşımlar köşe bulma algoritmalarının türlerini de özetlemektedir. Köşe bulma konusunda üç genel yaklaşım şöyle özetlenebilir:

1. Kenar bulma algoritmalarıyla, kenarların bulunmasından yararlanarak köşe hesaplama,
2. Gri yoğunluk esasına dayalı arama
3. Çeşitli açı ve büyüklükteki köşelerin şablon arama yöntemiyle bulma

Ancak, hangi yöntem kullanılırsa kullanılsın, bazı durumlarda köşelerin bulunması zorlaşmaktadır. Köşelerin bulunmasında karşılaşılan sorunlar şöyle özetlenebilir:

- Işık değişimi nedeniyle köşelerin resimlerde net algılanamaması
- Işık değişimden kaynaklanan bozulan kenar görüntüsünde değişik köşe noktaları oluşması
- Eğimli bir düzlemdeki bir eğri, bakış açısı nedeniyle yanlışlıkla köşe olarak değerlendirilebilir.

Görüntü üzerinde ne tür nesnelere bulunduğu dair öngörü bulunmadığı durumlarda, köşe bulma algoritmaları beklenen sonuçları vermekten uzaktır. Bazı köşelerin bulunamaması veya köşe olmayan ilgi noktalarının köşe olarak tespit edilmesi, nesne tanıma yöntemlerinde sıkıntı yaratmaktadır. Bu bağlamda, algoritmaların etkin çalışıp çalışmadığı, köşe noktalarının ne kadarının bulunduğu ve kaç tane köşenin yanlışlıkla köşe olarak işaretlediğiyle ölçülebilir. Genel olarak düşünüldüğünde, iyi bir köşe bulma algoritmasının şu koşullara uyması beklenebilir.

- Köşeler iyi algılanmalıdır. Olması gerekenden daha fazla köşenin bulunması gibi durumlar en aza indirilmelidir.
- Köşenin yeri doğru şekilde bulunmalıdır.
- Hızlı çalışmalıdırlar.

Algoritmanın hızının artırılması, sonuçları ters bir şekilde etkileyebilir. Bu nedenle, ilk iki maddede belirtilen kriterlerin iyi olduğu durumlarda, algoritmanın hızının düşünülmemesi yerinde olacaktır. Fakat projedeki asıl amaç hızdır ayrıca hassasiyetimizde bir o kadar önemlidir.

## 2. GENEL KISIMLAR

### 2.1 KÖŞE BULMA YÖNTEMLERİ

Köşelerin yüksek seviyeli görüntü işleme ve nesne tanıma işlerindeki önemleri nedeniyle, köşe bulma konusunda çok sayıda araştırma gerçekleştirilmiştir. İlk araştırmalardan birini yapan Moravec [1] birbirine bitişik iki piksel değeri arasındaki farkın, kenarlarda veya düz alanlarda küçük olduğunu, fakat kenarlarda bütün yönlerde daha büyük bir değer verdiğini gözlemiştir. Bu yöntem kenarlar ve köşeler arasındaki bu farkın kullanılmasına dayanmaktadır.

Chris Harris, Harris [2] algoritması olarak bilinen, Moravec [1] algoritmasının daha gelişmiş bir türü olan bir yöntem önermiştir. Beaudet [3] köşelerin yakınlarında daha yüksek değer verebilen determinanta dayalı DET işlecini geliştirmiştir. Kitchen ve Rosenfeld [4] doğrudan gri seviyeli görüntülerde kullanılan altı yaklaşım önermiştir. Tsai [5] yapay sinir ağlarına dayanan bir yöntem geliştirmiştir. Ji ve Haralick [6] kovaryans dağılımına dayanan bir teknik sunmuştur.

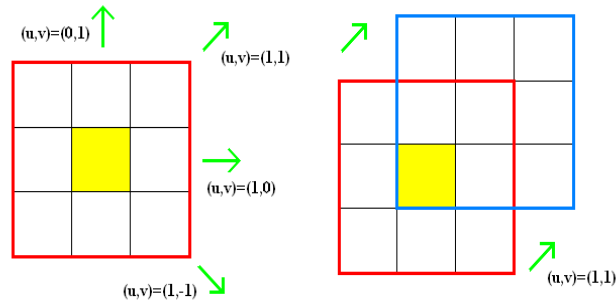
Kitchen ve Rosenfeld'in [4] önerdiği 6 köşe bulma algoritması içinden, gradyan açısının gradyan şiddetine bakılmasına dayanan yöntemeye dayalı bir tekniği, Fang ve Huang [7] önermiştir. Zhang ve Rockett, önerdikleri gri seviyeli köşe modeline dayanarak, köşeleri etiketlemeye yarayan Bayes kararına dayalı bir teknik geliştirmişlerdir [8]. Wu ve Rosenfeld görüntülerin  $x$  ve  $y$  koordinatlarındaki izdüşümlerindeki süreksizliğin incelenerek köşelerin bulunabileceği yaklaşımında bulunmuşlardır[9]. Paler ve diğerleri [10] yerel piksel değerlerinin dağılımını kullanan bir yöntem geliştirmişlerdir. Rangarajan ve diğerleri [11] köşelerde tepkisi artan optimal bir fonksiyon önermiştir. Arrebola ve diğerleri [12] nesnelerin kenarlarından kurulan zincir kodların yerel histogramlarının köşe bulma için kullanılabileceğini öne sürmüşlerdir. Sohn ve diğerleri [13] ortalama alan sertleştirme (mean field annealing) yöntemini köşe bulma için önermişlerdir. Kohlman [14] 2 boyutlu Hilbert dönüşümünü köşe bulmaya uygulamıştır. Mehrotra ve diğerleri [15] kenar ve köşe bulma için Gauss fonksiyonunun birinci ve ikinci yönlü türevlerine dayanan iki yöntem sunmuştur. Davies [16], Hough dönüşümünü köşe bulmaya uygulamıştır. Smith ve Brady [17], SUSAN ismini

verdikleri yöntemle dairesel bir maske kullanarak ve türev hesaplamadan köşelerin bulunabileceği bir yöntem önermişlerdir. Mokhtarian ve Suomela [18], yay evrimi düşüncesini kullanan ve eğrilerin değişik ölçeklerde değerlendirilmesine dayanan başarılı bir yöntem sunmuştur. Ayrıca Mahesh, Dr.M.V.Subramanyam yönlendirilebilir filtreler üzerinden geçirilen resimlere köşe bulma algoritmalarını uygulamayı denemiştir [19].

Yapılan çeşitli çalışmalar, önerilen farklı yaklaşımlar içinde Harris [2] köşe dedektörünün ve Hough [20] dönüşümünün uygulandığı yöntemler diğerlerine göre daha başarılı olduğu söylenebilir. Bu yüzden bu çalışmamızda bu iki yöntem ile yönlendirilebilir filtre yöntemleri denenmiş ve bunlar üzerine gidilmiştir.

### 2.1.1 Moravec Yöntemi

Bu yöntem Hans P. Moravec tarafından 1977 yılında geliştirilmiştir. Moravec [1] algoritmasında, her bir pikselin köşe olup olmadığına karar vermek için o pikseli merkez kabul eden belirli bir büyüklükteki alanın çevresine göre ne kadar farklı olduğuna bakılır. Moravec yoğunluk değişimlerini, küçük arama pencereleri kullanarak hesaplar. (3x3, 5x5, 7x7) Şekil 2.3.'te sarı ile belirtilen pikselin 3x3 büyüklüğünde kırmızı ile işaretlenmiş komşuluğu gösterilmiştir. Bu alanın aralarında 45° fark olan 4 farklı yöne kaydırılması sonucu kapsanan alandaki piksellerin parlaklıkları ile kaydırılmadan önceki alandaki piksellerin parlaklıkları arasındaki fark hesaplanır.



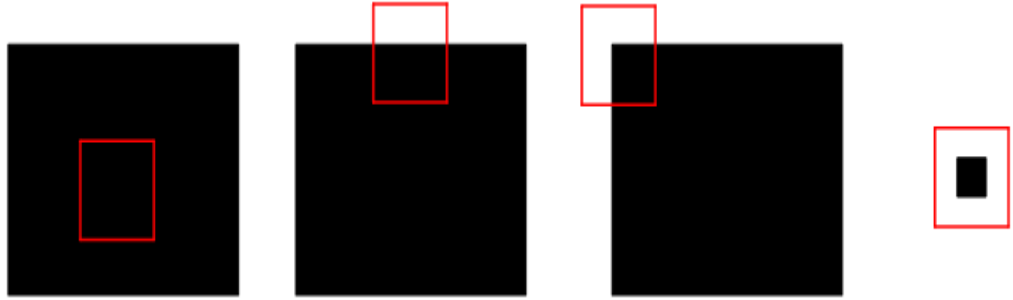
Şekil 2.1: Algoritmada kaymalar(solda) ve (1,1) yönündeki kayma(sağda)(Aydoğdu,2010).

$$F_{u,v}(x, y) = \sum_{m,n} w(m, n) (I(x + m, y + n) - I(x + m + u, y + n + v))^2 \quad (2.1)$$



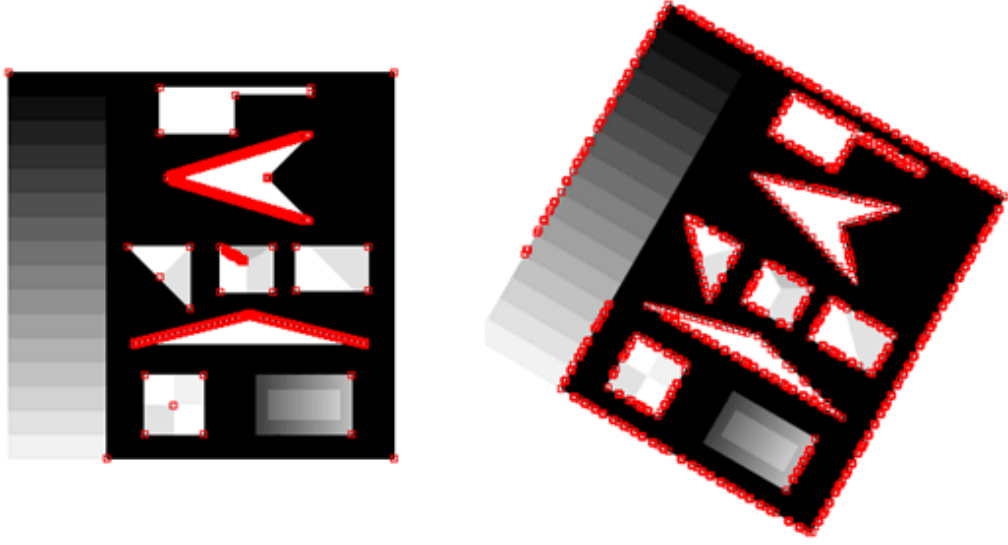
$$KOÖ = \min(F_{0,1}, F_{1,1}, F_{1,0}, F_{1,-1}) \quad (2.2)$$

I gri tonlamalı dijital görüntünün piksellerinin parlaklıklarını göstermek üzere, 2,1 eşitliğinde  $F_{u,v}(x, y)$ ,  $(x, y)$ , merkezli alanın piksellerinin parlaklık değerleri ile alanın  $(u,v)$  yönünde kaydırılması sonucu ulaşılan alandaki piksellerin parlaklıkları arasındaki farkın toplamını göstermektedir. Denklemden  $m$  ve  $n$  ise alanın yatay ve dikeydeki büyüklüğüdür. Algoritmada parlaklık farklarının çarpılacağı katsayıları belirleyen  $w(m,n)$  fonksiyonu alan içinde bir birim iken alan dışında sıfırdır. 2.2 eşitliğinde görüldüğü gibi her piksel için 4 farklı yönde hesaplanan parlaklık fark toplamlarından en küçüğü o piksel için köşe olma ölçüsü olarak kabul edilir. KOÖ'leri deneysel olarak belirlenen bir eşik değerinin üzerindeki pikseller Moravec [1] algoritmasına göre köşe özelliği göstermektedirler. Aşağıdaki şekillerde bir örnek üzerinden anlatılmaktadır.



**Şekil 2.2:** Moravec algoritması için değişik test noktaları(D. Parks and J.P. Gravel).

Soldan 1. şekilde pencere nesnenin içerisinde. Dolayısıyla yoğunluk değeri de sabit olacaktır. Bu noktadan herhangi bir yöne yapılacak kaydırma sonucunda yoğunlukta küçük değişimler gözlenebilir. 2. şekilde kenara dik yönde yapılacak kaydırma büyük değişimlere neden olacak iken, kenar boyunca yapılacak kaydırmalar küçük değişimler oluşturacaktır. 3. ve 4. şekillerde ise her yöne doğru yapılacak olan kaydırma işlemleri sonucunda büyük değişimler gözlenecektir. Bu şekilde 3. ve 4. Şekiller büyük değişimlere sebep olacağından sağlıklı bir köşe bulunamayabilir. Şekil üzerinde bulunan küçük piksel değişimleri köşe algılanabilir. Ayrıca Şekil 90 derece olmadığı takdirde kenarlarda köşe şeklinde algılanabilmektedir. Aşağıdaki şekilde bir örneği verilmiştir.



Şekil 2.3: 90° ve 30° döndürülmüş resimlerde Moravec uygulaması(D. Parks and J.P. Gravel).

### 2.1.2 Harris Yöntemi

Harris [2] Köşe Bulma Yöntemi 1988 yılında Chris Harris ve Mike Stephens tarafından geliştirilmiştir. Genelde Plessey Yöntemi olarak adlandırılır. Özellikle, Harris ve Stephens, tek bir kamera ile alınan görüntülere dayalı ortamı yorumlamak için hareket analiz teknikleri kullanmışlardır. Onlarda ardışık kareleri eşleyerek resim çerçevelerine karşılık gelen kenar ve köşe belirleme yöntemi ile ilgilenmişlerdir. Harris ve Stephens Moravec [1] operatörün eksikliklerini ele alarak bu eksiklikleri gidermeye çalıştılar.

Morevec [1] algoritmasının eksikliklerinden birincisi, Algoritmanın sadece 45°'lik kaymalardaki değişimleri dikkate aldığı için köşe tespitinin yönden bağımsız sonuçlar vermediğidir. Bunun için Harris [2] algoritmasında tüm yönlerdeki değişimlere bakılması gerektiği belirtilmiştir. 4 farklı yöndeki (u,v) kaymaları yerine, kaymalar tüm küçük ( $\Delta u, \Delta v$ ) yönlerini kapsayacak şekilde, 2.3 eşitliğindeki gibi geliştirilmiştir.

$$F_{\Delta u, \Delta v}(x, y) = \sum_{m, n} w(m, n) (I(x + m, y + n) - I(x + m + \Delta u, y + n + \Delta v))^2 \quad (2.3)$$

$I_x$  ve  $I_y$  sırasıyla x ve y yönündeki kısmi türevleri ifade etmek üzere, 2.4 eşitliğindeki gibi Taylor açılımı kullanılarak 2.3 eşitliği 2.5'deki haline dönüştürülür.

$$I(x + m + \Delta u, y + n + \Delta v) \approx I(x + m, y + n) + I_x(x + m, y + n)\Delta x + I_y(x + m, y + n)\Delta y \quad (2.4)$$

$$F_{\Delta u, \Delta v}(x, y) = \sum_{m,n} w(m, n) (I_x(x + m, y + n)\Delta x + I_y(x + m, y + n)\Delta y)^2 \quad (2.5)$$

Harris'in Moravec [1] algoritmasında gösterdiği ikinci eksiklik ise algoritmada kullanılan  $w(m,n)$  fonksiyonunun yalnızca iki değer alması sebebiyle algoritmanın gürültüye karşı dayanıksız olmasıdır. KOÖ hesaplanan pikselin komşuluğu içerisinde kalan tüm piksellerin KOÖ'nü merkeze yakın veya uzakta olmasına bakılmaksızın aynı seviyede etkileme gücüne sahip olmalarından dolayı oluşturacakları gürültüler  $w(m,n)$  fonksiyonunun katsayılarının Gaussian dağılımına uygun olarak seçilmesiyle 2.7 eşitliğinde gösterildiği gibi engellenmiştir.

$$w(m, n) = e^{-\frac{m^2+n^2}{2\sigma^2}} \quad (2.6)$$

$$F_{\Delta u, \Delta v}(x, y) = \sum_{m,n} e^{-\frac{m^2+n^2}{2\sigma^2}} (I_x(x + m, y + n)\Delta x + I_y(x + m, y + n)\Delta y)^2 \quad (2.7)$$

Sadeleştirmelerden sonra 2.8 eşitliği elde edilebilir.

$$F_{\Delta u, \Delta v}(x, y) = \sum_w (I_x\Delta x + I_y\Delta y)^2 \quad (2.8)$$

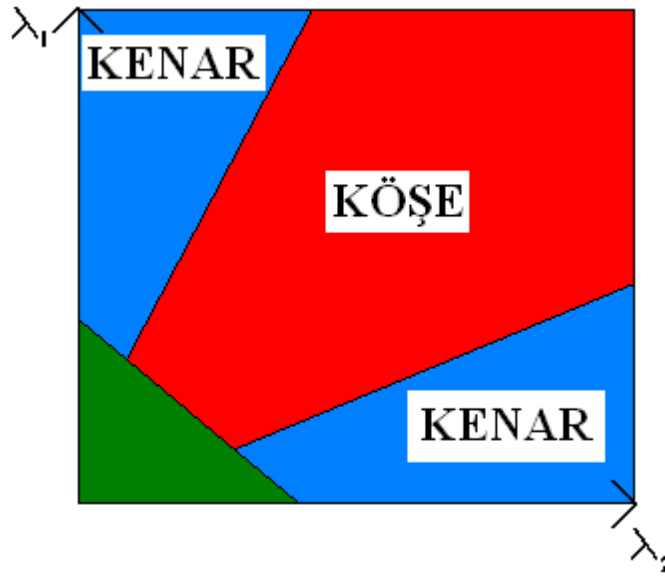
2.8 eşitliği matris formuna dönüştürülürse 2.9 eşitliğine ulaşılır.

$$F_{\Delta u, \Delta v}(x, y) = [\Delta x \ \Delta y] M(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.9)$$

Burada 2x2 boyutlarında olan M matrisi, eşitlik 2.10'da gösterildiği gibi komşuluk sınırları içerisinde kalan piksellerin  $G_{\phi_{xx}}$ ,  $G_{\phi_{yy}}$  ve  $G_{\phi_{xy}}$  gradyan çarpımlarıyla oluşturulmuştur.

$$M(x, y) = \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_w G_{\phi_{xx}} & \sum_w G_{\phi_{xy}} \\ \sum_w G_{\phi_{xy}} & \sum_w G_{\phi_{yy}} \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (2.10)$$

Harris tarafından diagonal olan M matrisinin özdeğerlerinin yerel oto korelasyon fonksiyonunun kavışılması ile orantılı olduğunu belirtilmiştir. Şekil 2.4.'te görüldüğü gibi matrisin özdeğerlerinden ( $\lambda_1, \lambda_2$ ) ikisinin de belirli bir eşik değerin üzerinde olmaları o pikselin köşe olduğuna, bir özdeğerin diğerine göre çok büyük olmasının ise pikselin kenar olduğuna işaret etmektedir. Özdeğerlerden ikisi de belirli bir büyüklüğün altında ise o pikselin köşe veya kenar özelliği göstermemektedir. Böylece Moravec [1] algoritmasındaki 3.sorun olan kenar ve köşe noktaların yakın KKOÖ'lerine sahip olması sorununu çözülmüştür.



Şekil 2.4: Özdeğerlerin büyüklüğüne göre pikselin sınıflandırılması(Aydoğdu,2010).

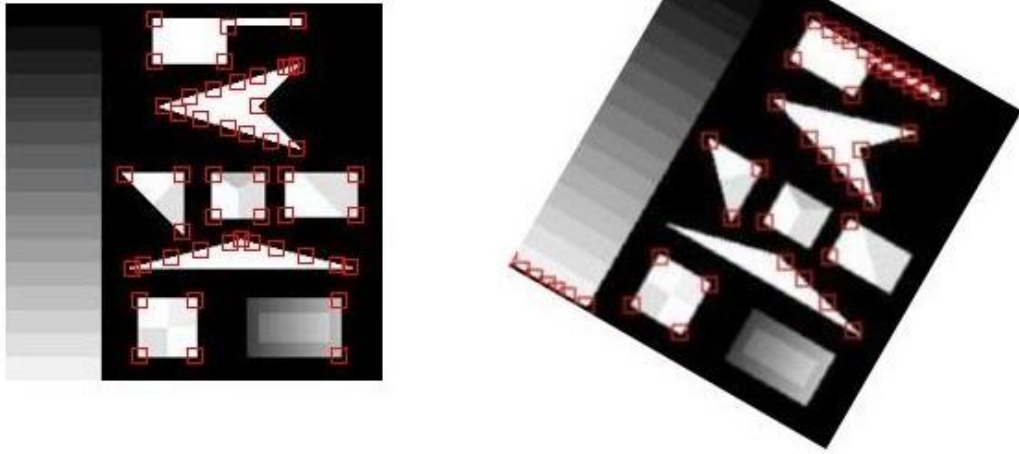
M matrisinin özdeğerlerinin hesaplanmasının fazla zaman almasından daha kolay hesaplanabilen bir köşe veya kenar olma ölçüsü önerilmiştir.

$$KKOÖ = \det(M) - k.Tr(M)^2 = AB - C^2 - k(A + B)^2 \quad (2.11)$$

Eşitlikteki k değiştirilebilen bir sabit olmak üzere; elde edilen KKOÖ pozitif ve belirli bir büyüklükte ise piksel köşeye, negatif ve belirli bir büyüklükte ise kenara karşılık gelmektedir. Genellikle uygulamalarda 0,04 ile 0,06 arasında seçilen k sabiti projede 0,04 olarak kabul edilmiştir.

Aşağıda Harris [2] algoritması orijinal ve 30 derece döndürülmüş 2 farklı resme uygulanmıştır. Görüldüğü üzere bulunan köşeler resmin döndürülmesi ile çok fazla şekilde değişmemiştir. Fakat aynı resim Moravec [1] algoritmasına uygulandığı zaman gereksiz birçok kenarı da köşe olarak da işaretlemiştir. Buradan da görüldüğü üzere Harris [2] teoremi yönden bağımsız hale gelmiştir.

Bu projede ölçülen saclar genelde açılı olmaktadır. Bu yüzden Moravec [1] algoritması yetersiz kalmaktadır.



Şekil 2.5: 90 derecelik ve 30 derecelik Resimler Harris Algoritması Uygulanması.

### 2.1.3 Hough Dönüşümü Yöntemi

Hough [20] dönüşümü, 1962 yılında Paul Hough tarafından tasarlanmış ve sonraları pek çok yazar tarafından geliştirilerek farklı görüntü işleme uygulamalarında kullanılmıştır. Hough [20] Dönüşümü düz ve elips çizgilerin tespiti için son derece başarılı bir yöntemdir. Temel olarak siyah beyaz görüntülerdeki düzgün doğruları tespit etmek için geliştirilmiştir. Görüntüden yakalanabilecek olan şekiller arasında, düzgün doğrular, eğriler, çemberler ve elipslere ek olarak doğrusal olmayan keyfi dağılımlı düzensiz şekiller de yer almaktadır. Elbette bunların hepsi için ayrı parametreler ve denklemler geliştirilmiştir. Hough [20] dönüşümü yöntemiyle bir sayısal görüntüden doğruları yakalamak istediğinizde, görüntünün içindeki doğrusal olmayan daire ve eğri

gibi şekiller, dönüşüm sonucu oluşacak olan yeni görüntüde yer almayacaklardır. Çıktı görüntüsünde sadece düzgün doğrulara yer verilecektir.

Hough [20] fonksiyonu, görüntüdeki sıfırdan farklı her bir noktayı (piksel) parametre uzayında bir sinusoide dönüştürür. Ters durumunda ise, parametre uzayında her bir nokta görüntüde bir düz doğruya karşılık gelir. Klasik dönüşüm, giriş görüntüsündeki doğruları saptamaya yöneliktir. Bir doğrunun parametre uzayında tanımı, Duda ve Hart [21] tarafından 1972 yılında tanımlanmış ve görüntünün parametre uzayında çıktısı ayrık veri durumunda aşağıda gösterildiği gibi verilebilir.

$$H(\rho, \theta) = \sum_m \sum_n f(m, n) \delta(\rho - m\Delta x \cos\theta - n\Delta y \sin\theta) \quad (2.12)$$

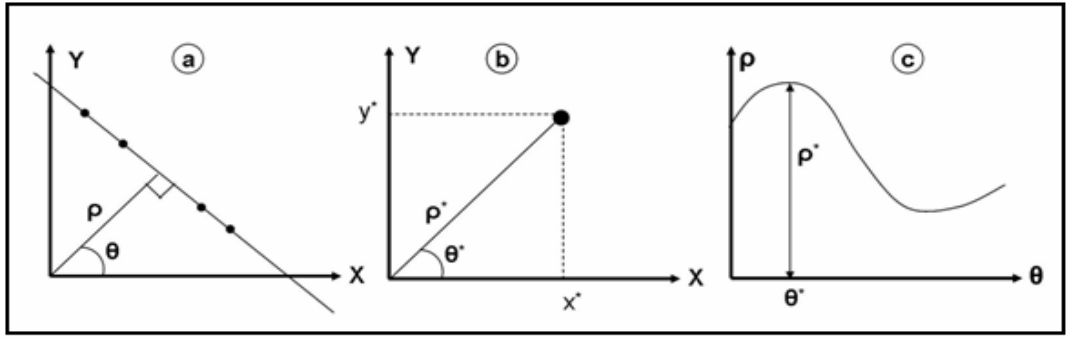
Burada,  $f(m, n)$ , sırası ile, x,y yönünde  $\Delta x$  ve  $\Delta y$  aralıklara sahip  $m \times n$  boyutunda ikili (binary) giriş görüntüsü,  $\delta$  ise impuls tepki fonksiyonu olarak tanımlanır. Yukarıda verilen bağıntıdaki impuls tepki fonksiyonu, giriş görüntüsündeki her bir noktanın,

$$\rho = m\Delta x \cos\theta + n\Delta y \sin\theta \quad (2.13)$$

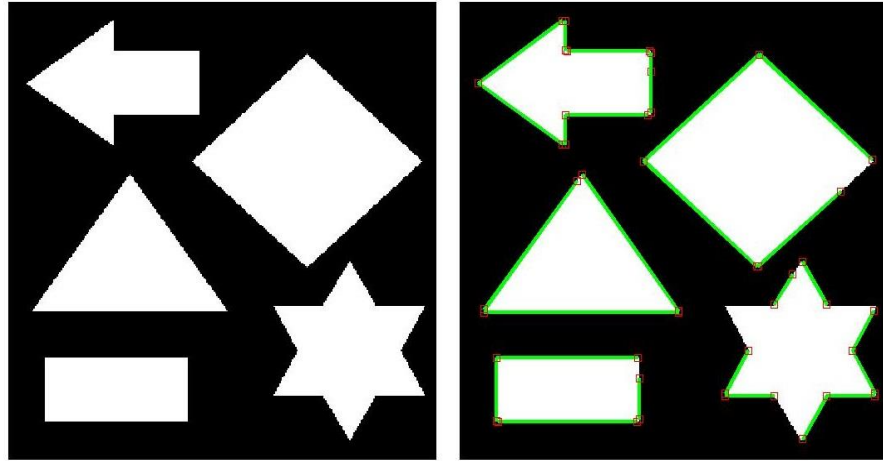
ifadesi ile parametre uzayında bir sinusoide dönüştürülmesine olanak sağlar.

Yukarıda verilen 2.13 bağıntısında,  $\rho(-D, D)$  olup Şekil 2.6a' da gösterildiği üzere, (D, giriş görüntüsünün diyagonal uzaklığını simgeler), doğruya dik bir vektör boyunca orijin noktasından doğruya kadar olan uzaklık olarak alınır. Burada,  $\theta(0, \pi)$  olup, x eksenini ile  $\rho$  vektörü arasındaki açıyı göstermektedir (Şekil 2.6a). Bir nokta kaynağa ait görüntü ve ilgili parametreler Şekil 2.6b' de gösterilmişlerdir. Hough sonuçları akümülatör, Hough [20] dönüşüm matrisi, olarak tanımlanan bir matriste ( $H(\rho, \theta)$ ) toplanırlar. Matrisin bir boyutunu  $\rho$  (uzaklık) değerleri, diğer boyutunu ise  $\theta$  (açı) değerleri oluşturur. Parametre uzayında (Hough uzayı) eğrinin maksimum olduğu nokta değerleri görüntü uzayındaki kaynak noktaya karşılık gelmektedir (Şekil 2.6c). Giriş görüntüsündeki aynı doğru üzerine düşen tüm noktalar, Hough [20] uzayında (parametre uzayı) tek bir noktada kesişen sinusoidleri oluşturur. Bu süreç, kenarları saptanmış bir giriş görüntüsündeki tüm noktalar için tekrarlanır. Hough [20] dönüşüm matrisinin her bir elemanı,  $(\rho, \theta)$  parametreleri ile bir doğru üzerinde kaç adet nokta

olduğunu belirler. Matrisin en büyük değerli elemanı, giriş görüntüsünde hangi doğrunun en çok temsil edildiğini gösterir. Matrisin kesişim noktalarında oluşan maksimumlar giriş görüntüsündeki doğruları gösteren hakim parametre değerleridir. Görüntü ortamındaki bir nokta, parametre uzayında sinusoidal bir eğriye karşılık gelmektedir. Görüntü uzayında aynı bir düz doğru üzerindeki noktalar, parametre uzayında aynı noktada kesişen eğrilere karşılık gelirler. Parametre uzayında aynı eğri üzerinde bulunan noktalar, görüntü uzayında aynı noktadan geçen doğrulara karşılık gelmektedirler.



Şekil 2.6: a) Düz bir doğrunun  $\rho$  ve  $\theta$  gösterimi, b) Görüntü uzayı, c) Hough uzayı.



Şekil 2.7: Hough Dönüşüm Testi.

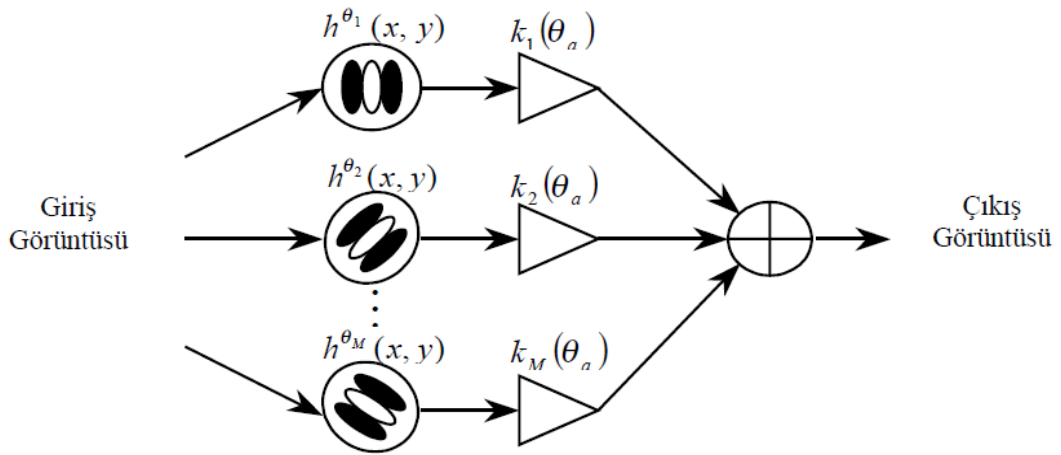
Şekilde görüldüğü gibi bir test resmine Hough [20] dönüşümü uygulanmış ve doğruların yaklaşık %90 yakalanmıştır. Fakat dikkat edilmelidir ki doğruların başlangıç ve bitiş noktaları tamamen doğru değildir. Bazı doğrular yarıda bitmiş köşe noktaya kadar gidememiştir. Bu da projede istenileni vermekten uzaktır. Ayrıca bazı doğrular 2

yaklaşık doğru ile gösterilmiştir. Bu da tezatlık oluşturmaktadır. Bu test resmindeki doğrular tamamen düz doğrular olup gürültüden uzaktır. Gürültü oluştuğunda doğru sayıları çoğalmakta istenilen sonuç bulunamamaktadır. Bunun için bu projede Hough [20] dönüşümüne ilaveten doğruların kesim noktaları aranmış ve kesim açıları kontrol edilmiştir. Bu şekilde başarı istenilen düzeye getirilmiş projede uygulanabilir bir hal almıştır.

Ayrıca Hough [20] dönüşümünü 2 renk üzerinden uygulayabildiğimiz için renk geçişlerinde, örneğin ışık yüzünden hafif renk açılması olduğu zaman diğer algoritmalar buraları köşe noktası olarak algılamakta fakat Hough [20] dönüşümü burayı köşe kabul etmemektedir. Bu durum bizim için artı bir yön olmaktadır.

#### 2.1.4 Yönlendirilebilir Filtre Yöntemi

Kenar belirleme, görüntü sıkıştırma ve iyileştirme, doku analizi gibi birçok görüntü işleme konularında uygulamaları bulunan yönlendirilmeli filtreler temel olarak belirli bir doğrultuda bant geçiren filtre özelliğini gösterirler. Bir görüntüdeki farklı yönelimdeki kenarlar, görüntünün farklı yönelimlere sahip temel filtrelerden geçirilip, yönelim alt bantlarına ayrılmasıyla elde edilebilir.



Şekil 2.8: Yönlendirilebilir Filtre Blok Şeması (Albora ve Diğerleri,2003).

Şekil 2.8'de bu işlemi gerçekleştiren yönlendirilmeli filtre blok şeması verilmiştir. Burada  $h^{\theta_1}(x, y)$ ,  $h^{\theta_2}(x, y)$ ,  $h^{\theta_m}(x, y)$ ,  $h(x, y)$  'nin  $\theta_i$ ,  $0 \leq i \leq M$ , değerlerinde dönmüş durumlarına karşılık düşen filtrelerin dürtü cevaplarıdır. Giriş



görüntüsü bu filtreler kullanılarak  $M$  alt banda ayrılır ve böylece filtrelerin yönelimlerine karşılık düşen yönelimdeki kenarlar belirlenir.

Bir  $h(x, y)$  fonksiyonunun gelişigüzel bir dönme değerindeki ifadesi, sabit dönme değerlerindeki ifadelerinin lineer bir kombinasyonu biçiminde yazılabiliyorsa bu fonksiyon yönlendirilmeli fonksiyon olarak adlandırılır. Böyle bir fonksiyonun matematiksel ifadesi aşağıdaki eşitlikte verilmiştir.

$$h^{\theta_a}(x, y) = \sum_{i=1}^M k_i(\theta_a) h^{\theta_i}(x, y) \quad (2.14)$$

$h^{\theta_a}(x, y)$  ,  $h(x, y)$  ' nin  $\theta_a$  kadar dönmüş durumuna karşılık düşer.  $k_i(\theta_a)$  ,  $0 \leq i \leq M$  , filtre yönelimlerini kontrol eden ara değerlendirme fonksiyonlarıdır. Ara değerlendirme fonksiyonları

$$e^{jl\theta_a} = \sum_{i=1}^M e^{jl\theta_i} k_i(\theta_a) \quad , 0 \leq l \leq N \quad (2.15)$$

koşulunu sağlıyorsa, yönlendirme fonksiyonları aşağıdaki gibi Fourier Serisine açılabilir.

$$h(r, \theta) = \sum_{n=-N}^N e^{jn\theta_a} a_n(r) \quad , 0 \leq l \leq N \quad (2.16)$$

Burada  $r = \sqrt{x^2 + y^2}$  ve  $\theta = \arg(x, y)$  'dir. Temel fonksiyonlar ve ara değerlendirme fonksiyonları belirlendikten sonra  $h(x, y)$  'nin  $\theta_a$  kadar dönmüş durumuna karşılık düşen  $h^{\theta_a}(x, y)$  , (1) nolu eşitlik kullanılarak bulunabilir. Böylece bir görüntüdeki  $\theta_a$  yönelimindeki kenarların belirlenmesi, bu görüntünün  $h^{\theta_i}(x, y)$  filtrelerinden geçirilmesiyle sağlanmış olur.

$\theta_a$  açılı yönlendirilmiş bant geçiren filtrelerin, bu doğrultusundaki enerjinin hesaplanması ayrıştırma işlevinin başarımını göstermek için önemli bir parametredir. Yönlendirilmiş enerji fonksiyonunun hesaplanması için, (2.17) nolu denklem verilmektedir.

$$E^{\theta_a} = [f(x, y) * h^{\theta_a}(x, y)]^2 + [f(x, y) * g^{\theta_a}(x, y)]^2 \quad (2.17)$$

Bu eşitlikte ‘\*’ konvolüsyon işlemini gösterir.  $f(x, y)$  giriş görüntüsü ve  $g^{\theta_a}(x, y)$  ise  $\theta_a$  açısındaki yönlendirmeli süzgeç dürtü yanıtı  $h^{\theta_a}(x, y)$  ifadesinin Hilbert dönüşümüdür. Bir görüntünün yönelimi, (2.17) nolu denklem ile bulunur. Her bir açı değerine karşılık gelen enerjilerin toplamı ile o görüntüye ait açı-enerji histogramını oluşturulur.[22]

#### 2.1.4.1 Yönlendirilebilir Filtreler ile Görüntü İşleme

Bir görüntü yönlendirilebilir bir filtreden geçirildikten sonra yönlendirilebilir filtre hangi açı değeri için seçilmiş ise görüntü üzerindeki o açı değerlerindeki kenarlar baskın olacak şekilde ortaya çıkarken bu açıdan uzaklaşan açı değerlerindeki yönelimler ise zayıflayarak veya tamamen yok olarak kendilerini göstermektedirler. Burada yönlendirilebilir filtre olarak  $h(x, y) = e^{-(x^2+y^2)/2}$  şeklindeki iki boyutlu Gaussian fonksiyonun birinci türevi kullanılmıştır.[22]. Aynı fonksiyonun, 90 derece dönmüş hali de aşağıda gösterilmiştir.

$$h^{0^\circ}(x, y) = \frac{\partial}{\partial x} e^{-(x^2+y^2)/2} = -xe^{-(x^2+y^2)/2} \quad (2.18)$$

$$h^{90^\circ}(x, y) = \frac{\partial}{\partial y} e^{-(x^2+y^2)/2} = -ye^{-(x^2+y^2)/2} \quad (2.19)$$

Buradan bu fonksiyonun herhangi bir açıdaki dönme değeri ara değerlendirme fonksiyonları da hesaba katılarak yazılırsa aşağıdaki denklem edilir. Bu yönlendirilebilir bir fonksiyondur.

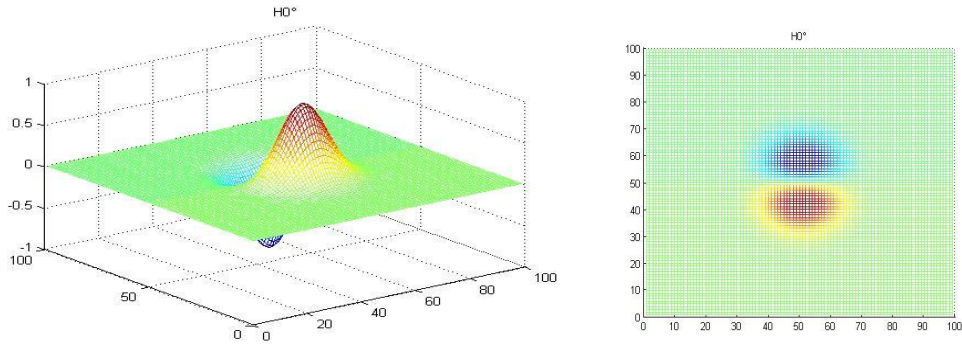
$$h^{\theta^\circ}(x, y) = \cos(\theta) h^{0^\circ}(x, y) + \sin(\theta) h^{90^\circ}(x, y) \quad (2.20)$$

Burada herhangi bir açı değerindeki dönmeye karşılık gelen  $h^{\theta^\circ}(x, y)$  fonksiyonu kendisinin  $h^{0^\circ}(x, y)$  ve  $h^{90^\circ}(x, y)$  şeklinde verilen sırasıyla 0 ve 90 derece dönme değerlerinin lineer bir kombinasyonundan oluşmaktadır. Şekil 2.11 ve 2.12 de  $\theta_a = 45$  ve 135 derece değerlerindeki yönlendirmeli süzgeç çıktıları incelenmiştir. (2.20) nolu denklem  $\theta_a = 45$  ve 135 derece için tekrar yazılırsa,

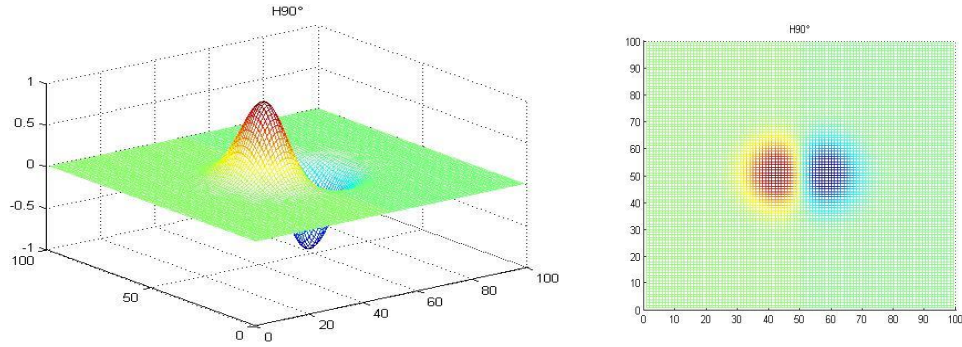
$$h^{45^\circ}(x, y) = \cos(45) h^0(x, y) + \sin(45) h^{90^\circ}(x, y) \quad (2.21)$$

$$h^{135^\circ}(x, y) = \cos(135) h^0(x, y) + \sin(135) h^{90^\circ}(x, y) \quad (2.22)$$

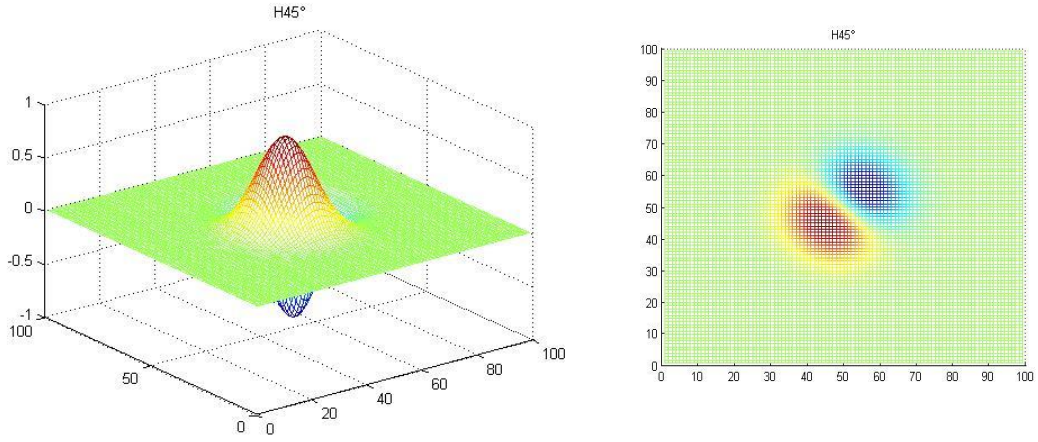
elde edilir. Baskın kenarları belirlenecek olan görüntü, hangi açı değerindeki baskın kenarlar ortaya çıkarılmak isteniyorsa o açı değerine ait yönlendirilebilir filtreden geçirilerek elde edilir. Şekil 2.9 ve 2.10 'da 0 ve 90 derecelerdeki dönme değerleri ve Şekil 2.11 ve 2.12 de bu dönme değerlerinin lineer kombinasyonundan elde edilen 45 ve 135 derecelerdeki dönme değerleri (2.21) ve (2.22) nolu denlemler ile elde edilmiştir.



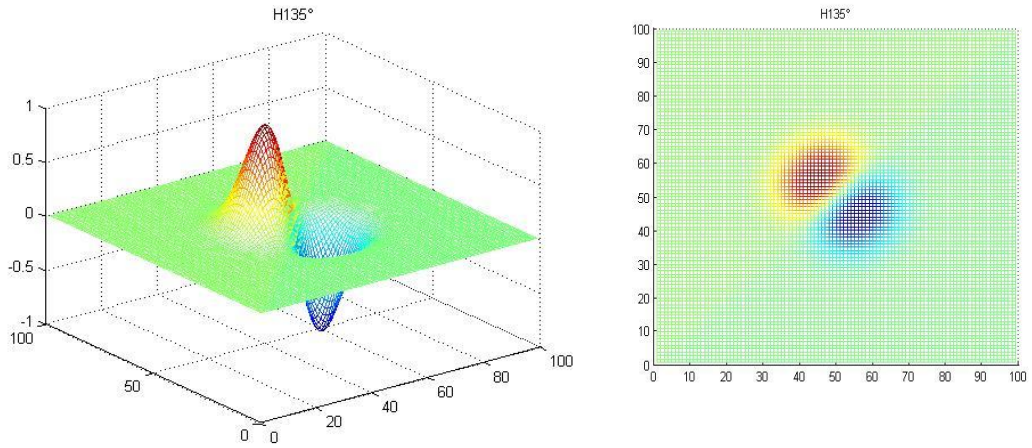
Şekil 2.9:  $g^0(x, y)$  için yönlendirmeli süzgeç çıktıları.



Şekil 2.10:  $g^{90}(x, y)$  için yönlendirmeli süzgeç çıktıları.

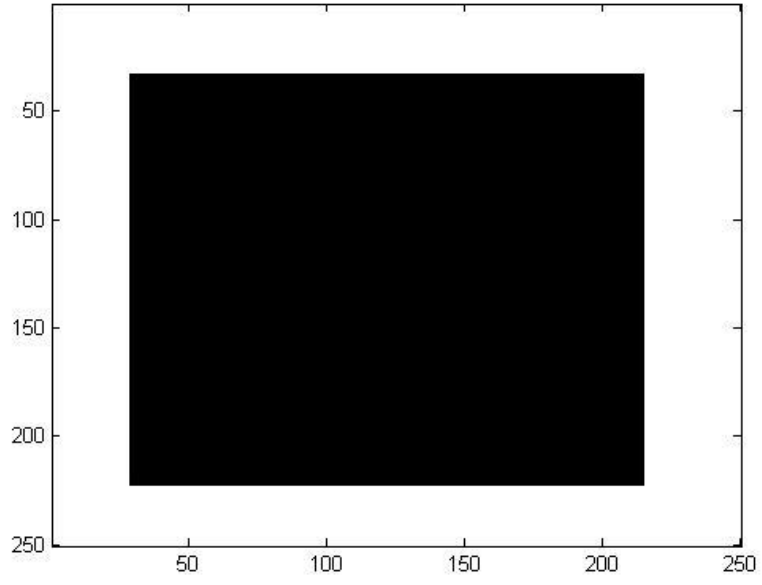


Şekil 2.11:  $g^{45^\circ}(x, y)$  için yönlendirmeli süzgeç çıktıları.

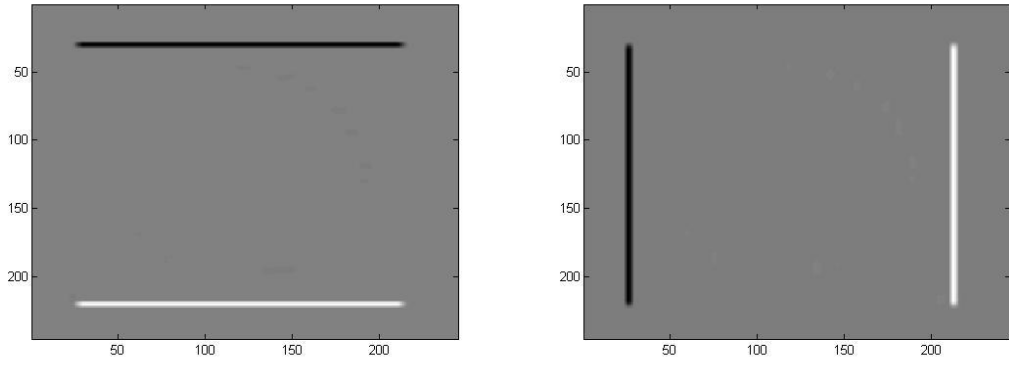


Şekil 2.12:  $g^{135^\circ}(x, y)$  için yönlendirmeli süzgeç çıktıları.

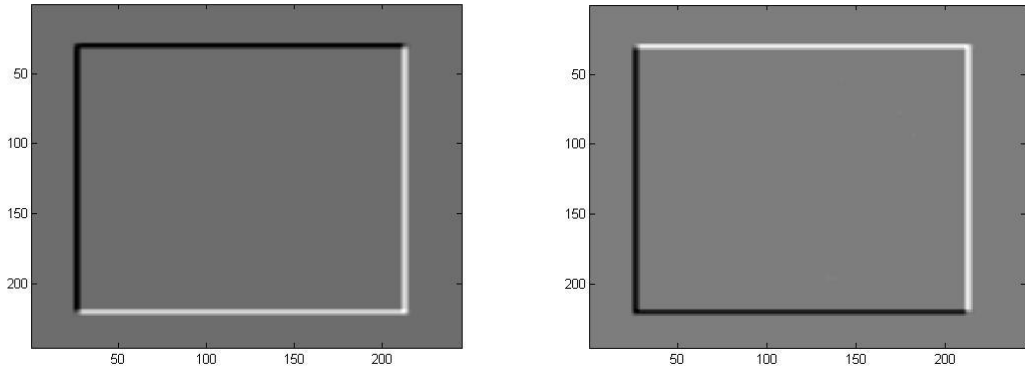
Şekil 2.13'de ise bir dairenin ve bir karenin 45 derecelik bir yönlendirilebilir filtreden geçtikten sonraki durumu görülmektedir. Bu şekiller üzerinde dikkat edilecek olursa sadece baskın kenarların olduğu kenarlar (burada 45 derecelik eğime sahip olan kenarlar) ortaya çıkarken, diğer açı değerlerine sahip olan kenarlar daha zayıf ve belirsiz olarak gözükmeaktedirler. Şekillerde gözüktüğü üzere 135 derece olmayan şekil 135 dereceden geçirildiği takdirde 90 derece ve 180 derece arası değerler filtreden geçmektedir. Bu da bu proje için olumsuzluk oluşturmaktadır.



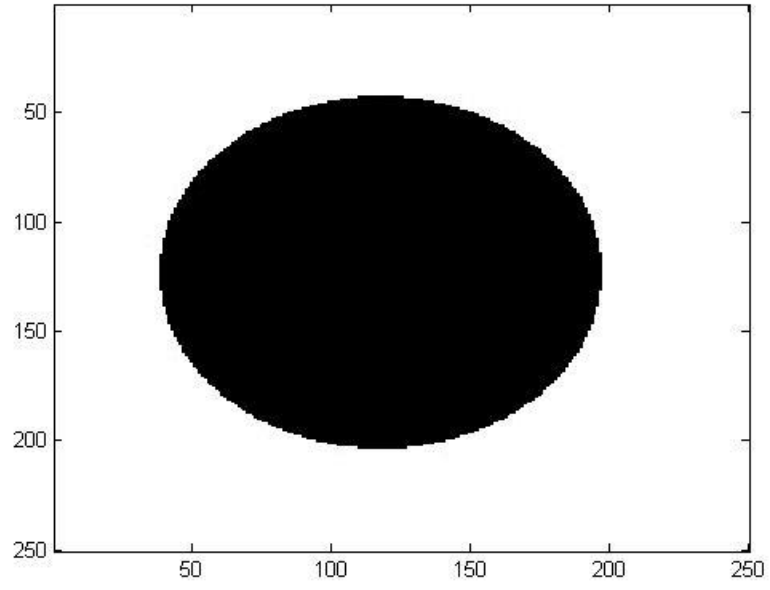
Şekil 2.13: Kare Deneme Şekli.



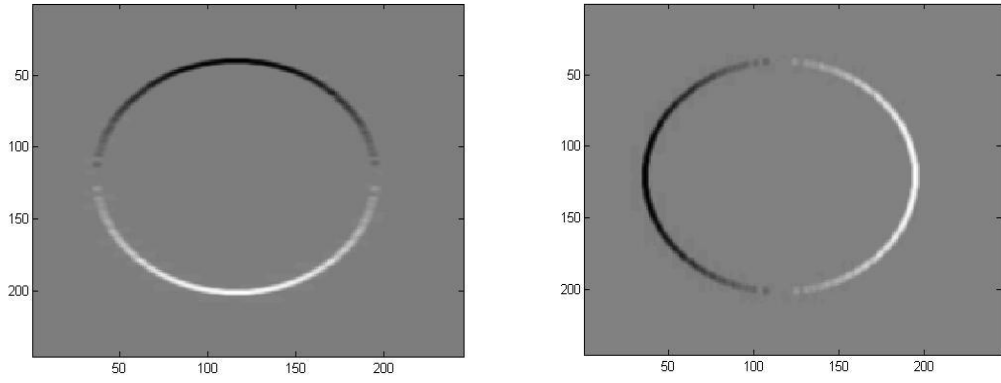
Şekil 2.14:  $0^\circ$  ve  $90^\circ$  Filtrelenmiş Kare Deneme Şekli.



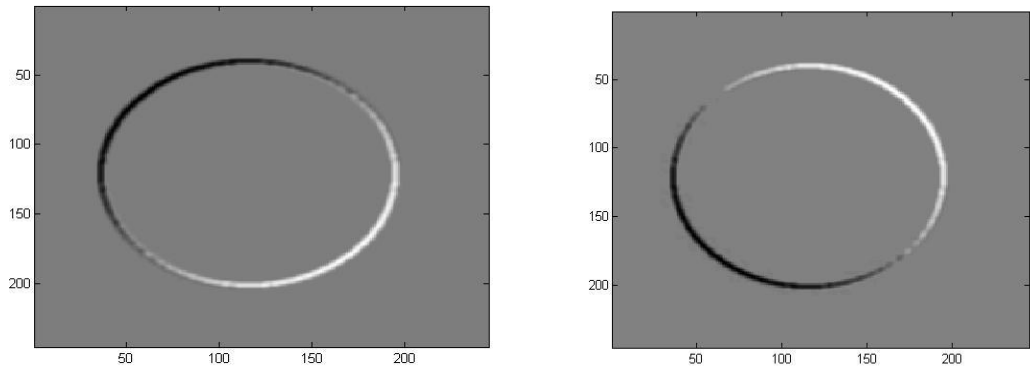
Şekil 2.15:  $45^\circ$  ve  $135^\circ$  Filtrelenmiş Kare Deneme Şekli.



Şekil 2.16: Daire Deneme Şekli.



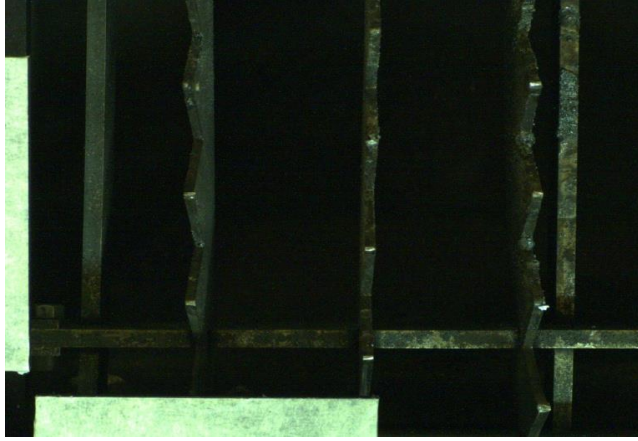
Şekil 2.17:  $0^\circ$  ve  $90^\circ$  Filtrenmiş Daire Deneme Şekli.



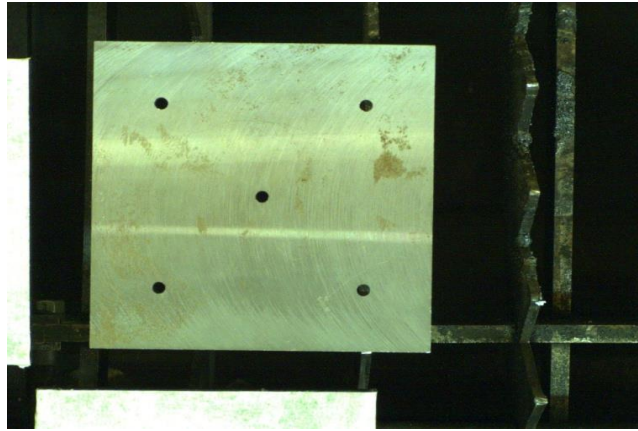
Şekil 2.18:  $45^\circ$  ve  $135^\circ$  Filtrenmiş Daire Deneme Şekli.

### 3. MALZEME VE YÖNTEM

Bu projede kullanılan malzemeler sac kesme makinesi, endüstriyel kamera (5 MP), hesaplamak için PC, sonuçların gözlenmesi için makine kontrolörü (endüstriyel PC) ve 10 x 10 cm deney sacıdır. Kameranın görebildiği bölge (2560x1920) yaklaşık 18.4 cm ve 13.8 cm' dir. Bu yüzden 10 cm kenara sahip kare deney sacı kullanılmıştır. Bu sac değişik açılarda değişik noktalara yerleştirilip denemeler yapılmıştır. Aşağıdaki şekillerde görüldüğü üzere makinenin üst ve yan kenarlarında 10 cm 'lik referans sacları bulunmaktadır. Bu sacların amacı 10 santimetreye düşen piksel sayısını algılayabilmektir. Ayrıca bu saclar normalde makinede bulunmaktadır. Kullanıcı sacı tam iterek dayaması için destek görevleri görebilmektedir. Sac şekillerde görüldüğü üzere ızgaraların üzerine konulmaktadır.

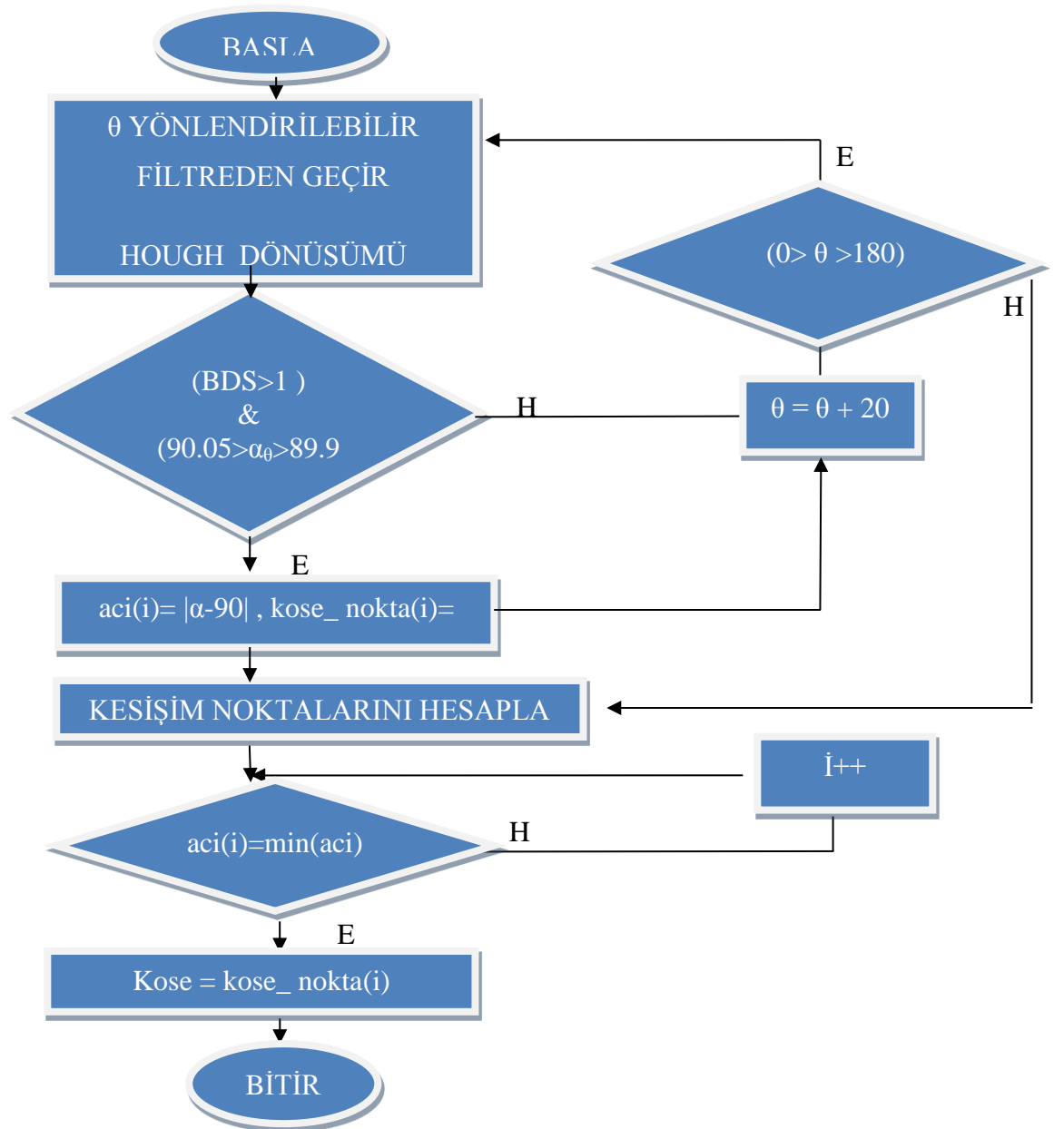


Şekil 3.1: Makinenin Referans için Çekilmiş Boş Resmi.



Şekil 3.2: Kullanılan Deney Sacı ile Çekilmiş Resim.

Bu tezde yeni bir yöntem denenmiştir. Tezde kullanılan makineler büyük uzunluk ve genişliklere sahip olduğu için kullanılan saclarda o şekilde büyük ve düzdür. Burada kullanılan sacların eğrilik, kesiklik yamukluk ihtimalleri çok düşük kabul edilirse projede Hough [20] Teoremi çok iyi işleyebilir. Yönlendirilebilir filtreler[22] yöntemi ile resimler filtrelenmiş ve bu resimler üzerine Hough [20] Teoremi uygulanmış ve başarı kriteri ölçülmüştür. Programın akış diyagramı aşağıda verilmiştir. Diyagramda denklem sayısı BDS, bulunan denklemlerin kesişim açısı  $\alpha$  olarak ifade edilmiştir. Kesişim açısı doğruların eksenler olan açısı ve ikisinin farkından hesaplanmaktadır.



Şekil 3.3: Program Akış Diyagramı.



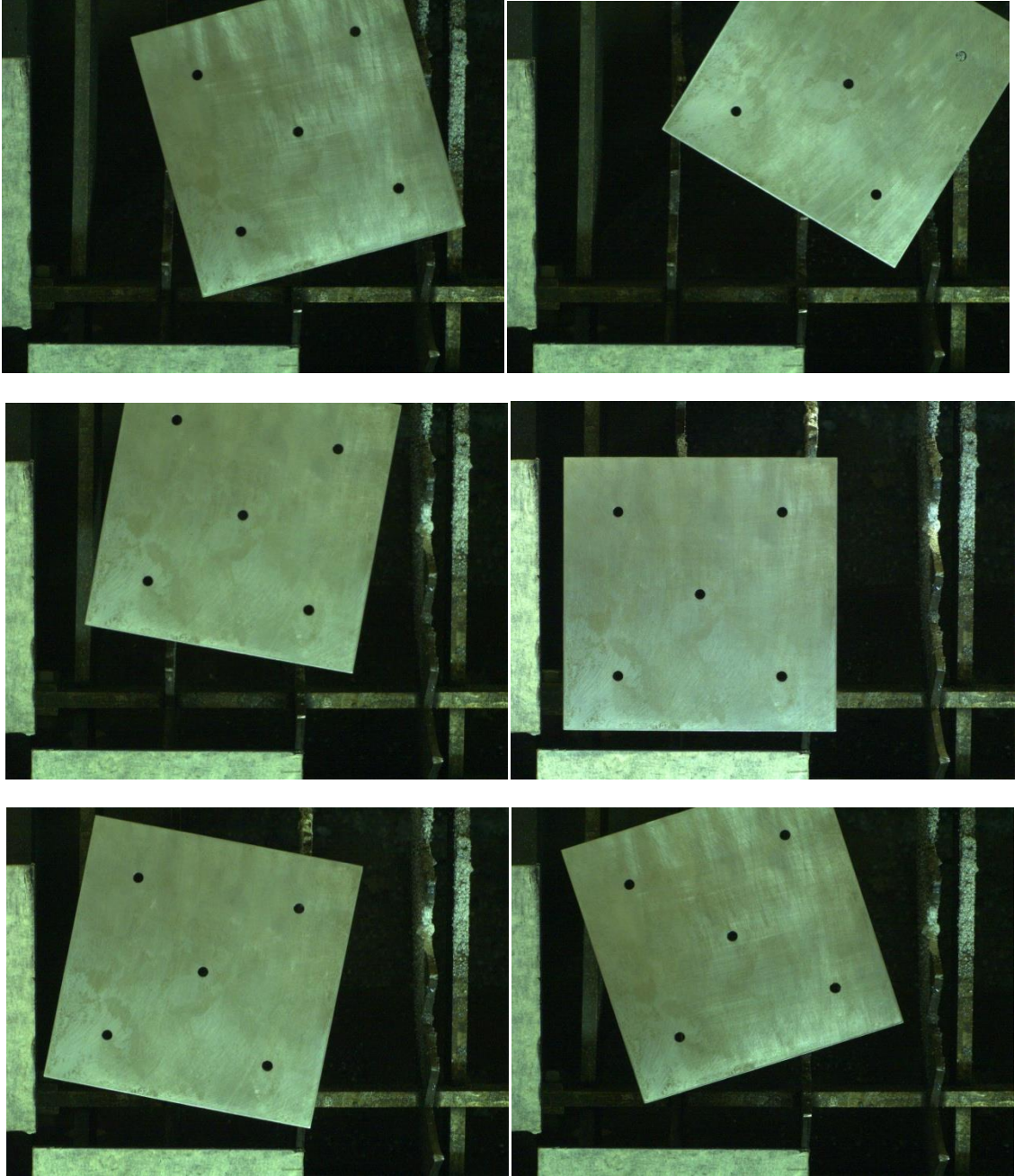
Programda ilk önce makine boşken referans resmi alınmış ardından sac koyulmuş şekilde resim alınmıştır. Alınan resimlerden sadece sac resmi elde edilmiş dolu resim boş resimden çıkarılmıştır. Sac resmine morfolojik işlemler uygulanmış ve medyan filtre ile temizlenmiştir. Ve en sonunda kullanılacak resim elde edilmiştir. Kullanılacak resim üzerinde akış diyagramı verilen program uygulanmıştır.

Projede 200 ile 360 derece arasındaki açılar 20 derece şeklinde arttırılarak bir resim için 9 deneme yapılmıştır. Bu 9 deneme sonunda açı değeri yaklaşılmaya başladığı zaman doğrular ve kesim noktaları oluşmaya başladığı görülmektedir. Bu noktalar köşe noktaları kabul edilmiştir.

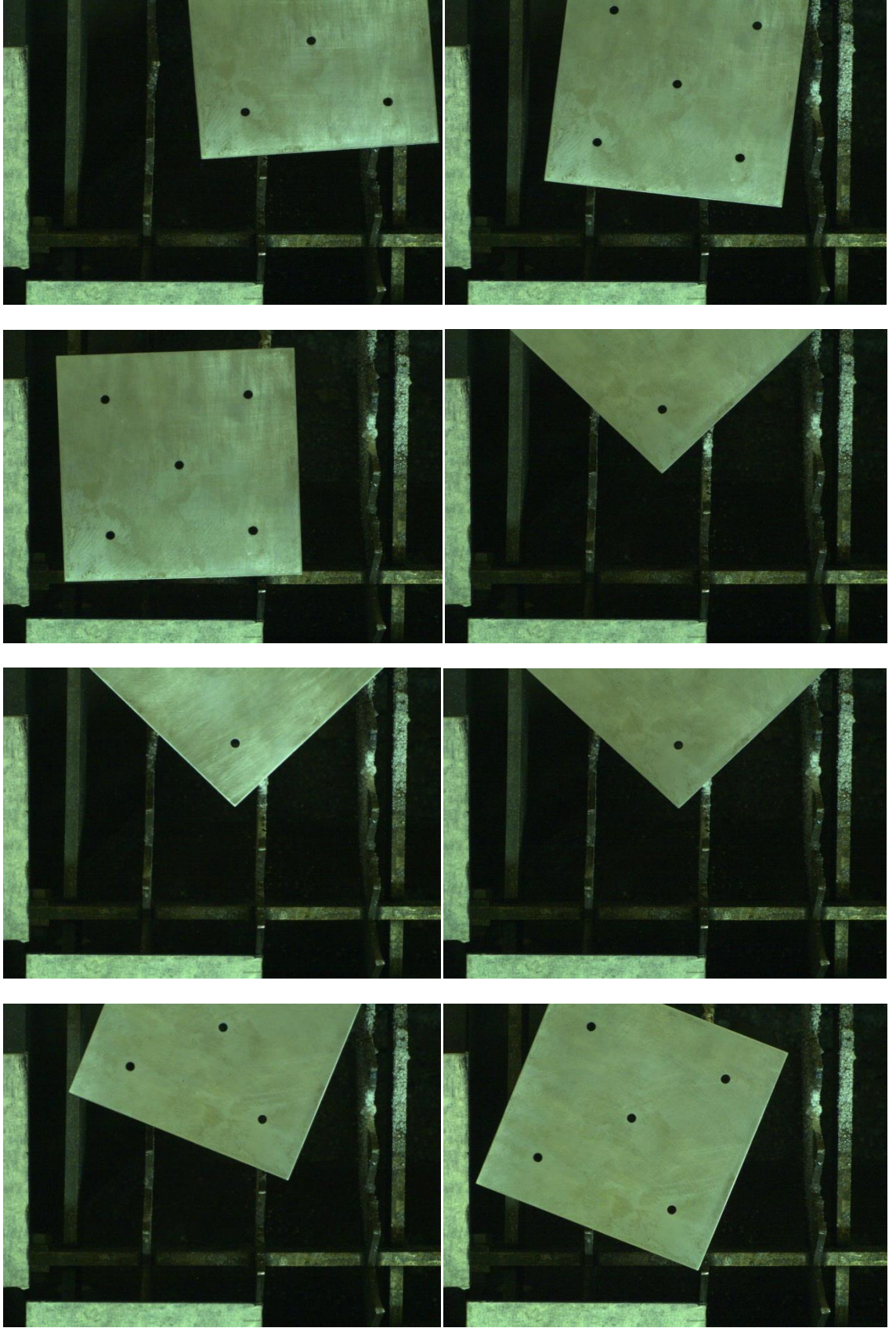
Bulunan köşe noktası pikseline karşılık gelen uzaklık hesaplanarak referans olarak alınan köşe noktasına eklenir ve gerçek dünyada hangi noktaya denk geldiği hesaplanır. Daha sonra hesaplanan noktaya kafa gönderilir.

#### 4. BULGULAR

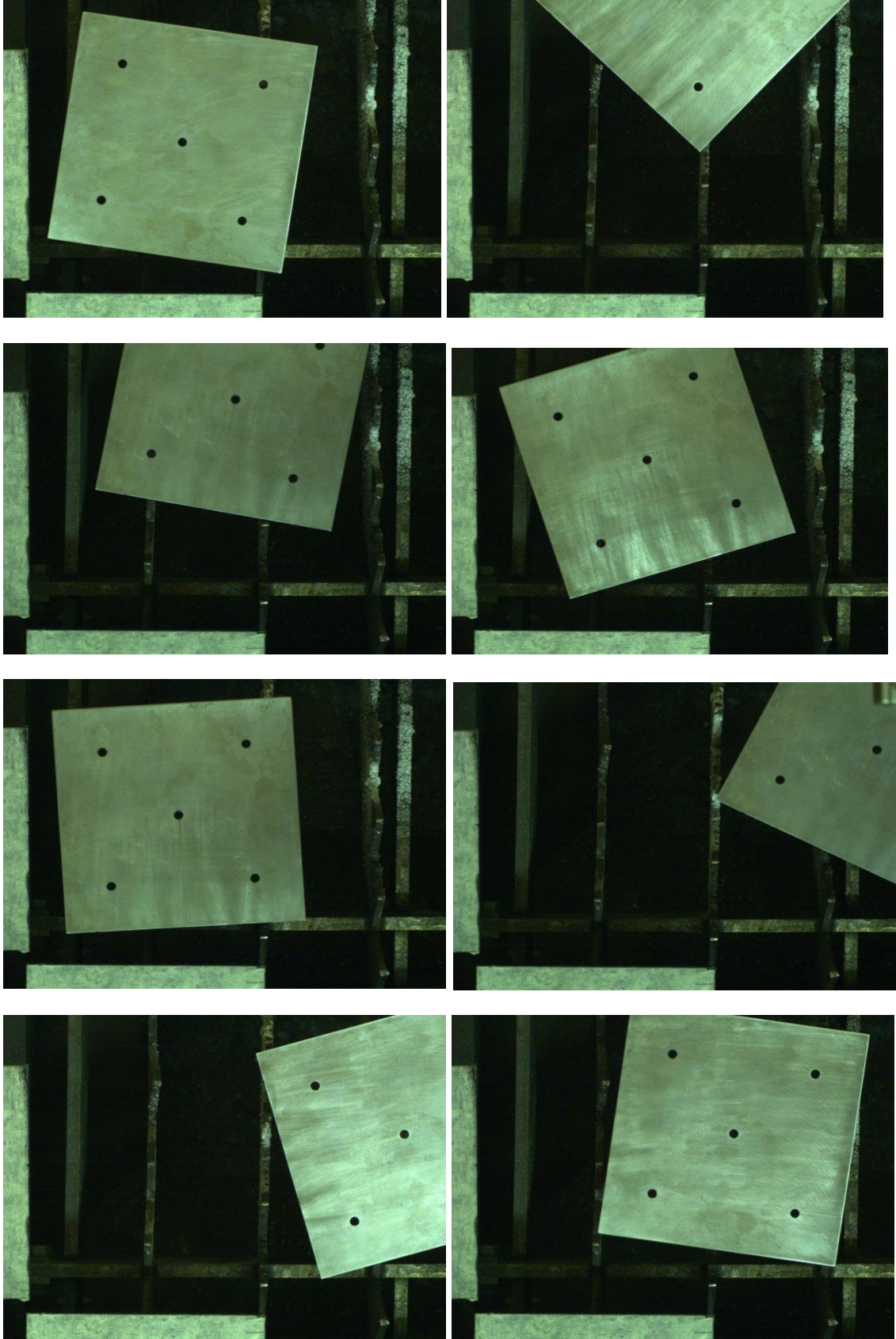
Bu projede Harris [2], Hough, Yönlendirilmiş Filtre - Harris ve Yönlendirilmiş Filtre – Hough dönüşümleri ölçülmüş ve kıyaslanmıştır. Harris[2] algoritmasını, Moravec [1] algoritmasının eksiklerinden geliştirildiği için Moravec [1] algoritmasını kıyaslamaya girmeye gerek görülmemiştir. Bu çalışmada test ölçümü için kullanılan resimler aşağıda verilmiştir.



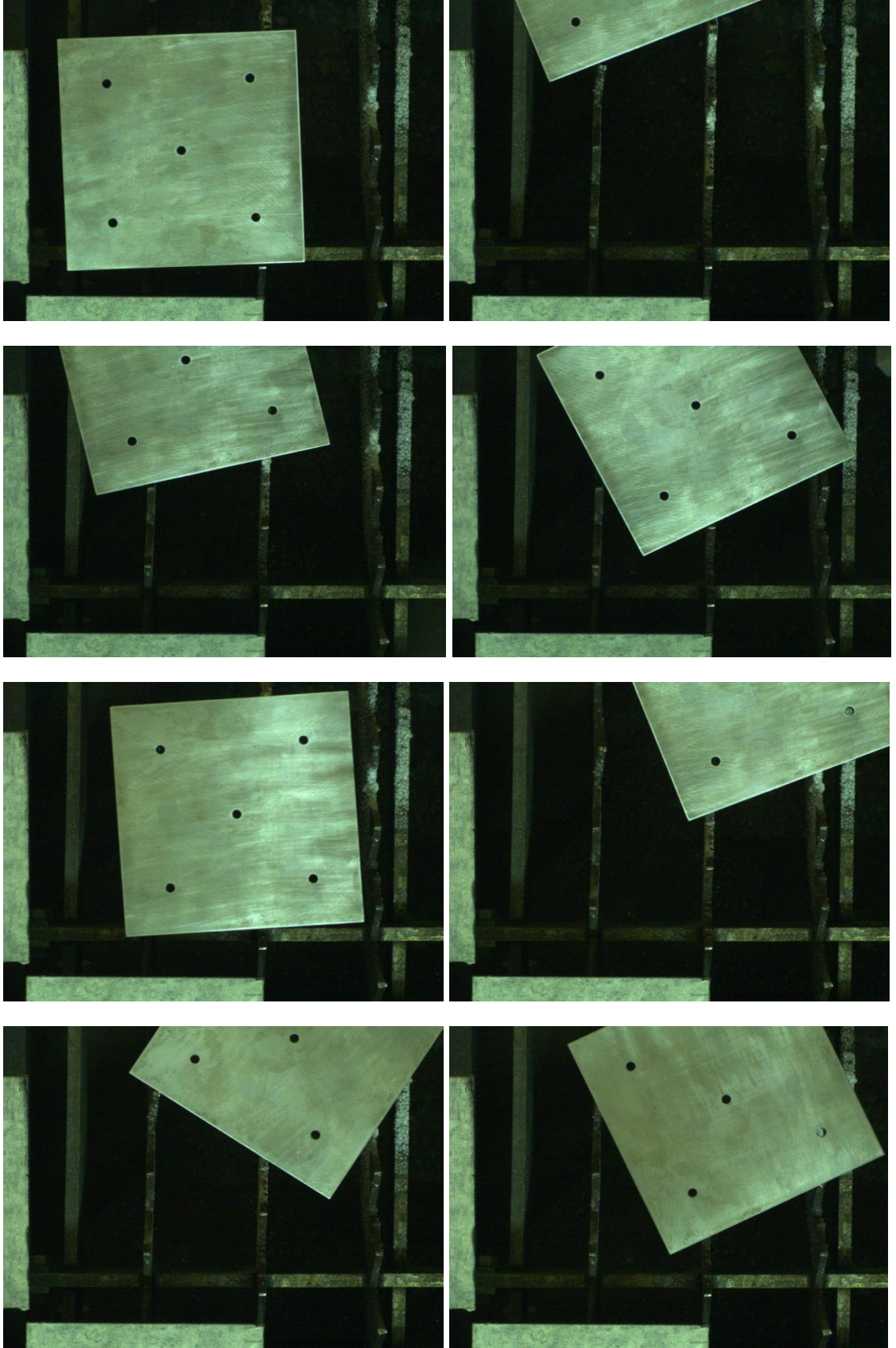
Şekil 4.1: Uygulanacak İlk 6 İmge.



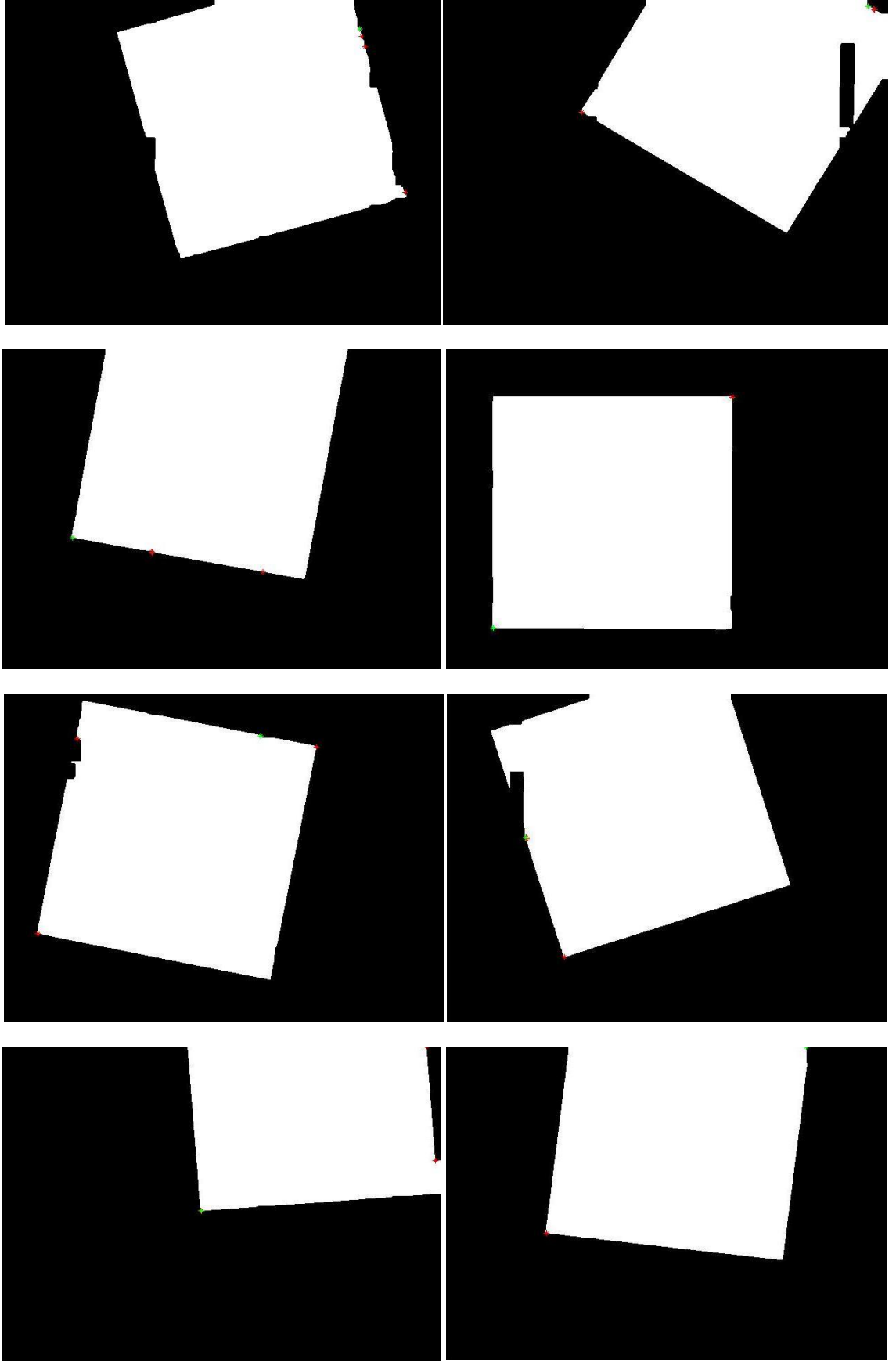
Şekil 4.2: Uygulanacak Sonraki 8 İmge İmge7- İmge14).



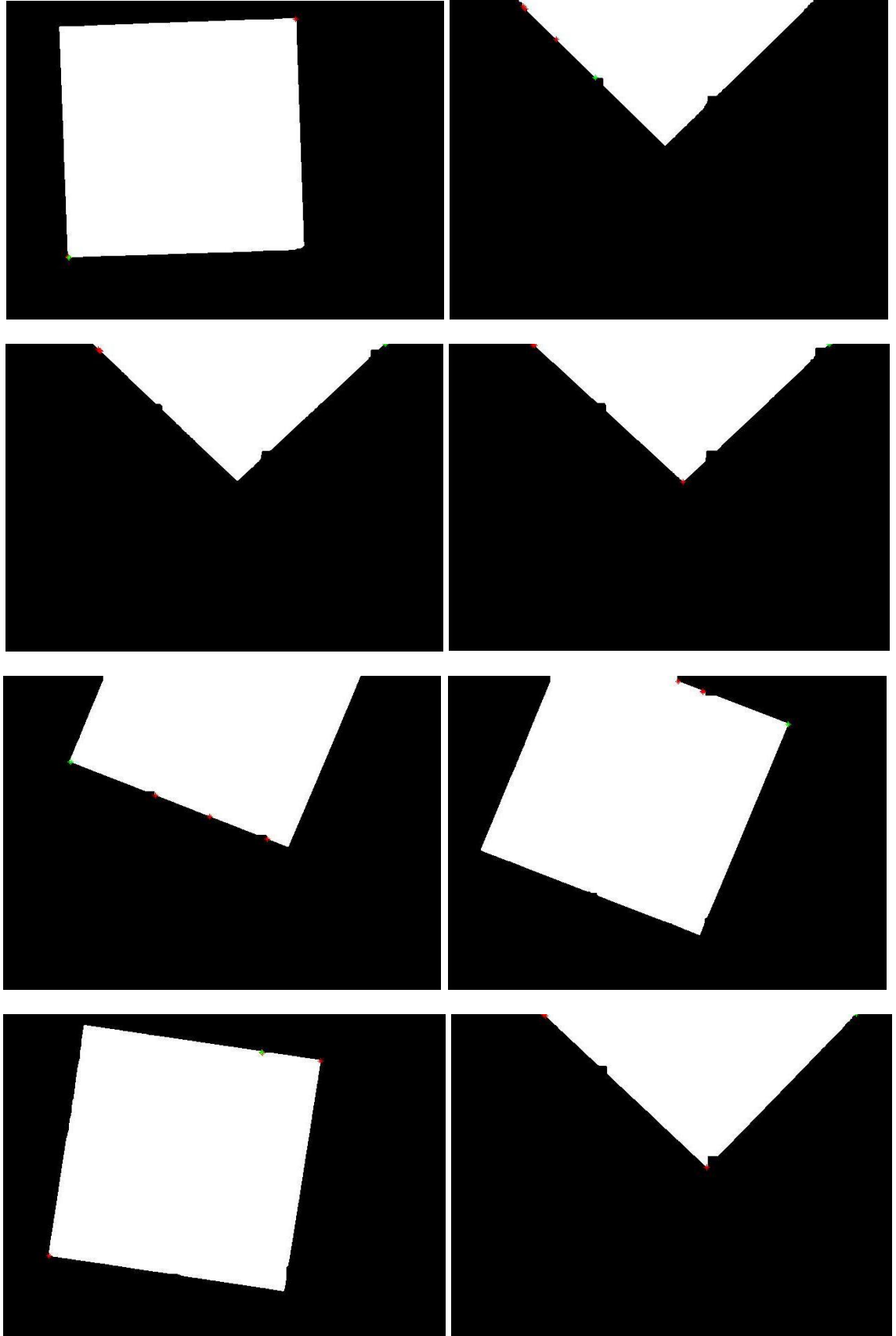
Şekil 4.3: Uygulanacak Sonraki 8 İmge(İmge15- İmge22).



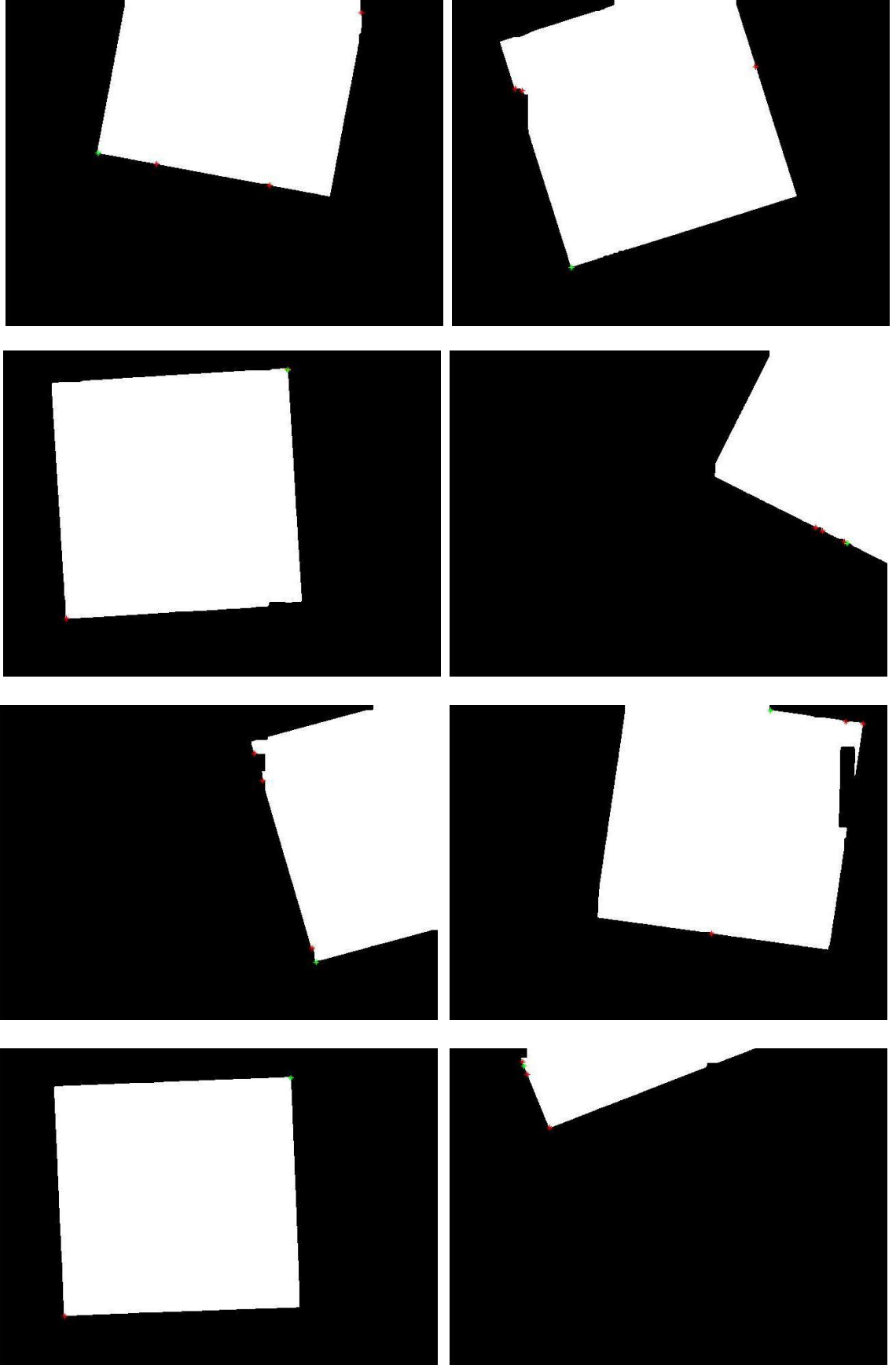
Şekil 4.4: Uygulanacak Sonraki 8 İmge(İmge23- İmge30).



Şekil 4.5: İlk 8 İmge için Harris Yöntemi Uygulaması(İmge1- İmge8).

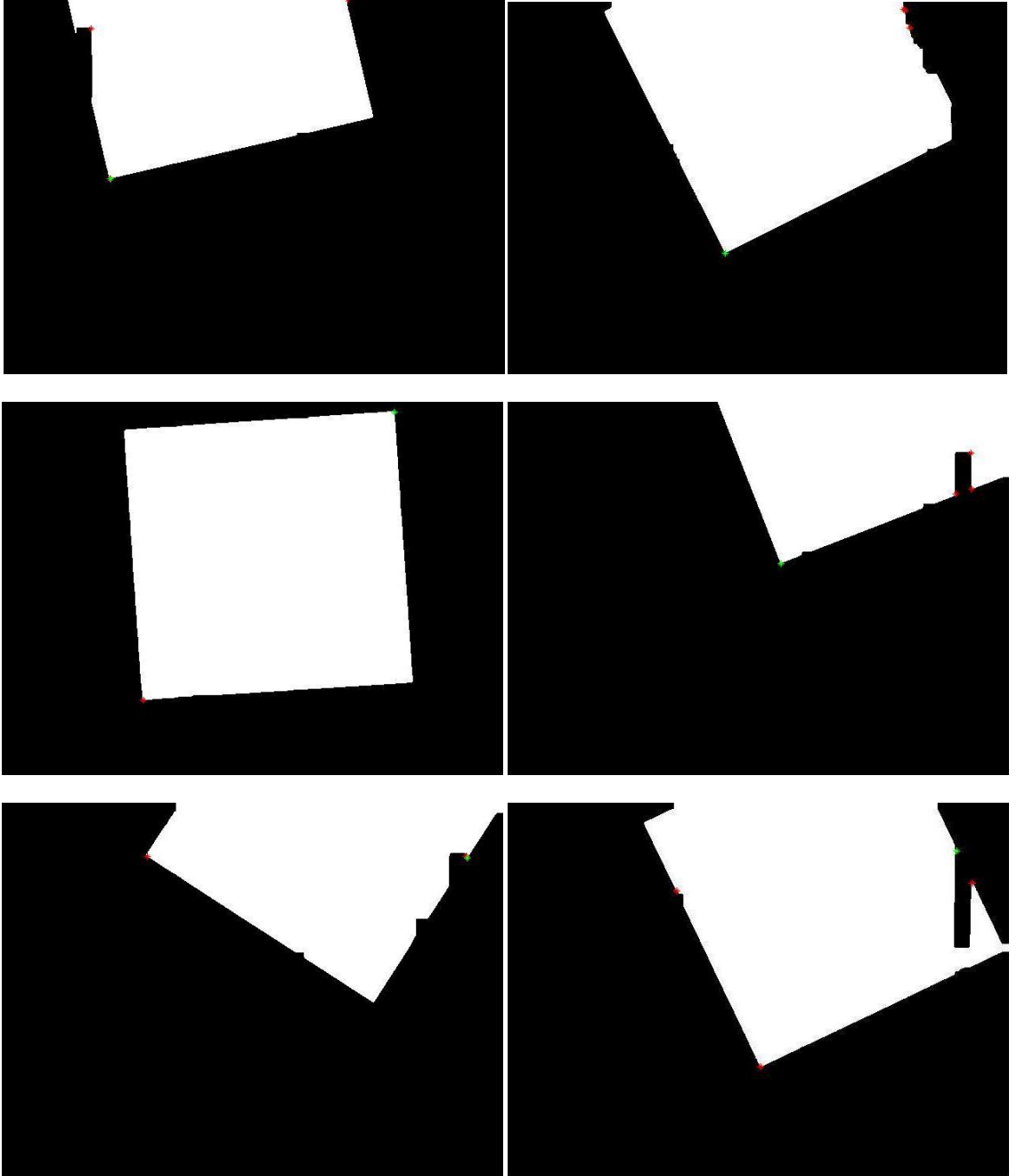


Şekil 4.6: Sonraki 8 İmge için Harris Yöntemi Uygulaması(İmge9- İmge16).



Şekil 4.7: Sonraki 8 İmge için Harris Yöntemi Uygulaması (İmge17- İmge24).





**Şekil 4.8:** Sonraki 6 İmge için Harris Yöntemi Uygulaması (İmge25- İmge30).

Yukarıdaki gösterilen şekillerde Harris[2] Dönüşümü uygulanmış, bulunan koordinatlar, gerçek koordinatlar ve hatalı mesafe aşağıda verilmiştir. Dönüşümde en yüksek köşe olma değerlikli nokta yeşil olarak gösterilmiş, diğer 3 nokta ise kırmızı ile işaretlenmiştir.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge10	X	76,501	46,220	47,23	60,063	-----	-----	-----	106,402				
	Y	95,215	125,785	124,703	111,653	-----	-----	-----	65,864	-----	-----	-----	65,146
imge11	X	175,132	54,439	55,016	55,305	-----	-----	-----	113,404				
	Y	128,813	126,434	125,857	125,568	-----	-----	-----	67,364	-----	-----	-----	82,239
imge12	X	174,338	113,199	50,690	51,123	-----	113,600	-----	-----				
	Y	128,813	66,953	128,452	128,02	-----	67,208	-----	-----	-----	0,475	-----	-----
imge16	X	184,721	122,067	54,007	54,656	-----	122,551	-----	-----				
	Y	128,741	63,492	128,452	127,876	-----	63,249	-----	-----	-----	0,541	-----	-----
imge20	X	182,846	169,578	172,248	181,620	-----	125,601	-----	-----				
	Y	47,414	54,119	52,6051	48,063	-----	76,751	-----	-----	-----	49,459	-----	-----
imge24	X	46,725	57,612	48,167	46,076	-----	57,885	-----	-----				
	Y	120,954	93,989	117,133	122,540	-----	93,840	-----	-----	-----	0,312	-----	-----
imge28	X	115,867	185,802	185,514	180,034	116,625	-----	-----	-----				
	Y	69,116	96,297	109,779	94,783	68,744	-----	-----	-----	0,844	-----	-----	-----

**Tablo 4.1:** Tek Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge2	X	191,426	193,733	194,093	72,969	-----	158,377	-----	73,981				
	Y	125,857	124,56	124,271	81,012	-----	29,158	-----	80,755	-----	101,742	-----	1,045
imge7	X	98,996	193,444	197,266	98,924	99,664	-----	197,492	-----				
	Y	56,859	128,813	78,777	56,931	57,436	-----	64,707	-----	0,883	-----	14,072	-----
imge8	X	165,831	57,395	165,831	165,759	-----	57,442	-----	155,738				
	Y	128,741	46,765	128,669	128,813	-----	46,452	-----	35,552	-----	0,317	-----	93,798
imge13	X	43,625	79,457	126,61	102,312	43,704	-----	135,499	-----				
	Y	90,961	76,325	57,147	66,953	91,068	-----	53,871	-----	0,133	-----	9,474	-----
imge17	X	54,295	165,254	126,393	78,881	54,469	152,164	-----	-----				
	Y	63,997	123,117	50,514	59,382	64,409	45,632	-----	-----	0,448	78,583	-----	-----
imge21	X	148,311	122,572	125,961	146,582	-----	121,233	-----	148,631				
	Y	16,556	107,255	95,503	22,684	-----	112,433	-----	16,639	-----	5,348	-----	6,383
imge25	X	54,584	141,678	47,59	54,511	-----	152,005	-----	55,165				
	Y	62,915	128,813	118,070	62,987	-----	85,505	-----	62,704	-----	44,522	-----	0,712
imge29	X	186,451	185,946	68,859	186,163	-----	-----	69,605	152,585				
	Y	108,265	108,914	108,697	108,697	-----	-----	109,184	55,164	-----	-----	0,891	63,192

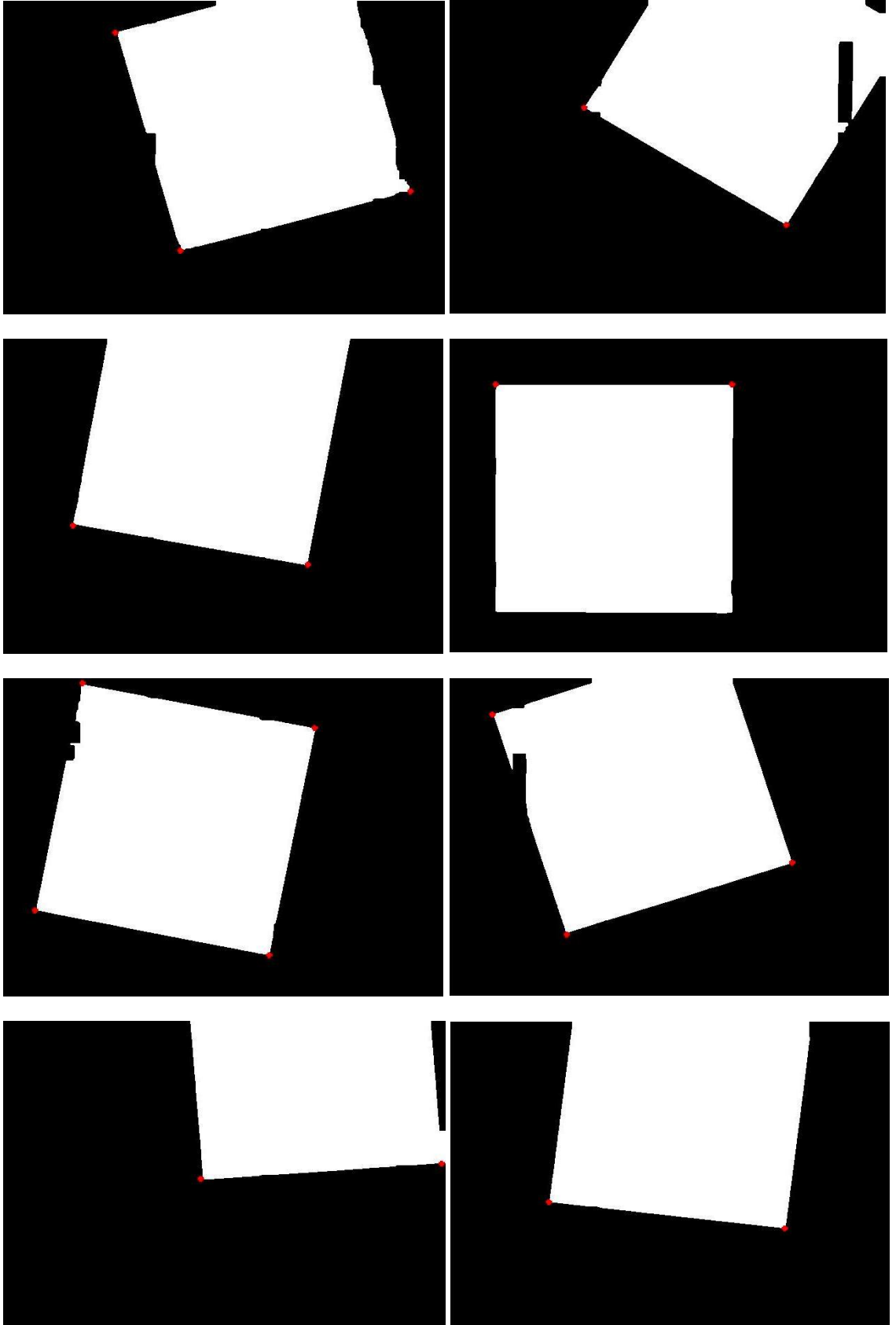
**Tablo 4.2:** İki Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge1	X	165,326	184,721	166,408	167,778	90,102	185,542	63,702	-----				
	Y	116,196	47,486	113,168	108,77	19,435	45,404	114,704	-----	122,562	2,238	102,717	-----
imge3	X	45,067	78,448	78,664	125,168	44,819	-----	162,012	143,614				
	Y	47,702	41,574	41,43	32,922	47,197	-----	128,339	29,999	0,563	-----	120,416	18,677
imge6	X	48,383	64,461	48,528	48,744	-----	64,545	158,742	34,746				
	Y	68,388	18,431	68,323	67,89	-----	18,758	48,557	111,954	-----	0,338	111,973	46,234
imge14	X	158,333	112,046	122,356	122,572	158,095	-----	29,201	121,297				
	Y	107,400	126,145	121,747	121,531	107,528	-----	52,271	15,434	0,270	-----	116,210	106,104
imge18	X	65,687	143,264	41,894	44,778	66,125	160,921	-----	36,127				
	Y	16,051	100,334	91,322	90,385	15,795	45,794	-----	110,989	0,507	57,327	-----	22,347
imge22	X	150,33	189,335	182,125	125,528	77,280	91,079	-----	174,975				
	Y	126,217	120,233	121,243	29,029	36,037	134,732	-----	22,140	116,055	99,320	-----	49,925
imge26	X	95,752	161,433	161,865	163,668	95,845	184,665	-----	51,745				
	Y	35,951	125,785	125,352	118,935	34,864	80,064	-----	124,624	1,091	51,285	-----	112,067
imge30	X	180,323	108,297	186,307	77,439	-----	108,785	198,805	65,825				
	Y	110,932	31,408	99,108	96,224	-----	31,084	74,184	120,784	-----	0,586	27,882	27,167

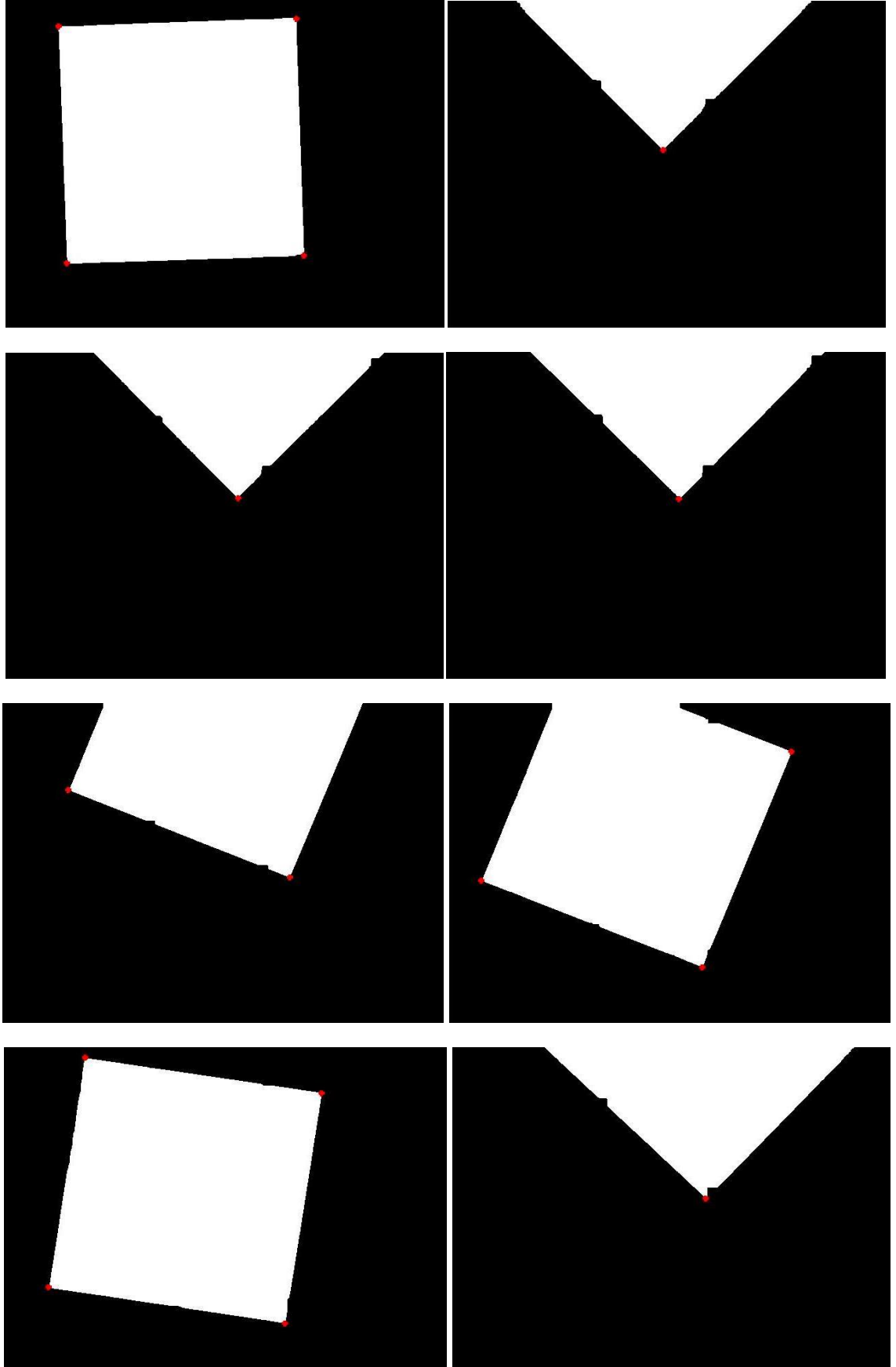
**Tablo 4.3:** Üç Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge4	X	35,045	134,54	34,973	34,9728	34,412	135,608	34,412	134,807				
	Y	8,986	107,976	8,986	34,9728	8,422	108,017	108,017	8,422	0,848	1,068	99,033	103,304
imge5	X	122,716	46,004	146,004	29,6376	147,203	48,809	127,205	29,211				
	Y	111,077	109,923	106,535	28,3081	107,553	125,152	8,758	28,357	24,739	15,485	99,567	0,429
imge9	X	41,606	137,208	41,534	41,4617	38,307	138,003	140,503	41,807				
	Y	180,701	120,738	18,142	18,1422	116,931	120,431	21,535	17,635	63,855	0,852	99,027	0,614
imge15	X	123,077	123,293	147,518	34,6124	132,470	49,473	147,869	34,475				
	Y	112,158	111,942	108,625	25,352	10,330	123,723	108,724	25,428	102,260	74,754	0,365	0,157
imge19	X	135,117	42,039	135,117	41,664	135,446	42,250	141,445	35,950				
	Y	120,449	15,547	120,522	15,547	120,833	15,738	21,536	115,033	0,505	0,285	99,188	99,650
imge23	X	137,713	42,543	137,713	137,641	138,076	38,381	141,876	42,381				
	Y	115,691	12,518	115,763	115,763	115,936	112,336	16,540	12,641	0,438	99,905	99,310	140,387
imge27	X	159,630	67,345	159,558	159,558	166,345	60,725	160,465	67,445				
	Y	124,631	18,935	124,703	124,775	25,528	117,568	124,741	18,568	99,330	98,855	0,907	140,588

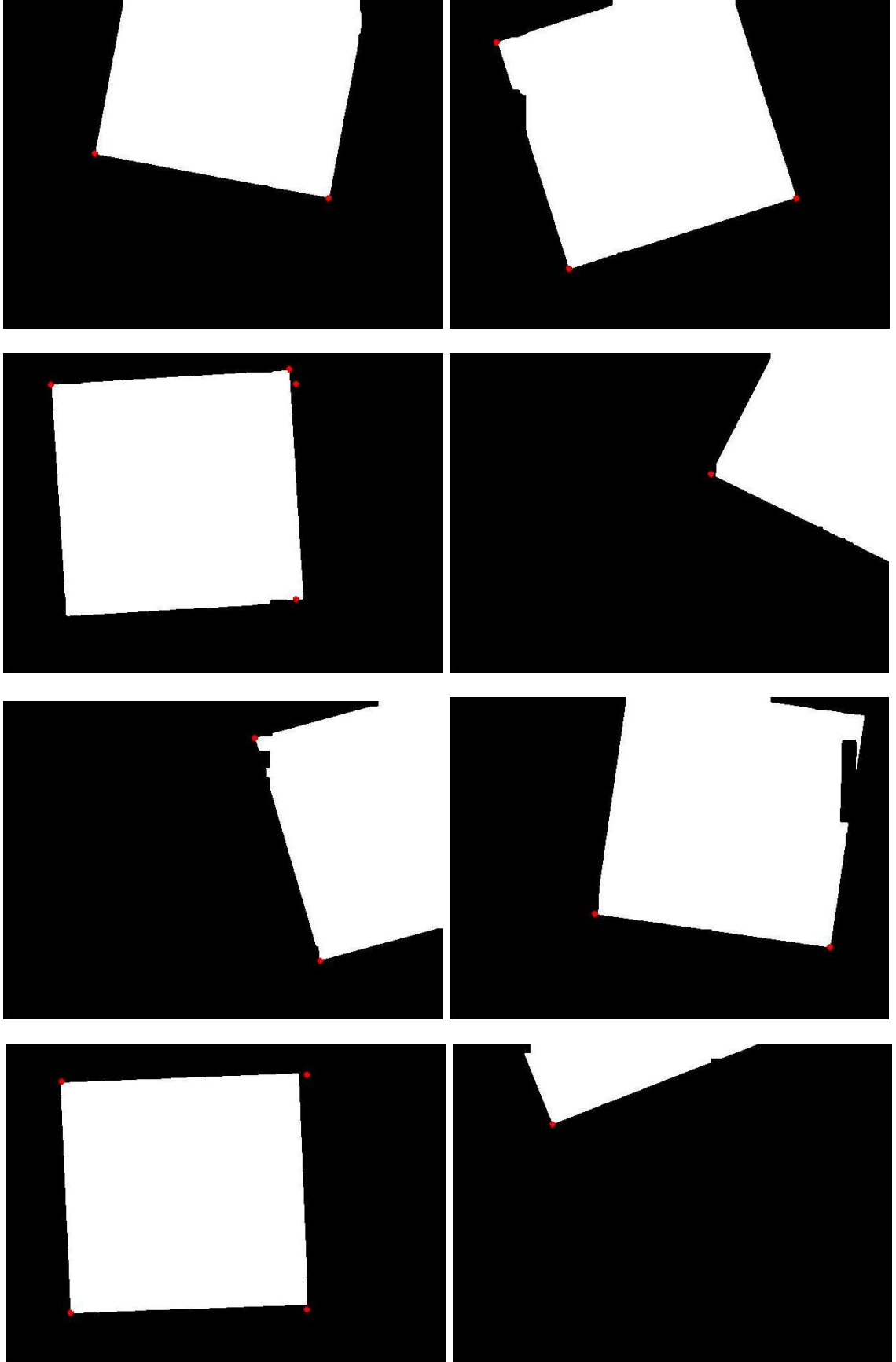
**Tablo 4.4:** Dört Köşeli İmge Koordinatları ve Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.



Şekil 4.9: İlk 8 İmge için Hough Yöntemi Uygulaması(İmge1 - İmge8).

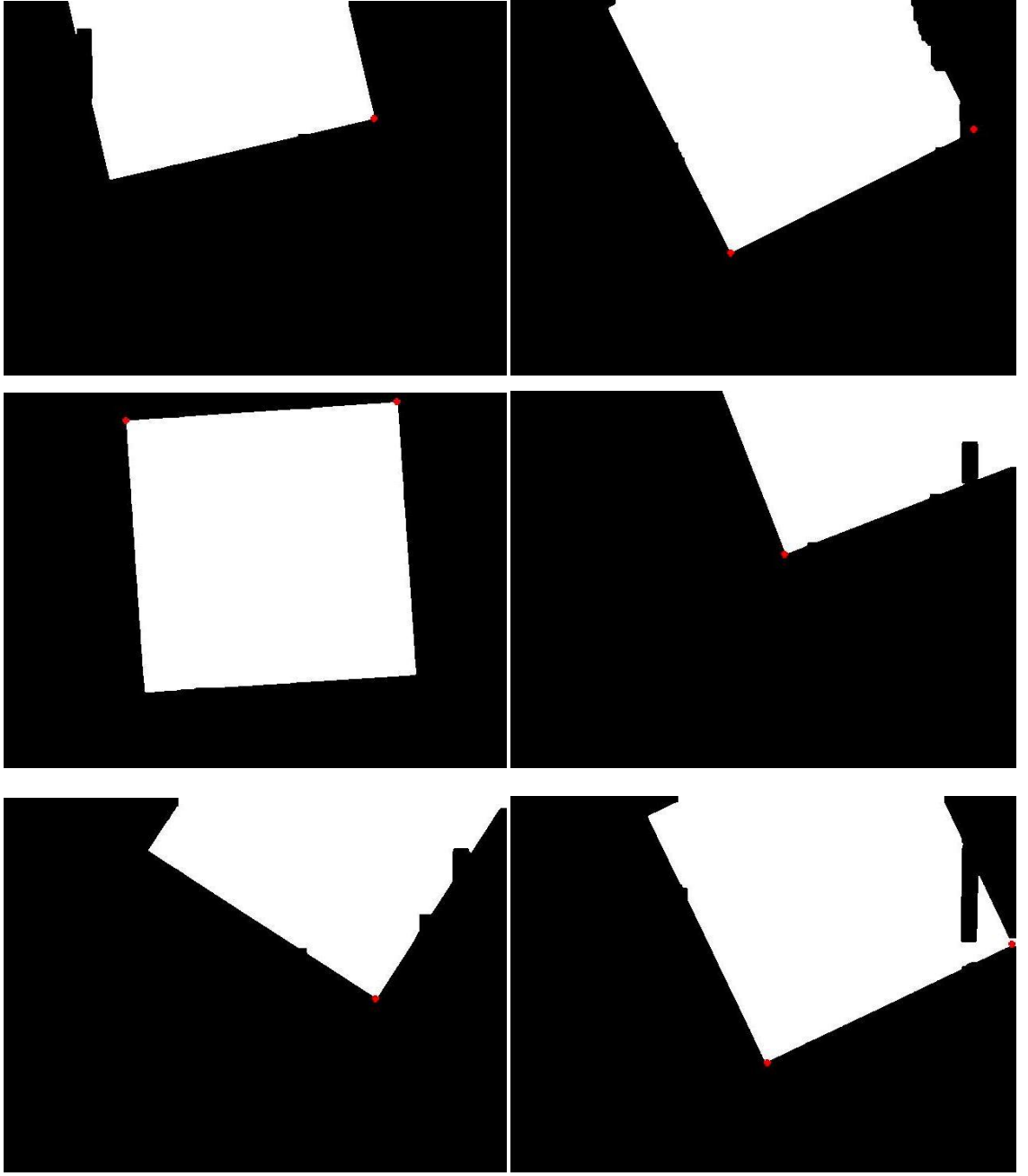


Şekil 4.10: Sonraki 8 İmge için Hough Yöntemi Uygulaması(İmge9- İmge16).



Şekil 4.11: Sonraki 8 İmge için Hough Yöntemi Uygulaması(İmge17- İmge24).





**Şekil 4.12:** Sonraki 6 İmge için Hough Yöntemi Uygulaması(İmge25- İmge30).

Yukarıda Hough [20] dönüşümü uygulanmış resimler verilmiştir ve tablolar aşağıda gösterilmiştir.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge10	X	105,917	-----	-----	-----	106,402	-----	-----	-----	0,485	-----	-----	-----
	Y	65,871	-----	-----	-----	65,864	-----	-----	-----				
imge11	X	113,091	-----	-----	-----	113,404	-----	-----	-----	0,313	-----	-----	-----
	Y	67,349	-----	-----	-----	67,364	-----	-----	-----				
imge12	X	112,872	-----	-----	-----	113,600	-----	-----	-----	0,847	-----	-----	-----
	Y	66,775	-----	-----	-----	67,208	-----	-----	-----				
imge16	X	121,993	-----	-----	-----	122,551	-----	-----	-----	0,580	-----	-----	-----
	Y	63,408	-----	-----	-----	63,249	-----	-----	-----				
imge20	X	125,01	-----	-----	-----	125,601	-----	-----	-----	0,667	-----	-----	-----
	Y	76,441	-----	-----	-----	76,751	-----	-----	-----				
imge24	X	57,307	-----	-----	-----	57,885	-----	-----	-----	0,578	-----	-----	-----
	Y	93,817	-----	-----	-----	93,840	-----	-----	-----				
imge28	X	115,314	-----	-----	-----	116,625	-----	-----	-----	1,370	-----	-----	-----
	Y	69,141	-----	-----	-----	68,744	-----	-----	-----				

**Tablo 4. 5:** Tek Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge2	X	72,197	157,535	-----	-----	73,981	158,377	-----	-----				
	Y	81,359	30,056	-----	-----	80,755	29,158	-----	-----	1,883	1,231	-----	-----
imge7	X	197,935	97,696	-----	-----	197,492	99,664	-----	-----				
	Y	64,09	57,291	-----	-----	64,707	57,436	-----	-----	0,760	1,973	-----	-----
imge8	X	56,74	155,759	-----	-----	57,442	155,738	-----	-----				
	Y	46,916	34,953	-----	-----	46,452	35,552	-----	-----	0,841	0,599	-----	-----
imge13	X	135,412	42,947	-----	-----	135,499	43,704	-----	-----				
	Y	53,761	91,316	-----	-----	53,871	91,068	-----	-----	0,140	0,797	-----	-----
imge17	X	53,741	151,49	-----	-----	54,469	152,164	-----	-----				
	Y	64,426	45,676	-----	-----	64,409	45,632	-----	-----	0,728	0,675	-----	-----
imge21	X	148,049	120,7	-----	-----	148,631	121,233	-----	-----				
	Y	16,62	112,437	-----	-----	16,639	112,433	-----	-----	0,582	0,533	-----	-----
imge25	X	151,12	0	-----	-----	152,005	55,165	-----	-----				
	Y	85,392	0	-----	-----	85,505	62,704	-----	-----	0,892	83,516	-----	-----
imge29	X	151,537	0	-----	-----	152,585	69,605	-----	-----				
	Y	54,914	0	-----	-----	55,164	109,184	-----	-----	1,077	129,484	-----	-----

**Tablo 4.6:** İki Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

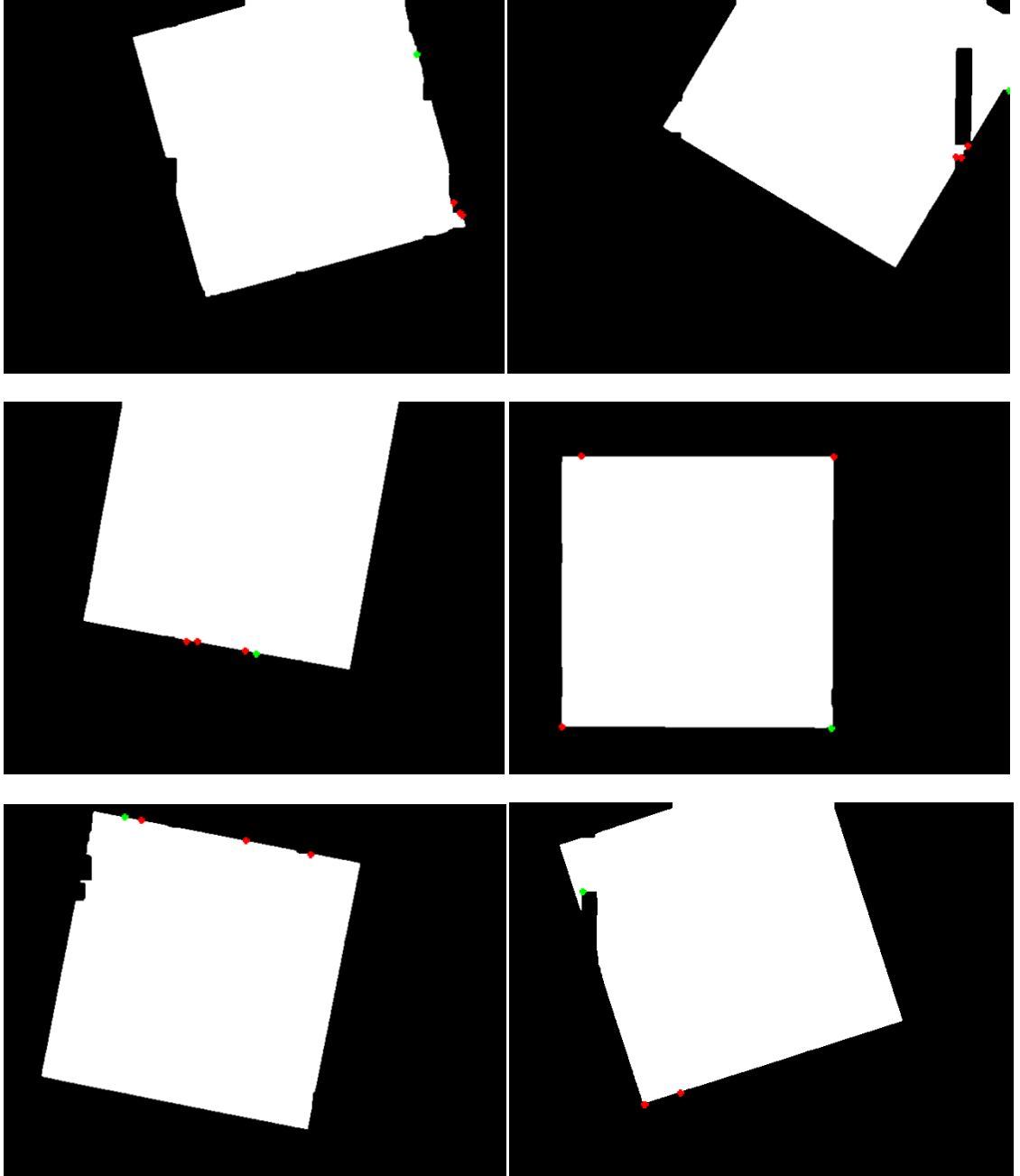
		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge1	X	62,197	185,313	89,335	-----	63,702	185,542	90,102	-----				
	Y	114,493	45,113	18,983	-----	114,704	45,404	19,435	-----	1,520	0,370	0,890	-----
imge3	X	142,848	44,616	0	-----	143,614	44,819	162,012	-----				
	Y	30,079	47,163	0	-----	29,999	47,197	128,339	-----	0,770	0,206	206,685	-----
imge6	X	33,397	158,965	64,519	-----	34,746	158,742	64,545	-----				
	Y	112,807	48,751	17,681	-----	111,954	48,557	18,758	-----	1,596	0,296	1,077	-----
imge14	X	28,81	121,023	158,348	-----	29,201	121,297	158,095	-----				
	Y	52,419	15,159	107,725	-----	52,271	15,434	107,528	-----	0,418	0,388	0,321	-----
imge18	X	34,984	160,633	65,384	-----	36,127	160,921	66,125	-----				
	Y	110,726	45,557	16,342	-----	110,989	45,794	15,795	-----	1,173	0,373	0,921	-----
imge22	X	76,393	175,069	0	-----	77,280	174,975	91,079	-----				
	Y	36,066	22,116	0	-----	36,037	22,140	134,732	-----	0,887	0,097	162,629	-----
imge26	X	95,117	184,044	0	-----	95,845	184,665	51,745	-----				
	Y	35,705	80,405	0	-----	34,864	80,064	124,624	-----	1,112	0,708	134,940	-----
imge30	X	109,031	198,067	0	-----	108,785	198,805	65,825	-----				
	Y	31,15	74,466	0	-----	31,084	74,184	120,784	-----	0,255	0,790	137,556	-----

**Tablo 4.7:** Üç Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

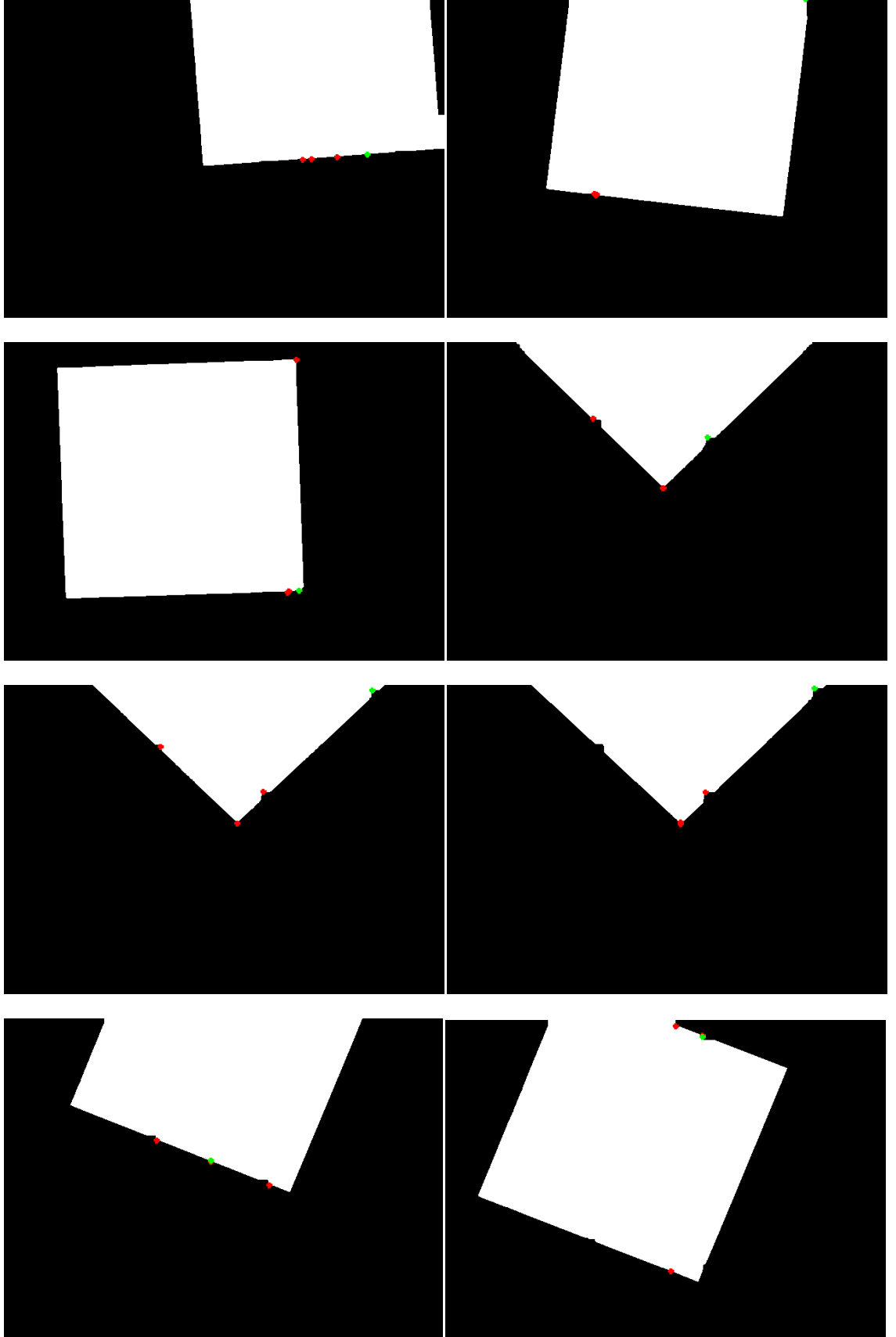
		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge4	X	34,685	134,324	0	0	34,412	135,608	34,412	134,807				
	Y	108,409	108,409	0	0	108,017	108,017	8,422	8,422	0,478	1,343	35,428	135,070
imge5	X	48,497	145,883	28,842	126,725	48,809	147,203	29,211	127,205				
	Y	126,353	106,964	28,438	9,024	125,152	107,553	28,357	8,758	1,241	1,445	0,378	0,549
imge9	X	37,599	140,59	41,089	137,535	38,307	140,503	41,807	138,003				
	Y	117,534	21,382	18,180	120,741	116,931	21,535	17,635	120,431	0,930	0,176	0,901	0,561
imge15	X	132,284	33,817	147,456	49,246	132,470	34,475	147,869	49,473				
	Y	10,307	25,75	108,911	123,973	10,330	25,428	108,724	123,723	0,187	0,733	0,453	0,338
imge19	X	137,929	35,481	135,223	138,073	141,445	35,950	135,446	42,250				
	Y	22,757	114,86	121,51	115,258	21,536	115,033	120,833	15,738	3,722	0,500	0,713	138,153
imge23	X	141,245	38,545	42,17	141,245	141,876	38,381	42,381	138,076				
	Y	14,249	112,423	12,717	115,547	16,540	112,336	12,641	115,936	2,376	0,186	0,224	3,193
imge27	X	60,041	159,493	0	0	60,725	160,465	166,345	67,445				
	Y	118,28	125,128	0	0	117,568	124,741	25,528	18,568	0,987	1,046	168,292	69,954

**Tablo 4.8:** Dört Köşeli İmge Koordinatları ve Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

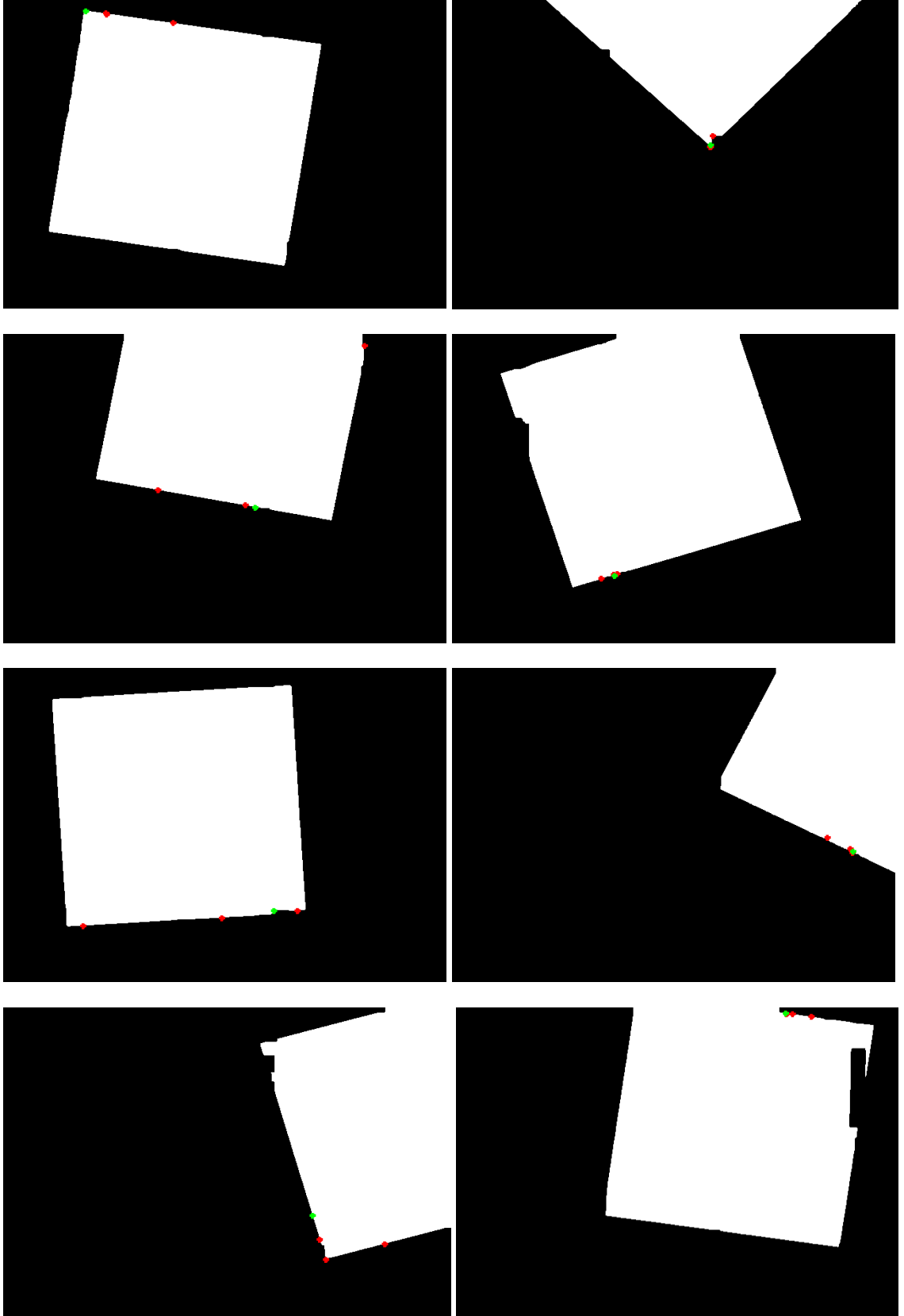
Aşağıda Yönlendirilmiş filtreli resimlere Haris [2] yöntemi uygulanmış ve bulunan köşeler toplanmıştır. Uygulanan yöntemde en büyük 4 köşe işaretlenmiştir. Yönlendirilmiş filtre 20 derecelik aralıklarla 9 defa döndürülmüş ve böylece bulunan köşeler köşe olma derecesine göre sıralanmıştır. En büyük köşe olma derecesine sahip köşe yeşil olarak gösterilmiştir.



Şekil 4.13: İlk 6 İmge için Yönlendirilmiş Filtre-Harris Uygulaması(İmge1-İmge6).

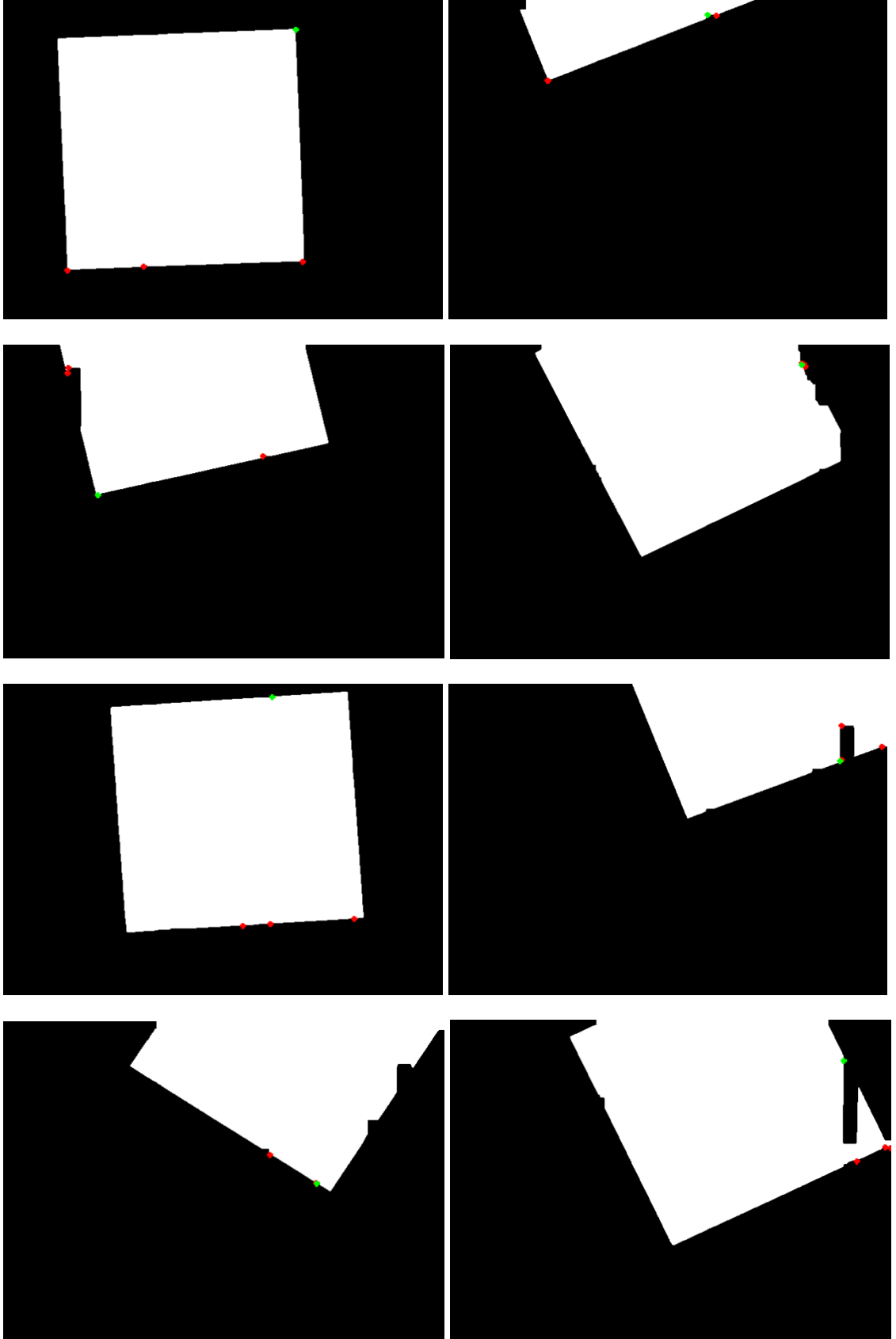


Şekil 4.14: Sonraki 8 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge7- İmge14).



Şekil 4.15: Sonraki 8 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge15-İmge22).





Şekil 4.16: Sonraki 8 İmge için Yönlendirilmiş Filtre -Harris Uygulaması(İmge23-İmge30).

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge10	X	105,917	124,374	76,573	105,701	-----	-----	-----	106,402				
	Y	65,5107	87,356	95,648	65,511	-----	-----	-----	65,864	-----	-----	-----	0,785
imge11	X	169,508	123,8697	80,971	113,055	-----	-----	-----	113,404				
	Y	126,073	81,012	101,055	67,097	-----	-----	-----	67,364	-----	-----	-----	0,439
imge12	X	169,075	123,509	113,199	113,199	-----	-----	-----	113,6				
	Y	126,938	80,795	66,592	67,457	-----	-----	-----	67,208	-----	-----	-----	0,472
imge16	X	122,067	123,149	121,923	122,067	-----	-----	-----	122,551				
	Y	63,997	68,2504	63,1314	63,131	-----	-----	-----	63,249	-----	-----	-----	0,498
imge20	X	171,455	182,125	181,909	180,899	125,601	-----	-----	-----				
	Y	54,191	48,135	47,702	49,361	76,751	-----	-----	-----	51,103	-----	-----	-----
imge24	X	124,2302	127,907	57,107	57,2511	-----	-----	-----	57,885				
	Y	122,108	121,747	93,8452	93,701	-----	-----	-----	93,84	-----	-----	-----	0,649
imge28	X	179,746	180,467	180,467	197,482	-----	-----	-----	116,625				
	Y	94,422	94,927	110,067	100,622	-----	-----	-----	68,744	-----	-----	-----	86,914

**Tablo 4.9:** Tek Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge2	X	181,765	199,501	184,216	179,89	73,981	-----	-----	158,377	<b>108,307</b>	-----	-----	<b>46,591</b>
	Y	70,125	94,71	74,595	70,485	80,755	-----	-----	29,158				
imge7	X	154,872	167,345	143,985	140,308	-----	197,492	-----	99,664	-----	<b>30,298</b>	-----	<b>40,702</b>
	Y	60,68	61,69	59,887	59,599	-----	64,707	-----	57,436				
imge8	X	77,727	77,078	78,16	165,47	57,442	-----	-----	155,738	<b>20,421</b>	-----	-----	<b>93,839</b>
	Y	44,098	44,746	44,386	128,885	46,452	-----	-----	35,552				
imge13	X	79,602	102,385	102,312	126,826	43,704	-----	-----	135,499	<b>38,974</b>	-----	-----	<b>9,105</b>
	Y	75,893	67,241	66,592	56,643	91,068	-----	-----	53,871				
imge17	X	79,818	116,011	165,687	120,265	54,469	-----	-----	152,164	<b>25,857</b>	-----	-----	<b>32,413</b>
	Y	59,31	52,533	123,2612	51,379	64,409	-----	-----	45,632				
imge21	X	142,6152	145,643	172,464	148,239	121,233	-----	-----	148,631	<b>79,415</b>	-----	-----	<b>0,540</b>
	Y	35,951	25,136	23,261	16,268	112,433	-----	-----	16,639				
imge25	X	42,183	42,687	123,798	54,944	-----	-----	152,005	55,165	-----	-----	<b>28,781</b>	<b>0,234</b>
	Y	115,907	118,359	79,786	62,627	-----	-----	85,505	62,704				
imge29	X	146,148	146,292	126,898	146,364	69,605	-----	-----	152,585	<b>91,753</b>	-----	-----	<b>7,033</b>
	Y	58,589	58,373	70,774	58,445	109,184	-----	-----	55,164				

**Tablo 4.10:** İki Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

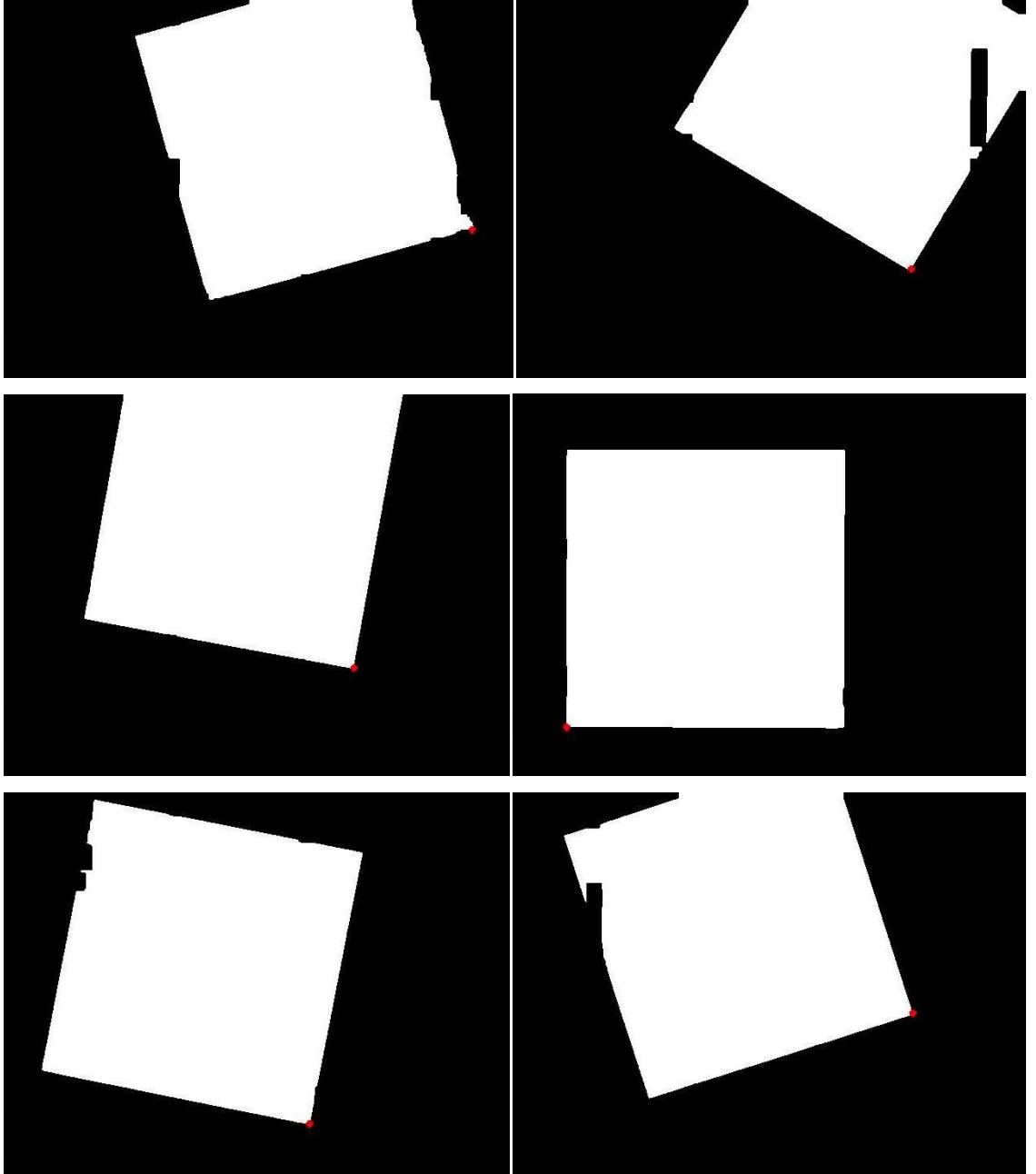
		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge1	X	167,417	180,899	184,144	183,279	63,702	-----	185,542	90,102				
	Y	108,481	54,119	49,289	50,226	114,704	-----	45,404	19,435	<b>103,902</b>	-----	<b>4,129</b>	<b>98,133</b>
imge3	X	82,774	108,441	86,739	104,403	44,819	-----	162,012	143,614				
	Y	40,276	35,662	40,06	36,816	47,197	-----	128,339	29,999	<b>38,581</b>	-----	<b>116,014</b>	<b>39,799</b>
imge6	X	42,399	64,749	78,015	64,966	34,746	-----	158,742	64,545				
	Y	95,936	17,998	22,252	18,07	111,954	-----	48,557	18,758	<b>17,752</b>	-----	<b>84,905</b>	<b>0,807</b>
imge14	X	111,829	123,005	123,005	109,667	29,201	-----	158,095	121,297				
	Y	125,785	121,675	121,098	19,728	52,271	-----	107,528	15,434	<b>110,597</b>	-----	<b>37,623</b>	<b>12,397</b>
imge18	X	84,000	82,774	82,99	77,367	36,127	-----	160,921	66,125				
	Y	22,036	21,675	21,243	19,945	110,989	-----	45,794	15,795	-----	-----	<b>81,707</b>	<b>11,984</b>
imge22	X	152,637	155,521	153,07	163,5237	77,280	91,079	-----	174,975				
	Y	125,929	125,568	125,496	124,4148	36,037	134,732	-----	22,14	<b>117,300</b>	<b>65,09</b>	-----	<b>102,914</b>
imge26	X	162,875	163,74	162,875	164,317	51,745	-----	184,665	95,845				
	Y	120,305	120,017	119,728	118,863	124,624	-----	80,064	34,864	<b>111,214</b>	-----	<b>45,255</b>	<b>108,371</b>
imge30	X	179,746	199,501	197,266	185,225	65,825	-----	198,805	108,785				
	Y	110,86	73,225	73,514	67,602	120,784	-----	74,184	31,084	<b>114,352</b>	-----	<b>1,679</b>	<b>84,715</b>

**Tablo 4.11:** Üç Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

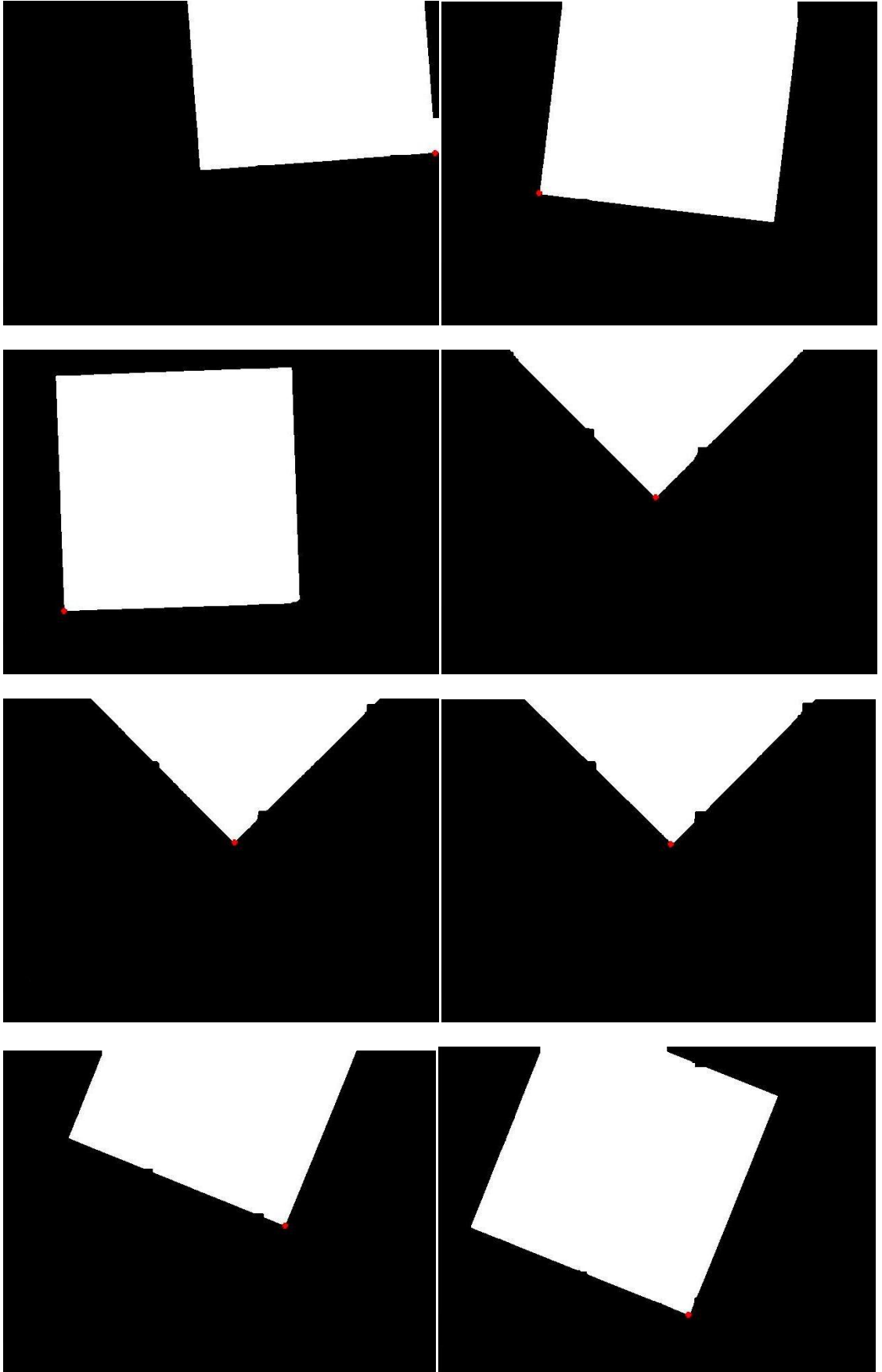
		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge4	X	34,829	42,039	134,973	133,964	34,412	34,412	135,608	134,807				
	Y	8,625	108,409	108,121	8,049	8,422	108,017	108,017	8,422	0,464	7,637	0,643	0,922
imge5	X	66,047	59,847	128,051	104,259	29,211	48,809	147,203	127,205				
	Y	122,54	123,766	109,995	115,114	28,357	125,152	107,553	8,758	101,130	11,124	19,307	108,803
imge9	X	138,866	137,641	134,612	134,108	38,307	138,003	140,503	41,807				
	Y	21,315	120,882	21,098	20,666	116,931	120,431	21,535	17,635	138,761	0,578	5,907	92,351
imge15	X	58,116	49,753	86,018	58,549	34,475	49,473	147,869	132,47				
	Y	122,54	123,478	118,286	122,108	25,428	123,723	108,724	10,33	99,948	0,372	62,586	134,010
imge19	X	106,206	137,857	128,051	48,527	35,950	135,446	141,445	42,25				
	Y	19,152	22,396	22,396	15,691	115,033	120,833	21,536	15,738	118,866	98,467	13,422	6,277
imge23	X	74,194	138,145	140,957	42,327	38,381	138,076	141,876	42,381				
	Y	13,889	115,835	15,835	12,158	112,336	115,936	16,540	12,641	104,759	0,122	1,158	0,486
imge27	X	128,124	127,475	162,514	115,939	60,725	160,465	166,345	67,445				
	Y	122,684	22,540	24,7753	21,747	117,568	124,741	25,528	18,568	67,593	107,394	3,904	48,598

**Tablo 4.12:** Dört Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Harris Yöntemi Uygulaması ile Bulunan Koordinatlar.

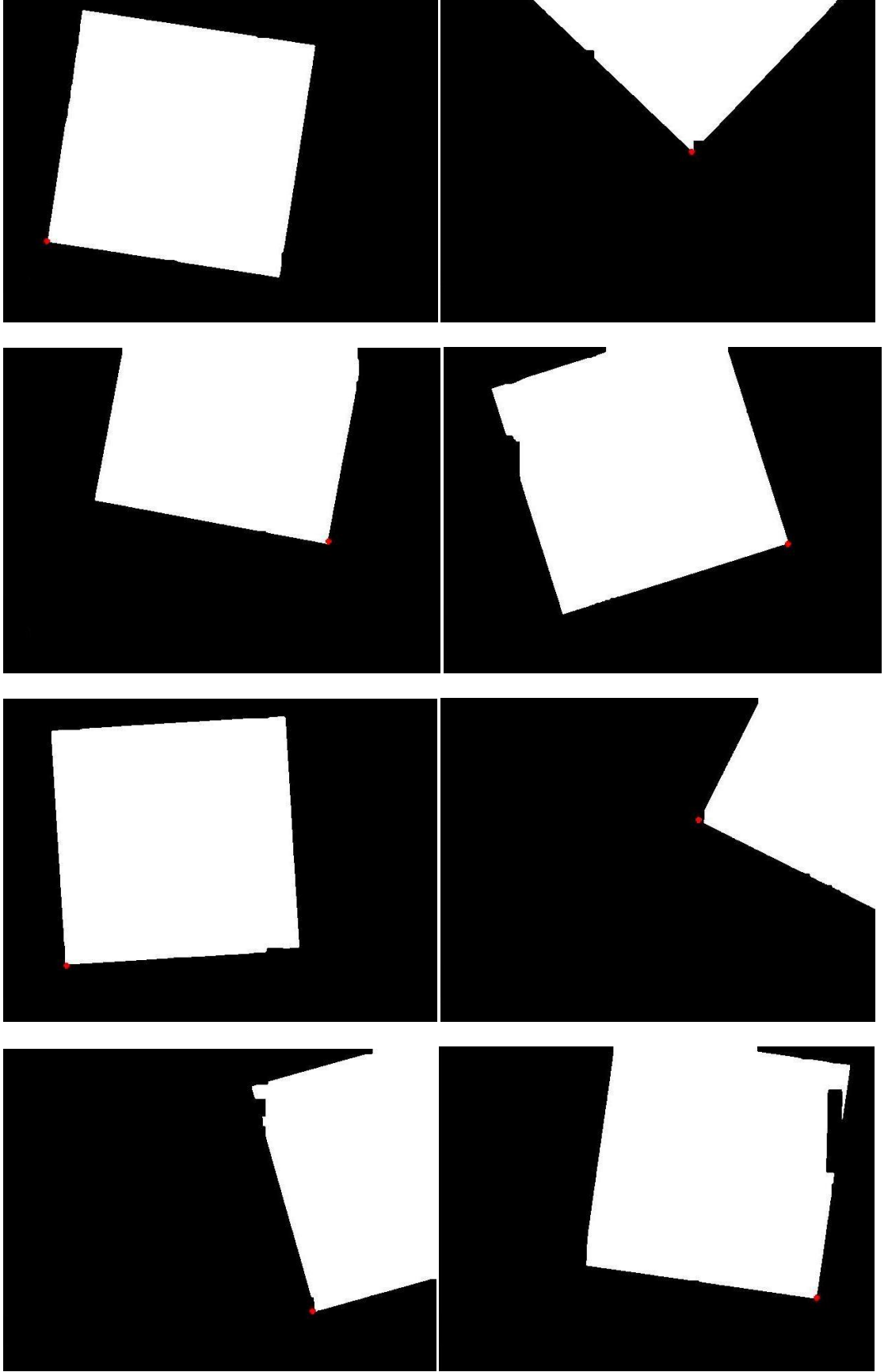
Son olarak Yönlendirilmiş Filtre – Hough Dönüşümü uygulanmış tablo ve resimler aşağıda verilmiştir.



Şekil 4.17: İlk 6 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge1-İmge6).

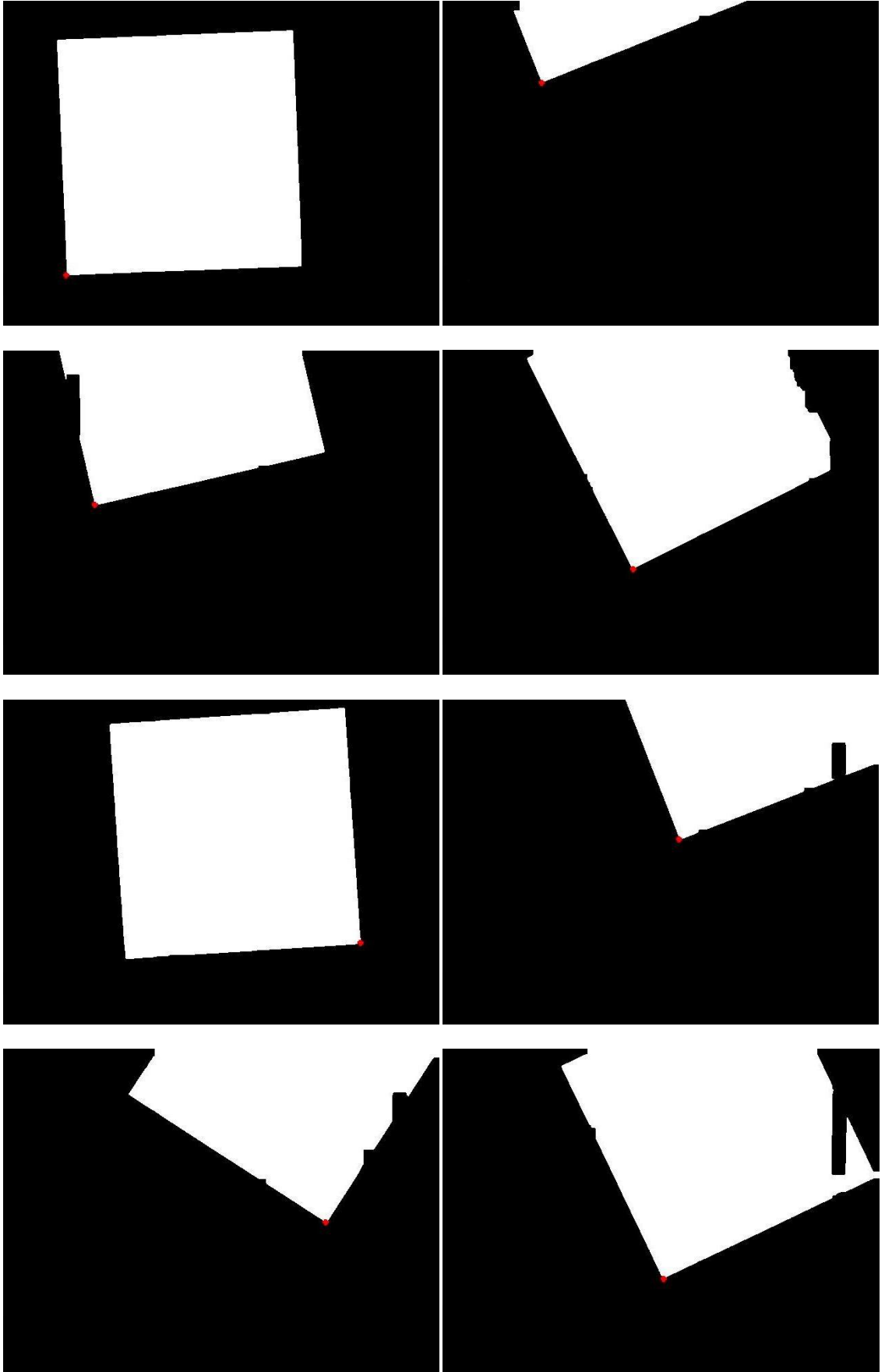


Şekil 4.18: Sonraki 8 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge7-İmge14).



Şekil 4.19: Sonraki 8 İmge için Yönlendirilmiş Filtre -Hough Uygulaması (İmge15-İmge22).





Şekil 4.20: Sonraki 8 İmge için Yönlendirilmiş Filtre -Hough Uygulaması(İmge23-İmge30).

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge10	X	105,989	-----	-----	-----	106,402	-----	-----	-----				
	Y	66,087	-----	-----	-----	65,864	-----	-----	-----	0,469	-----	-----	-----
imge11	X	113,199	-----	-----	-----	113,404	-----	-----	-----				
	Y	67,457	-----	-----	-----	67,364	-----	-----	-----	0,225	-----	-----	-----
imge12	X	112,695	-----	-----	-----	113,600	-----	-----	-----				
	Y	66,953	-----	-----	-----	67,208	-----	-----	-----	0,940	-----	-----	-----
imge16	X	121,779	-----	-----	-----	122,551	-----	-----	-----				
	Y	63,420	-----	-----	-----	63,249	-----	-----	-----	0,791	-----	-----	-----
imge20	X	125,240	-----	-----	-----	125,601	-----	-----	-----				
	Y	76,398	-----	-----	-----	76,751	-----	-----	-----	0,505	-----	-----	-----
imge24	X	57,395	-----	-----	-----	57,885	-----	-----	-----				
	Y	93,845	-----	-----	-----	93,840	-----	-----	-----	0,490	-----	-----	-----
imge28	X	115,362	-----	-----	-----	116,625	-----	-----	-----				
	Y	69,332	-----	-----	-----	68,744	-----	-----	-----	1,393	-----	-----	-----

**Tablo 4.13:** Tek Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge2	X	158,261	-----	-----	-----	158,377	73,981	-----	-----				
	Y	30,543	-----	-----	-----	29,158	80,755	-----	-----	1,390	-----	-----	-----
imge7	X	197,914	-----	-----	-----	197,492	99,664	-----	-----				
	Y	64,141	-----	-----	-----	64,707	57,436	-----	-----	0,706	-----	-----	-----
imge8	X	56,674	-----	-----	-----	57,442	155,738	-----	-----				
	Y	47,270	-----	-----	-----	46,452	35,552	-----	-----	1,122	-----	-----	-----
imge13	X	135,550	-----	-----	-----	135,499	43,704	-----	-----				
	Y	53,831	-----	-----	-----	53,871	91,068	-----	-----	0,065	-----	-----	-----
imge17	X	151,339	-----	-----	-----	152,164	54,469	-----	-----				
	Y	45,612	-----	-----	-----	45,632	64,409	-----	-----	0,825	-----	-----	-----
imge21	X	148,167	-----	-----	-----	148,631	121,233	-----	-----				
	Y	16,484	-----	-----	-----	16,639	112,433	-----	-----	0,489	-----	-----	-----
imge25	X	54,079	-----	-----	-----	55,165	152,005	-----	-----				
	Y	63,131	-----	-----	-----	62,704	85,505	-----	-----	1,167	-----	-----	-----
imge29	X	151,628	-----	-----	-----	152,585	69,605	-----	-----				
	Y	55,056	-----	-----	-----	55,164	109,184	-----	-----	0,963	-----	-----	-----

**Tablo 4.14:** İki Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge1	X	184,865	-----	-----	-----	185,542	63,702	90,102	-----				
	Y	44,819	-----	-----	-----	45,404	114,704	19,435	-----	0,895	-----	-----	-----
imge3	X	142,832	-----	-----	-----	143,614	44,819	162,012	-----				
	Y	30,111	-----	-----	-----	29,999	47,197	128,339	-----	0,790	-----	-----	-----
imge6	X	159,054	-----	-----	-----	158,742	34,746	64,545	-----				
	Y	49,072	-----	-----	-----	48,557	111,954	18,758	-----	0,602	-----	-----	-----
imge14	X	120,842	-----	-----	-----	121,297	29,201	158,095	-----				
	Y	15,330	-----	-----	-----	15,434	52,271	107,528	-----	0,467	-----	-----	-----
imge18	X	160,207	-----	-----	-----	160,921	36,127	66,125	-----				
	Y	45,612	-----	-----	-----	45,794	110,989	15,795	-----	0,737	-----	-----	-----
imge22	X	175,204	-----	-----	-----	174,975	77,280	91,079	-----				
	Y	22,180	-----	-----	-----	22,140	36,037	134,732	-----	0,232	-----	-----	-----
imge26	X	96,040	-----	-----	-----	95,845	184,665	51,745	-----				
	Y	35,806	-----	-----	-----	34,864	80,064	124,624	-----	0,962	-----	-----	-----
imge30	X	108,729	-----	-----	-----	108,785	198,805	65,825	-----				
	Y	31,120	-----	-----	-----	31,084	74,184	120,784	-----	0,067	-----	-----	-----

**Tablo 4.15:** Üç Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

		BULUNAN KOORDİNATLAR				GERÇEK KOORDİNATLAR				FARK(MM)			
RESİM ADI	KONUM TÜRÜ	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. KONUM	2. KONUM	3. KONUM	4. KONUM	1. FARK	2. FARK	3. FARK	4. FARK
imge4	X	34,901	-----	-----	-----	34,412	135,608	34,412	134,807				
	Y	8,697	-----	-----	-----	8,422	108,017	108,017	8,422	0,561	-----	-----	-----
imge5	X	126,682	-----	-----	-----	127,205	147,203	29,211	48,809				
	Y	9,202	-----	-----	-----	8,758	107,553	28,357	125,152	0,686	-----	-----	-----
imge9	X	41,101	-----	-----	-----	41,807	140,503	38,307	138,003				
	Y	18,070	-----	-----	-----	17,635	21,535	116,931	120,431	0,829	-----	-----	-----
imge15	X	33,819	-----	-----	-----	34,475	132,470	147,869	49,473				
	Y	25,785	-----	-----	-----	25,428	10,330	108,724	123,723	0,747	-----	-----	-----
imge19	X	42,183	-----	-----	-----	42,250	35,95	135,446	141,445				
	Y	15,186	-----	-----	-----	15,738	115,033	120,833	21,536	0,556	-----	-----	-----
imge23	X	41,894	-----	-----	-----	42,381	38,381	141,876	138,076				
	Y	12,591	-----	-----	-----	12,641	112,336	16,54	115,936	0,490	-----	-----	-----
imge27	X	166,408	-----	-----	-----	166,345	160,465	60,725	67,445				
	Y	25,713	-----	-----	-----	25,528	124,741	117,568	18,568	0,195	-----	-----	-----

**Tablo 4.16:** Dört Köşeli İmge Koordinatları ve Yönlendirilmiş Filtre - Hough Yöntemi Uygulaması ile Bulunan Koordinatlar.

## 5. TARTIŞMA VE SONUÇ

Tablolarda görüldüğü üzere gidilecek koordinat sayısı 4'e ulaşabilmektedir. Fakat bunların hangisine gideceği uzaklık farkı değerlerinin başarısına göre sıralanmış ve tabloda gösterilmiştir. Buna göre başarı yüzdesini hesaplamak için 1. Koordinatlar değerlerini dikkate almak daha doğru olacaktır. Böyle bir hesaplama yapılması sonucunda başarı değerleri aşağıda verilmiştir.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	14,28	85,68	0	85,68

**Tablo 5.1:** 1 mm tolerans payı dikkate alınmış tek köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	37,50	75	0	62,5

**Tablo 5.2:** 1 mm tolerans payı dikkate alınmış iki köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	37,50	50	0	100

**Tablo 5.3:** 1 mm tolerans payı dikkate alınmış üç köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	42,86	57,14	14,28	100

**Tablo 5.4:** 1 mm tolerans payı dikkate alınmış dört köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	33,33	66,96	3,33	86,67

**Tablo 5.5:** 1 mm tolerans payı dikkate alınmış tüm resimler için başarı yüzdesi.

Yukarıdaki tablolarda görülmektedir ki Yönlendirilmiş Filtre – Hough yöntemi diğer yöntemlere göre çok daha başarılıdır. Bu hesaplama uzunluk farkı 1 mm den küçük olması koşulu ile hesaplanmıştır. Fakat yukarıdaki sonuçlara bakıldığı takdirde sacın 2 nokta arası uzaklığı gerçek uzaklığa göre +- 1 derece fark ettiği görülmektedir. Bu da kullanıcıdan doğan hatayı göstermektedir. Ayrıca kullanılan aletin ışık yarıçapı referans noktası bulunurken yaklaşık 2 mm' den fazladır bu da referans noktasının yaklaşık 2 mm hata payının olduğunu göstermektedir. Ayrıca şu anda alınan hata payı 5 mm ye kadar tolerans payına sahiptir. Bu kriterler göz önüne alınıp tekrar bir başarı yüzdesi hesaplanırsa sonuçlar aşağıdaki gibi olacaktır.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	14,28	100	0	100

**Tablo 5.6:** 2 mm tolerans payı dikkate alınmış tek köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	37,50	100	0	100

**Tablo 5.7:** 2 mm tolerans payı dikkate alınmış iki köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	50	100	0	100

**Tablo 5.8:** 2 mm tolerans payı dikkate alınmış üç köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	42,84	71,4	14,28	100

**Tablo 5.9:** 2 mm tolerans payı dikkate alınmış dört köşeli resimler için başarı yüzdeleri.

KULLANILAN YÖNTEM	Harris Yöntemi	Hough Yöntemi	Yönlendirilmiş Filtre – Harris Yöntemi	Yönlendirilmiş Filtre – Hough Yöntemi
Başarı Oranı(%)	36,16	92,85	3,57	100

**Tablo 5.10:** 2 mm tolerans payı dikkate alınmış tüm resimler için başarı yüzdesi.

Aşağıda uygulama zamanları verilmiştir. Bu zamanlar göz önüne alındığında Yönlendirilmiş Filtre – Harris[19] yönteminin uygulanması zorlaşmaktadır. Çünkü hem başarı oranı düşük hem de uygulama süresi çok uzundur. Sistemimizde kullanıcının referans aldırma süresi 50 sn 60 sn civarında olduğu göz önüne alınırsa eğer Yönlendirilmiş Filtre – Harris[19] yöntemi kullanılırsa 14 sn hesaplama 1-5 sn gelme süresi dersek zaman kaybımız çok yüksek olmaktadır. Zaman bakımının en iyi olduğu Hough[20] Yöntemi bizim için başarılı bir yöntem olarak kabul edilebilir. Fakat Yönlendirilmiş Filtre – Hough yöntemi başarı oranı bakımından en yüksek olduğu için bu yöntem daha sağlıklı bulunmuştur. Ayrıca bu yöntem zaman bakımından da Yönlendirilmiş Filtre – Harris[19] yöntemi gibi çok yüksek zaman kaybına neden olmamaktadır.



<b>Resim Adı</b>	<b>Harris Yöntemi</b>	<b>Hough Yöntemi</b>	<b>Yönlendirilmiş Filtre – Harris Yöntemi</b>	<b>Yönlendirilmiş Filtre – Hough Yöntemi</b>
<b>imge1</b>	3,359	3,088	12,770	5,519
<b>imge2</b>	3,421	2,982	12,816	5,689
<b>imge3</b>	3,191	3,004	12,830	5,557
<b>imge4</b>	3,138	3,081	12,345	5,461
<b>imge5</b>	3,297	3,444	12,953	5,398
<b>imge6</b>	3,451	3,228	13,028	5,518
<b>imge7</b>	3,265	3,033	13,507	6,500
<b>imge8</b>	3,466	3,120	12,734	5,397
<b>imge9</b>	3,251	3,136	12,539	5,490
<b>imge10</b>	3,212	2,828	12,967	5,650
<b>imge11</b>	3,141	2,904	12,909	5,600
<b>imge12</b>	3,204	2,854	12,780	5,769
<b>imge13</b>	3,250	2,968	13,795	6,117
<b>imge14</b>	3,402	3,070	13,321	5,460
<b>imge15</b>	3,330	2,787	13,116	5,454
<b>imge16</b>	3,266	2,790	13,181	5,785
<b>imge17</b>	3,212	2,934	13,838	6,419
<b>imge18</b>	3,317	3,040	12,566	5,442
<b>imge19</b>	3,119	2,949	12,126	5,484
<b>imge20</b>	3,139	3,031	12,695	5,828
<b>imge21</b>	3,177	3,101	13,215	6,291
<b>imge22</b>	3,172	3,076	12,381	5,503
<b>imge23</b>	3,081	3,125	12,173	5,547
<b>imge24</b>	3,102	2,698	12,217	5,108
<b>imge25</b>	3,136	2,819	13,435	6,018
<b>imge26</b>	3,166	3,258	13,080	5,422
<b>imge27</b>	3,109	2,991	12,395	5,422
<b>imge28</b>	3,146	2,835	13,091	5,793
<b>imge29</b>	3,152	3,008	13,259	6,081
<b>imge30</b>	3,234	3,156	12,814	5,545

**Tablo 5.11:** Uygulanan yöntemlerin saniye cinsinden uygulama süreleri.

Bu makine kullanılan saclar düz ve köşeye ulaştığı varsayılmaktadır. Genelde kullanılan şekli budur. Fakat eğer sacda ovallık varsa veya kameranin göremediği yere konulmuşsa bu yöntem geçerliliğini kaybedecektir. Bu koşullar için ise kameranin görüş alanı oldukça büyük olmalıdır. Fakat bu makinelerin en ve boy oranları genişletilebilir olduğundan uygulanabilirliği düşük olmaktadır. Bu yüzden sadece genelde kullanılan büyük ve düz saclar için bu yöntem önerilmiştir. Bu saclar için kullanılabilirliği gösterilmiştir.

## KAYNAKLAR

- [1]. H. P. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Proc. Int'l Joint Conf. Artificial Intelligence*, p. 584, 1977.
- [2]. C. Harris, and M. Stephens, "Combined Edge and Corner Detection," *Fourth Alvey Vision Conference*, pp. 147-151, 1988.
- [3]. P. R. Baudet, "Rotational invariant image operators," in *4th International Conference on Pattern Recognition*, Tokyo, Japan, 1978, pp. 579-583.
- [4]. L. Kitchen, and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, 1(1), pp. 8, Dec, 1982.
- [5]. D. M. Tsai, "Boundary-based corner detection using neural networks," *Pattern Recognition*, 30(1), pp. 85-97, Jan, 1997.
- [6]. Q. Ji, and R. M. Haralick, "Breakpoint detection using covariance propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), pp. 845-851, Aug, 1998.
- [7]. J. Q. Fang, and T. S. Huang, "A Corner Finding Algorithm for Image Analysis and Registration," in *AAAI Conference*, 1982, pp. 46-49.
- [8]. Y. Zhang, and P. I. Rockett, "The Bayesian operating point of the canny edge detector," *IEEE Transactions on Image Processing*, 15(11), pp. 3409-3416, Nov, 2006.
- [9]. Z. O. Wu, and A. Rosenfeld, "Filtered Projections as an Aid in Corner Detection," *Pattern Recognition*, 16(1), pp. 31-38, 1983.
- [11]. K. Paler, J. Föglein, J. Illingworth et al., "Local ordered grey levels as an aid to corner detection," *Pattern Recognition*, 17(5), pp. 535-543, 1984.
- [12]. K. Rangarajan, M. Shah, and D. van Brackle, "Optimal Corner Detector," in *Computer Vision. Second International Conference on*, 1988, pp. 90-94.
- [13]. F. Arrebola, A. Bandera, P. Camacho et al., "Corner detection by local histograms of contour chain code," *Electronics Letters*, 33(21), pp. 1769- 1771, Oct 9, 1997.
- [14]. K. Sohn, J. H. Kim, and W. E. Alexander, "mean field annealing approach to robust corner detection," *IEEE Transactions on systems Man and Cybernetics: Part B*, 28(1), pp. 82-90, 1998, 1998.
- [15]. K. Kohlmann, "Corner detection in natural images based on the 2-D Hilbert transform," *Signal Processing*, 48(3), pp. 225-234, Feb, 1996.

- [16]. E. R. Davies, "Application of the Generalized Hough Transform to Corner Detection," *Iee Proceedings-E Computers and Digital Techniques*, 135(1), pp. 49-54, Jan, 1988.
- [17]. S. M. Smith, and J. M. Brady, "SUSAN - A new approach to low level image processing," *International Journal of Computer Vision*, 23(1), pp. 45- 78, May, 1997.
- [18]. A. Rosenfeld, "Fuzzy Digital Topology," *Information and Control*, vol. 1, pp. 76-87, January 1979, 1979.
- [19]. Mahesh, Dr.M.V.Subramanyam, "Corner Detection Enhancement Using Steerable filters," *International Journal of Scientific & Engineering Research Volume 3*, Issue 8, August 2012.
- [20]. Paul V. C. Hough.: 'Method and means for recognising complex patterns', *US patent 3069654*, 1962.
- [21]. Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM*, Vol. 15, pp. 11–15 (January, 1972).
- [22]. Atilla ÖZMEN, Osman N. UÇAN, A.Muhittin ALBORA, "Yönlendirmeli Fitreler Yardımıyla Konya Bölgesi civarındaki Gömülü Fayların Tespiti ," *Doğuş Üniversitesi Dergisi*, 4 (1), 107-116, 2003.
- [23]. D.Parks and J.P. Gravel. Corner Detection. <http://www.cim.mc-gill.ca/dparks/CornerDetector/harris.ht>, pp.15-30.
- [24]. M.Fatih Aydoğdu, Fpga-Tabanlı, Steryo Görme Tabanlı Robotta 3 Boyutlu Eş Zamanlı Konumlama ve Haritalama, *TOBB Ekonomi ve Teknoloji Üniversitesi Yüksek Lisans Tezi*, 2010.

## EKLER

### Harris.m

```

clc;

for i=1:30
    tic
    i
    str=strcat('dene',num2str(i));
    str=strcat(str, '.jpg');

    I=imread('referans1.jpg');
    I1=imread(str);
    I=im2double(I);
    I1=im2double(I1);
    IF=I1-I;

    level=graythresh(IF);
    BW=im2bw(IF,level);

    BW=BW(:,:,1);
    BW(:,1:144)=0;
    se = strel('square',7);
    BW= imerode(BW,se);
    se = strel('square',77);
    BW=imclose(BW,se);
    MF=medfilt2(BW,[3 3]);
    MF=medfilt2(MF,[3 3]);
    MF=medfilt2(MF,[5 5]);
    MF=medfilt2(MF,[5 5]);
    MF=medfilt2(MF,[7 7]);

    MF=imfill(MF,'holes');

    figure();
    imshow(MF);

    MF=im2double(MF);
    [a,b,c]=size(I);

    MF1=MF;
    %MF2=MF.*I(:,:,2);
    %MF3=MF.*I(:,:,3);

    X1=conv2(MF1,[-1 0 1]);
    Y1=conv2(MF1,[-1 0 1]');

    W=gausswin(5);
    w=conv2(W,W');

```

```

A1=conv2(X1.^2,w);
B1=conv2(Y1.^2,w);
C1=conv2((X1(1:a,1:b).*Y1(1:a,1:b)),w);

Det1=(A1(1:a,1:b).*B1(1:a,1:b)-(C1(1:a,1:b).^2)-
0.04*((A1(1:a,1:b)+B1(1:a,1:b)).^2));

k=0;
md=0;
[ix,iy]=size(Det1);
max(max(Det1));
for i=1:ix
    for j=1:iy;
        if(Det1(i,j)>0)
            k=k+1;
            md(k)=Det1(i,j);
            m(k,:)=[i,j];
        else
            %Det1(i,j)=0;
        end
    end
end

[ak,sira]=sort(md,'descend');

for g=1:4
    hold on;
    plot(m(sira(g),2),m(sira(g),1),'r*','LineWidth',3);
    X(g)=m(sira(g),2);
    Y(g)=m(sira(g),1);
end
g=1;
hold on;
plot(m(sira(g),2),m(sira(g),1),'g*','LineWidth',3);
X(g)=m(sira(g),2);
Y(g)=m(sira(g),1);
X1=(X-144)*100/1387)+25.528
Y1=((1777-Y)*100/1387)+1.055
toc
clear all;

end

```

## Hough.m

```

function varargout = deneme_son(varargin)
% DENEME_SON MATLAB code for deneme_son.fig
%     DENEME_SON, by itself, creates a new DENEME_SON or raises the
existing
%     singleton*.
%
%     H = DENEME_SON returns the handle to a new DENEME_SON or the
handle to
%     the existing singleton*.
%
%     DENEME_SON('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in DENEME_SON.M with the given input
arguments.
%
%     DENEME_SON('Property','Value',...) creates a new DENEME_SON or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before deneme_son_OpeningFcn gets called. An
unrecognized
%     property name or invalid value makes property
application
%     stop. All inputs are passed to deneme_son_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help deneme_son

% Last Modified by GUIDE v2.5 19-Oct-2013 23:46:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @deneme_son_OpeningFcn, ...
                  'gui_OutputFcn',  @deneme_son_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before deneme_son is made visible.
function deneme_son_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to deneme_son (see VARARGIN)
%handles.referans_pixel_sayisi=1369;
%handles.x_bas_mm=82.442;
%handles.y_bas_mm=6.318;
%handles.y_bas_piksel=1763;
%handles.x_bas_piksel=327;

handles.referans_pixel_sayisi=1387;
handles.x_bas_mm=25.528;
handles.y_bas_mm=1.055;
handles.y_bas_piksel=1775;
handles.x_bas_piksel=144;

set(handles.edit2, 'String', '1387');
set(handles.edit3, 'String', '25.528');
set(handles.edit4, 'String', '1.055');
set(handles.edit5, 'String', '144');
set(handles.edit6, 'String', '1777');

handles.resim='denel.jpg';
handles.IBOS='referans1.jpg';
%handles.IBOS='re2.jpg';
handles.IBOS=imread(handles.IBOS);
handles.IF=imread(handles.resim);
%handles.IF=imresize(handles.IF,2,'bicubic');
%handles.IBOS=imresize(handles.IBOS,2,'bicubic');
handles.IBOS=im2double(handles.IBOS);
handles.IT=im2double(handles.IF);
handles.IF=handles.IT-handles.IBOS;
set(handles.text1, 'Visible', 'off')
set(handles.text2, 'Visible', 'off')
set(handles.text3, 'Visible', 'off')
set(handles.text4, 'Visible', 'off')
set(handles.text5, 'Visible', 'off')

set(handles.edit2, 'Visible', 'off')
set(handles.edit3, 'Visible', 'off')
set(handles.edit4, 'Visible', 'off')
set(handles.edit5, 'Visible', 'off')
set(handles.edit6, 'Visible', 'off')

set(handles.text10, 'Visible', 'off')
set(handles.text9, 'Visible', 'off')
set(handles.text8, 'Visible', 'off')
set(handles.text7, 'Visible', 'off')
set(handles.text6, 'Visible', 'off')

set(handles.edit1, 'String', handles.resim);
handles.bool=1;

```



```

handles.ayarlar=0;
%18.310 0.999
% Update handles structure

handles.output = hObject;
guidata(hObject, handles);
axes(handles.axes2);
imshow(handles.IF);
axes(handles.axes1);
imshow(handles.IT);
%end
% UIWAIT makes deneme_son wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = deneme_son_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
tic
if(isequal(handles.resim,get(handles.edit1,'String')) &
(handles.bool==0))
    h=msgbox('Resmi Degistirmeden Tekrar Ölçüm Yapamazsiniz...!');
else
    handles.bool=0;
    handles.resim=get(handles.edit1,'String')
    handles.IF=imread(handles.resim);
    %handles.IF=imresize(handles.IF,2,'bicubic');
    handles.IT=im2double(handles.IF);
    %handles.IF=handles.IF*21-handles.IBOS*15;
    handles.IF=handles.IT-handles.IBOS;
    axes(handles.axes1);
    imshow(handles.IT);

level=graythresh(handles.IF);

```

```

BW=im2bw(handles.IF,level);

%BW=edge(BW,'prewitt');
%axes(handles.axes2);
%imshow(BW);

%se = strel('square',5);
%BW = imdilate(BW,se);

%BW=imclose(BW,se);

se = strel('square',7);
BW = imerode(BW,se);
se = strel('square',77);
BW=imclose(BW,se);

axes(handles.axes2);
imshow(BW);

MF=medfilt2(BW,[3 3]);
MF=medfilt2(MF,[3 3]);
MF=medfilt2(MF,[5 5]);
MF=medfilt2(MF,[5 5]);
MF=medfilt2(MF,[7 7]);

%MF = edge(MF,'canny');
MF=imfill(MF,'holes');
%figure();
axes(handles.axes2);
imshow(MF);

%H=[0,1,0;1,-4,1;0,1,0];

F=edge(MF,'canny');

% MF=im2double(MF);
%F=conv2(H,MF);

%[m,n]=size(F);
%for l = 1:m
%   for k = 1:n
%       if (l<5 || k>n-5 || l>m-5 || k<5)
%           F(l, k) = 0;
%       end
%   end
%end

[H,theta,rho] = hough(F);
P = houghpeaks(H,5,'threshold',ceil(0.35*max(H(:)))));
lines = houghlines(F,theta,rho,P,'FillGap',15,'MinLength',20);
lines2 = houghlines(F,theta,rho,P,'FillGap',5,'MinLength',10);
[a,b]=size(lines);

```

```

[a1,b1]=size(lines);
for i=1:b1
lines(b+i).point1 = lines2(i).point1;
lines(b+i).point2 = lines2(i).point2;
end
R=handles.IF;
%axes(handles.axes2);7
MF(:,1:149)=0;

max_len = 0;
%----for k = 1:length(lines)
%----    xy = [lines(k).point1; lines(k).point2];
%----    plot(xy(:,1),xy(:,2), 'LineWidth',2, 'Color','green');

% Plot beginnings and ends of lines
%plot(xy(1,1),xy(1,2), 'x', 'LineWidth',2, 'Color','yellow');
%plot(xy(2,1),xy(2,2), 'x', 'LineWidth',2, 'Color','red');

% Determine the endpoints of the longest line segment
%----    len(k) = norm(lines(k).point1 - lines(k).point2);
%----    plot(xy(:,1),xy(:,2), 'LineWidth',2, 'Color','red');

%----end

% highlight the longest line segment

i=0;
d=0;
    kosex=zeros(length(lines),length(lines));
    kosey=zeros(length(lines),length(lines));
for k = 1:length(lines)-1
for l = 1:length(lines)-1

    d=d+1;
    xy1 = [lines(k).point1; lines(k).point2];
    m1 = (xy1(1,2)-xy1(2,2))/(xy1(1,1)-xy1(2,1));
    Q=180*atan(m1)/pi;
    xy2 = [lines(l).point1; lines(l).point2];
    m=(xy2(1,2)-xy2(2,2))/(xy2(1,1)-xy2(2,1));
    Q1=180*atan(m)/pi;
    Sonuc(d) = abs((Q)-(Q1));
%for r = 1:length(lines)
%xy = [lines(r).point1; lines(r).point2]
%imshow(F)
%hold on;
%plot(xy(:,1),xy(:,2), 'LineWidth',2, 'Color','red');
%end

if ((Sonuc(d)>88.0) && (Sonuc(d)<92))
i=i+1;
d
    %hold on
    %plot(xy1(1,1),xy1(1,2), 'O', 'LineWidth',1, 'Color','blue');
    % plot(xy2(1,1),xy2(1,2), 'O', 'LineWidth',1, 'Color','blue');

```

```

%plot(xy2(1,1),xy2(1,2),'o','LineWidth',1,'Color','blue');
%plot(xy2(2,1),xy2(2,2),'o','LineWidth',1,'Color','blue');

kose=((xy1(2,2)-xy1(1,2))* (xy1(1,1)-xy2(1,1))- (xy1(1,2)-
xy2(1,2))* (xy1(2,1)-xy1(1,1))) / ((xy1(2,2)-xy1(1,2))* (xy2(2,1)-
xy2(1,1))- (xy2(2,2)-xy2(1,2))* (xy1(2,1)-xy1(1,1)));

kosex(1,i)=xy2(1,1)+(xy2(2,1)-xy2(1,1))*kose;
kosey(1,i)=xy2(1,2)+(xy2(2,2)-xy2(1,2))*kose;

kosex(2,i)=Sonuc(d);
kosey(2,i)=Sonuc(d);

%plot(round(a1(1)),round(a1(2)),'x','LineWidth',1,'Color','cyan');
%plot(round(a2(1)),round(a2(2)),'x','LineWidth',1,'Color','cyan');
%plot(round(a3(1)),round(a3(2)),'x','LineWidth',1,'Color','cyan');
%plot(round(a4(1)),round(a4(2)),'x','LineWidth',1,'Color','cyan');

%plot(round(median(kosex(1,6))),round(median(kosey(1,6))),'x','LineWid
th',1,'Color','cyan');

%plot(round(median(kosex(1,5))),round(median(kosey(1,5))),'x','LineWid
th',1,'Color','cyan');

end
end
end
kose2x=0;
kose2y=0;
kose3x=0;
kose3y=0;
kose4x=0;
kose4y=0;
uzaklik1x=0;
uzaklik1y=0;
uzaklik2x=0;
uzaklik2y=0;
uzaklik3x=0;
uzaklik3y=0;
uzaklik4x=0;
uzaklik4y=0;
aci1=0;
aci2=0;
aci3=0;
aci4=0;

```

```

[B, IX] = sort(abs(90-kosex(2,:)), 2);

kose1x=kosex(1,IX(1));
kose1y=kosey(1,IX(1));
aci1 =kosex(2,IX(1));
kosex;
kosey;
hold on
plot(kose1x,kose1y,'*', 'LineWidth',3, 'Color','red');

for sayici=2:2:i
    if(kosex(1,IX(sayici))>0 & kosey(1,IX(sayici))>0)
        if(abs(kosex(1,IX(sayici))- kose1x) > 50 ||
abs(kosey(1,IX(sayici))- kose1y) > 50 )
            kose2x=kosex(1,IX(sayici));
            kose2y=kosey(1,IX(sayici));
            aci2=kosex(2,IX(sayici));

            hold on
            plot(kose2x,kose2y,'*', 'LineWidth',3, 'Color','red');

            break;
        end
    end
end

for sayici=4:2:i
    if(kosex(1,IX(sayici))>0 & kosey(1,IX(sayici))>0)
        if((abs(kosex(1,IX(sayici))- kose1x) > 50 ||
abs(kosey(1,IX(sayici))- kose1y) > 50 ) && (abs(kosex(1,IX(sayici))-
kose2x) > 50 || abs(kosey(1,IX(sayici))- kose2y) > 50 ))
            kose3x=kosex(1,IX(sayici));
            kose3y=kosey(1,IX(sayici));
            aci3=kosex(2,IX(sayici));
            hold on
            plot(kose3x,kose3y,'*', 'LineWidth',3, 'Color','red');

            break;
        end
    end
end

for sayici=4:2:i
    if(kosex(1,IX(sayici))>0 & kosey(1,IX(sayici))>0)
        if((abs(kosex(1,IX(sayici))- kose1x) > 50 ||
abs(kosey(1,IX(sayici))- kose1y) > 50 ) && (abs(kosex(1,IX(sayici))-
kose2x) > 50 || abs(kosey(1,IX(sayici) )- kose2y) > 50 )&&
(abs(kosex(1,IX(sayici))- kose3x) > 50 || abs(kosey(1,IX(sayici) )-
kose3y) > 50 ))
            kose4x=kosex(1,IX(sayici));
            kose4y=kosey(1,IX(sayici));
            aci4=kosex(2,IX(sayici));
            hold on
            plot(kose4x,kose4y,'*', 'LineWidth',3, 'Color','red');

```

```

        break;
    end
end
end
[kose1x kose2x kose3x kose4x;
 kose1y kose2y kose3y kose4y;
 aci1 aci2 aci3 aci4];

%uzakliky=100*(round(median(kose2x)))/(handles.referans_pixel_sayisi)
%uzaklikx=100*(round(median(kose1x)))/(handles.referans_pixel_sayisi)
if((kose1y<handles.y_bas_piksel)&(kose1x>handles.x_bas_piksel))
uzakliky=100*(handles.y_bas_piksel
kose1y)/handles.referans_pixel_sayisi;
uzaklikx=100*(kose1x
handles.x_bas_piksel)/handles.referans_pixel_sayisi;

uzaklik1x=uzaklikx + handles.x_bas_mm;
uzaklik1y=uzakliky + handles.y_bas_mm;
end

if((kose2y<handles.y_bas_piksel)&(kose2x>handles.x_bas_piksel))
uzakliky=100*(handles.y_bas_piksel
kose2y)/handles.referans_pixel_sayisi;
uzaklikx=100*(kose2x
handles.x_bas_piksel)/handles.referans_pixel_sayisi ;

uzaklik2x=uzaklikx + handles.x_bas_mm;
uzaklik2y=uzakliky + handles.y_bas_mm;
end

if((kose3y<handles.y_bas_piksel)&(kose3x>handles.x_bas_piksel))
uzakliky=100*(handles.y_bas_piksel
kose3y)/handles.referans_pixel_sayisi;
uzaklikx=100*(kose3x
handles.x_bas_piksel)/handles.referans_pixel_sayisi;

uzaklik3x=uzaklikx + handles.x_bas_mm;
uzaklik3y=uzakliky + handles.y_bas_mm;
end

if((kose4y<handles.y_bas_piksel)&(kose4x>handles.x_bas_piksel))
uzakliky=100*(handles.y_bas_piksel
kose4y)/handles.referans_pixel_sayisi;
uzaklikx=100*(kose4x
handles.x_bas_piksel)/handles.referans_pixel_sayisi;

uzaklik4x=uzaklikx + handles.x_bas_mm;
uzaklik4y=uzakliky + handles.y_bas_mm;
end

A=[uzaklik1x uzaklik2x uzaklik3x uzaklik4x;
    uzaklik1y uzaklik2y uzaklik3y uzaklik4y];

s1x=strcat('X = ',num2str(uzaklik1x));
s1y=strcat(' Y = ',num2str(uzaklik1y));
s1=strcat(s1x,s1y);

set(handles.text1, 'Visible', 'on')

```

```

set(handles.text1, 'String', s1);

s2x=strcat('X = ', num2str(uzaklik2x));
s2y=strcat(' Y = ', num2str(uzaklik2y));
s2=strcat(s2x, s2y);

set(handles.text2, 'Visible', 'on')
set(handles.text2, 'String', s2);

s3x=strcat('X = ', num2str(uzaklik3x));
s3y=strcat(' Y = ', num2str(uzaklik3y));
s3=strcat(s3x, s3y);

set(handles.text3, 'Visible', 'on')
set(handles.text3, 'String', s3);

s4x=strcat('X = ', num2str(uzaklik4x));
s4y=strcat(' Y = ', num2str(uzaklik4y));
s4=strcat(s4x, s4y);

set(handles.text4, 'Visible', 'on')
set(handles.text4, 'String', s4);

s4x=strcat('X = ', num2str(uzaklik4x));
s4y=strcat(' Y = ', num2str(uzaklik4y));
s4=strcat(s4x, s4y);

set(handles.text4, 'Visible', 'on')
set(handles.text4, 'String', s4);
toc

Q1%if((uzaklik1x~=0)&(uzaklik1y~=0)&(uzaklik2x~=0)&(uzaklik2y~=0))
Q=180*atan((abs(uzaklik1x-uzaklik2x)/abs(uzaklik1y-uzaklik2y)))/pi
s5=strcat('AÇI = ', num2str(Q));
set(handles.text5, 'Visible', 'on')
set(handles.text5, 'String', s5);

%end
end

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function text1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit1 as text

```

```
%          str2double(get(hObject,'String')) returns contents of edit1
as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: place code in OpeningFcn to populate axes2
```

```
% --- Executes during object creation, after setting all properties.
function text2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% --- Executes during object creation, after setting all properties.
function text3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% --- Executes during object creation, after setting all properties.
function text4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% --- Executes during object creation, after setting all properties.
function text5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```



```

% handles      empty - handles not created until after all CreateFcns
called

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
if(handles.ayarlar==0)
set(handles.edit2, 'Visible', 'on')
set(handles.edit3, 'Visible', 'on')
set(handles.edit4, 'Visible', 'on')
set(handles.edit5, 'Visible', 'on')
set(handles.edit6, 'Visible', 'on')

set(handles.text10, 'Visible', 'on')
set(handles.text9, 'Visible', 'on')
set(handles.text8, 'Visible', 'on')
set(handles.text7, 'Visible', 'on')
set(handles.text6, 'Visible', 'on')

set(handles.pushbutton2, 'String', 'AYARLARI KAYDET');
handles.ayarlar=1;
else
set(handles.edit2, 'Visible', 'off')
set(handles.edit3, 'Visible', 'off')
set(handles.edit4, 'Visible', 'off')
set(handles.edit5, 'Visible', 'off')
set(handles.edit6, 'Visible', 'off')

set(handles.text10, 'Visible', 'off')
set(handles.text9, 'Visible', 'off')
set(handles.text8, 'Visible', 'off')
set(handles.text7, 'Visible', 'off')
set(handles.text6, 'Visible', 'off')

handles.referans_pixel_sayisi=str2double(get(handles.edit2, 'String'));
handles.x_bas_mm=str2double(get(handles.edit3, 'String'));
handles.y_bas_mm=str2double(get(handles.edit4, 'String'));
handles.x_bas_piksel=str2double(get(handles.edit5, 'String'));
handles.y_bas_piksel=str2double(get(handles.edit6, 'String'));

set(handles.pushbutton2, 'String', 'AYARLAR');
handles.ayarlar=0;
end

datacursormode on
menu = findall(get(gcf, 'Children'), 'Type', 'uicontextmenu');
menuCallback = get(menu, 'Callback');
dataCursor = menuCallback{2};

% Get the coordinates if a datatip exists.

```

```

info = getCursorInfo(dataCursor);
if ~isempty(info)
disp(info.Position)
set(handles.edit5, 'String', info.Position(1));
set(handles.edit6, 'String', info.Position(2));
datacursormode off
end

guidata(hObject, handles);

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit2 as text
%        str2double(get(hObject, 'String')) returns contents of edit2
%        as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%        called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit3 as text
%        str2double(get(hObject, 'String')) returns contents of edit3
%        as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%        called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of edit4
as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of edit5
as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6
as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function text6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function text7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function text8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function text9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

```

```

% --- Executes during object creation, after setting all properties.
function text10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

### Yönlendirilebilir\_Harris.m

```

clc;

clear all;

for i=1:30

tic
i
str=strcat('dene',num2str(i));
str=strcat(str, '.jpg');
%for i = [1:length(theta)]
% [J,H] = steerGauss([],theta(i),3,true);
% filters{i} = H;
% pause(0.1);
%end
theta = [200:20:360];

I=imread('referans1.jpg');
I1=imread(str);
I=im2double(I);
I1=im2double(I1);
I=I1-I;

```

```

level=graythresh(I);
I=im2bw(I,level);

I(:,1:144)=0;
se = strel('square',7);
I = imerode(I,se);
se = strel('square',77);
I=imclose(I,se);
MF=medfilt2(I,[3 3]);
MF=medfilt2(MF,[3 3]);
MF=medfilt2(MF,[5 5]);
MF=medfilt2(MF,[5 5]);
MF=medfilt2(MF,[7 7]);

MF=imfill(MF,'holes');
figure();
imshow(MF);

MF=im2double(MF);
[a,b,c]=size(I);

MF1=MF;
md=0;
%MF2=MF.*I(:, :, 2);
%MF3=MF.*I(:, :, 3);
for it = 1:length(theta)

    [J,H] = steerGauss(MF1,theta(it),3,true);
    filters{it} = H;

    level=graythresh(J);
    J=im2bw(J,level);

    JD=im2double(J);
    X1=conv2(JD,[-2 ,0, 2]);
    Y1=conv2(JD,[-2 ,0, 2]');

W=gausswin(5);
w=conv2(W,W');

A1=conv2(X1.^2,w);
B1=conv2(Y1.^2,w);
C1=conv2((X1(1:a,1:b).*Y1(1:a,1:b)),w);

Det1=(A1(1:a,1:b).*B1(1:a,1:b)-(C1(1:a,1:b).^2)-
0.04*((A1(1:a,1:b)+B1(1:a,1:b)).^2));
k=0;

[ix,iy]=size(Det1);
max(max(Det1));
for i=1:ix
    for j=1:iy;
        if(Det1(i,j)>0)
            k=k+1;
        end
    end
end

```

```

        md((it-1)*length(theta)+k)=Det1(i,j);
        m((it-1)*length(theta)+k,:)=[i,j];
    else
        %Det1(i,j)=0;
    end
end
end

end

[ak,sira]=sort(md,'descend');

for g=1:4
    hold on;
    plot(m(sira(g),2),m(sira(g),1),'r*','LineWidth',3);
    X(g)=m(sira(g),2);
    Y(g)=m(sira(g),1);
end
g=1;
hold on;
plot(m(sira(g),2),m(sira(g),1),'g*','LineWidth',3);
X(g)=m(sira(g),2);
Y(g)=m(sira(g),1);

X1=((X-144)*100/1387)+25.528
Y1=((1777-Y)*100/1387)+1.055
toc

clear all
end;
Yönlendirilebilir_Hough.m

% Clear Matlab command window.
clc;

clear all;
for i=1:1
    tic
    i
    str=strcat('dene',num2str(i));
    str=strcat(str,'.jpg');
    % Example #1: Basic usage.
    % Note: Filters are computed for each run.

    %disp('Example #1: Basic Usage');
    theta = [200:20:360];
    %for i = [1:length(theta)]

```

```

% [J,H] = steerGauss([],theta(i),3,true);
% filters{i} = H;
% pause(0.1);
%end
%theta=340

I=imread('referans1.jpg');
I1=imread(str);
I=im2double(I);
I1=im2double(I1);
I=I1-I;

level=graythresh(I);
I=im2bw(I,level);

se = strel('square',7);
I = imerode(I,se);
se = strel('square',77);
I=imclose(I,se);
MF=medfilt2(I,[3 3]);
MF=medfilt2(MF,[3 3]);
MF=medfilt2(MF,[5 5]);
MF=medfilt2(MF,[5 5]);
I=medfilt2(MF,[7 7]);

I=imfill(I,'holes');

pixel_degeri=zeros(1,length(theta));

for it = 1:length(theta)

    [J,H] = steerGauss(I,theta(it),3,true);
    filters{it} = H;

    level=graythresh(J);
    J=im2bw(J,level);
    %J=im2bw(J,0);
    %J(1:10,1:2560)=0;
    [HT,thet,rho] = hough(J);
    P = houghpeaks(HT,5,'threshold',ceil(0.35*max(HT(:)))));

    lines = houghlines(J,thet,rho,P,'FillGap',15,'MinLength',20);
    lines2 = houghlines(J,thet,rho,P,'FillGap',5,'MinLength',10);
    [a,b]=size(lines);
    [a1,b1]=size(lines2);
    for i=1:b1
        lines(b+i).point1 = lines2(i).point1;
        lines(b+i).point2 = lines2(i).point2;
    end
    %Create a plot that superimposes the lines on the original image.
    %for k = 1:length(lines)
        %xy = [lines(k).point1; lines(k).point2];
%imshow(I)
% hold on;
% plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');

```



```

%%end
xy1=0;
xy2=0;

kosex=zeros(length(lines),length(lines));
kosey=zeros(length(lines),length(lines));
sa=0;
%%---imshow(J);
for k = 1:length(lines)-1
for l = 1:length(lines)-1

xy1 = [lines(k).point1; lines(k).point2];
m1 = (xy1(1,2)-xy1(2,2))/(xy1(1,1)-xy1(2,1));
Q=180*atan(m1)/pi;
xy2 = [lines(l).point1; lines(l).point2];
m=(xy2(1,2)-xy2(2,2))/(xy2(1,1)-xy2(2,1));
Q1=180*atan(m)/pi;
Sonuc = abs((Q)-(Q1));
Q;
Q1;

if ((Sonuc>85) && (Sonuc<95))
sa=sa+1;

% plot(xy1(1,1),xy1(1,2),'*','LineWidth',1,'Color','green');
% plot(xy1(2,1),xy1(2,2),'*','LineWidth',1,'Color','green');

% plot(xy2(1,1),xy2(1,2),'o','LineWidth',1,'Color','blue');
% plot(xy2(2,1),xy2(2,2),'o','LineWidth',1,'Color','blue');

kose=((xy1(2,2)- xy1(1,2))* (xy1(1,1)-xy2(1,1))- (xy1(1,2)-
xy2(1,2))* (xy1(2,1)-xy1(1,1))) / ((xy1(2,2)-xy1(1,2))* (xy2(2,1)-
xy2(1,1))- (xy2(2,2)-xy2(1,2))* (xy1(2,1)-xy1(1,1)));

kosex(1,sa)=xy2(1,1)+(xy2(2,1)-xy2(1,1))*kose;
kosey(1,sa)=xy2(1,2)+(xy2(2,2)-xy2(1,2))*kose;

kosex(2,sa)=Sonuc;
kosey(2,sa)=Sonuc;
Sonuc;
Q;
Q1;

%plot(round(a1(1)),round(a1(2)),'x','LineWidth',1,'Color','cyan');
%plot(round(a2(1)),round(a2(2)),'x','LineWidth',1,'Color','cyan');
%plot(round(a3(1)),round(a3(2)),'x','LineWidth',1,'Color','cyan');
%plot(round(a4(1)),round(a4(2)),'x','LineWidth',1,'Color','cyan');
DX(sa+it) = round(median(kosex(1,sa)));
DY(sa+it) = round(median(kosey(1,sa)));
DS(sa+it) = Sonuc;

```

```

%plot(round(median(kosex(1,6))),round(median(kosey(1,6))),'x','LineWid
th',1,'Color','cyan');

%plot(round(median(kosex(1,5))),round(median(kosey(1,5))),'x','LineWid
th',1,'Color','cyan');
    %%----hold on;
    %%---
plot(round(median(kosex(1,sa))),round(median(kosey(1,sa))),'x','LineWi
dth',3,'Color','red');

%plot(round(median(kosex(1,2))),round(median(kosey(1,2))),'x','LineWid
th',1,'Color','cyan');
    %%---title(['Filtered Image (\theta = ',num2str(Sonuc),'{\circ})']);

    end
    end
    end

end

[a b]=sort(abs(DS-90),'ascend');

X=((DX(b(1))-144)*100/1387)+25.528
Y=((1777-DY(b(1)))*100/1387)+1.055

toc
figure();imshow(I);
hold on;
plot(DX(b(1)),DY(b(1)),'*','LineWidth',3,'Color','red');
%plot(round(median(kosex(1,2))),round(median(kosey(1,2))),'x','LineWid
th',1,'Color','cyan');
title(['Filtered Image (\theta = ',num2str(DS(b(1))),'{\circ})']);
clear all;
end

```

## ÖZGEÇMİŞ



### Kişisel Bilgiler

Adı Soyadı	İsmail PINAR
Uyruğu	T.C.
Doğum tarihi, Yeri	1987, Yozgat/Yerköy
Telefon	0542 557 42 03
E-mail	ismailpinar.66@gmail.com

### Eğitim

Derece	Kurum/Anabilim Dalı/Program	Yılı
Yüksek Lisans	İstanbul Üniversitesi Fen Bilimleri Enstitüsü/Elektrik Elektronik Mühendisliği Anabilim Dalı/Elektrik Elektronik Mühendisliği Programı	2014
Lisans	Uludağ Üniversitesi Elektronik Mühendisliği Bölümü	2010
Lise	Bursa Süleyman Çelebi Lisesi	2005