

TOPOLOJİ TEMELLİ KÖŞE BULMA

Tolga ÜNAL
Yüksek Lisans Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı
Kasım – 2008

JÜRİ VE ENSTİTÜ ONAYI

Tolga ÜNAL'ın **Topoloji Temelli Köşe Bulma** başlıklı **Elektrik Elektronik Mühendisliği** Anabilim Dalındaki, Yüksek Lisans tezi 27/10/2008 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	İmza
Üye (Tez Danışmanı) :	Yard. Doç. Dr. HAKAN GÜRAY ŞENEL
Üye	: Yard. Doç. Dr. ATAKAN DOĞAN
Üye	: Yard. Doç. Dr. ÖZGÜR YILMAZEL

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
tarikh ve **sayılı kararıyla onaylanmıştır.**

Enstitü Müdürü

ÖZET**Yüksek Lisans Tezi****TOPOLOJİ TEMELLİ KÖŞE BULMA****Tolga ÜNAL****Anadolu Üniversitesi****Fen Bilimleri Enstitüsü****Elektrik - Elektronik Mühendisliği Anabilim Dalı****Danışman: Yard. Doç. Dr. Hakan Güray ŞENEL
2008, 50 sayfa**

Köşe bulma, görüntü işleme alanı içinde önemli uygulamalardan biridir. Daha yüksek seviyeli, nesne tanıma, yüz tanıma ve bölütleme gibi işlerde kullanılan temel işlemlerden biridir. Köşe bulma görevi için çok sayıda algoritma önerilmiştir. Bunlar içinde en yaygın kullanılanlar, iyi sonuçlar veren ve büyük işlem kapasitesi gerektirmeyen, SUSAN, CSS, Kitchen-Rosenfeld ve Plessey algoritmalarıdır. Bu tez çalışmasında, bulanık topoloji yaklaşımı kullanan topolojik bir köşe bulma algoritması önerilmiştir. Yöntem, uygulandığı algoritmaya gürültüden en az şekilde etkilenme, dolayısıyla daha az hata yapma ve daha çok doğru köşe bulma gibi özellikler kazandırmaktadır. Bu çalışma, geliştirilen yöntemin SUSAN algoritmasına katılmasına dayanmaktadır. SUSAN algoritmasının seçilmesinin sebebi genel olarak diğer algoritmalara göre daha başarılı olmasıdır. Elde edilen sonuçlara göre, topolojik SUSAN algoritması geleneksel algoritmaya göre biraz yavaş fakat daha başarılıdır. Özellikle, yanlış köşelerin bulunma oranını %40 düşürmektedir.

Anahtar Kelimeler: Görüntü İşleme, Köşe Bulma, Bulanık Topoloji, Sayısal Topoloji.

ABSTRACT
Master of Science Thesis
TOPOLOGY BASED CORNER DETECTION
Tolga ÜNAL

Anadolu University
Graduate School of Sciences
Electrical - Electronics Engineering Program

Supervisor: Assist. Prof. Dr. Hakan Güray ŞENEL
2008, 50 pages

Corner detection is one of the most important tasks in the field of image processing. It is one of basic operations used in higher level tasks such as object recognition, face recognition and segmentation. So far, many algorithms are proposed for corner detection. Among them, the most widely used algorithms are SUSAN, CSS, Kitchen Rosenfeld and Plessey algorithms due to their successes and requiring less computation power. In this thesis work, a fuzzy topological corner detection algorithm is presented. The proposed method enhances the algorithm by decreasing noise effect, thus ensures lower false detections and more true corners. This thesis work is based on extending the SUSAN algorithm with the proposed method. SUSAN algorithm is chosen because it is generally more successful than the others. According to the results obtained, topological version is superior to the conventional version but it is slightly slower. Especially, false corner detection rates are reduced by %40.

Keywords: Image Processing, Corner Detection, Fuzzy Topology, Digital Topology

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	iv
ÇİZELGELER DİZİNİ	vi
SİMGELER VE KISALTMALAR DİZİNİ	vii
1. GİRİŞ	1
1.1 Köşe Bulma Yöntemleri	2
1.2 SUSAN Yöntemi	4
1.3 Kitchen - Rosenfeld Yöntemi	7
1.4 Eğrilik Ölçek Uzayı Yöntemi	10
1.5 Plessey Yöntemi	11
1.6 Değerlendirme	14
2. BULANIK TOPOLOJİ	16
2.1 Bulanık Topoloji.....	16
2.2 BDH'nin Özellikleri	17
3. BDH ÖZELLİKLERİ	21
4. TOPOLOJİK KÖŞE BULMA	29
4.1 Topolojik SUSAN Yöntemi	29
4.2 Köşe Bulma Yöntemlerinin Tepkileri	31
4.3 Topolojik SUSAN Yönteminin Sonuçları.....	43
3. SONUÇ	49
KAYNAKLAR	50

ŞEKİLLER DİZİNİ

1.1 Dairesel maskelerin bir görüntüye uygulanması	5
1.2 Plessey yönteminde, özdeğerlerin konum özellikleri	14
2.1 İki yakın çizginin oluşturduğu görüntü	18
2.2 Gürültü eklenmiş iki çizginin görüntüsü	19
2.3 Adım kenarın kesiti	19
3.1 BDHp ve BDHk görüntüleri - 1	22
3.2 BDHp ve BDHk görüntüleri - 2	22
3.3 BDHp ve BDHk görüntüleri - 3	23
3.4 BDHp ve BDHk görüntüleri - 4	24
3.5 BDHp ve BDHk görüntüleri - 5	24
3.6 BDHp ve BDHk görüntüleri - 6	25
3.7 BDHp ve BDHk görüntüleri - 7	25
3.8 BDHp ve BDHk görüntüleri - 8	26
3.9 BDHp ve BDHk görüntüleri - 9	26
3.10 BDHp ve BDHk görüntüleri - 10	27
3.11 BDHp ve BDHk görüntüleri - 11	28
4.1 Laboratuar test görüntüsü	32
4.2 Laboratuar test görüntüsü sonuçları	33
4.3 Blocks test görüntüsü	33
4.4 Blocks test görüntüsü sonuçları	34
4.5 SUSAN test görüntüsü	34
4.6 SUSAN test görüntüsü sonuçları	35
4.7 Rotating squares test görüntüsü	36
4.8 Rotating squares test görüntüsü sonuçları	36
4.9 Gürültü eklenmiş laboratuar test görüntüsü	37
4.10 Gürültü eklenmiş laboratuar test görüntüsü sonuçları	38
4.11 Gürültü eklenmiş blocks test görüntüsü	38
4.12 Gürültü eklenmiş blocks test görüntüsü sonuçları	39
4.13 Gürültü eklenmiş SUSAN test görüntüsü	40
4.14 Gürültü eklenmiş SUSAN test görüntüsü sonuçları	41

4.15 Gürültü eklenmiş rotating squares test görüntüsü	41
4.16 Gürültü eklenmiş rotating squares test görüntüsü sonuçları.....	42
4.17 Laboratuvar test görüntüsünün karşılaştırmalı sonucu.....	43
4.18 Blocks test görüntüsünün karşılaştırmalı sonucu	44
4.19 SUSAN test görüntüsünün karşılaştırmalı sonucu	44
4.20 Rotating squares test görüntüsünün karşılaştırmalı sonucu.....	45
4.21 Gürültülü laboratuvar test görüntüsünün karşılaştırmalı sonucu.....	45
4.22 Gürültülü blocks test görüntüsünün karşılaştırmalı sonucu	46
4.23 Gürültülü SUSAN test görüntüsünün karşılaştırmalı sonucu.....	46
4.24 Gürültülü rotating squares test görüntüsünün karşılaştırmalı sonucu	47

ÇİZELGELER DİZİNİ

4.1 Köşe bulma sürelerinin karşılaştırması	47
4.2 Köşe bulma başarılarının karşılaştırması.....	47
4.3 Köşe bulma hatalarının karşılaştırması	47
4.4 Köşe bulma sürelerinin karşılaştırması (Gürültülü)	48
4.5 Köşe bulma başarılarının karşılaştırması (Gürültülü)	48
4.6 Köşe bulma hatalarının karşılaştırması (Gürültülü)	48

SİMGELER ve KISALTMALAR DİZİNİ

X	: Noktalar kümesi
ψ	: Morfolojik dönüşüm
B	: Yapısal eleman
W	: Görüntü üzerinde uygulanacak pencere ($3 \times 3, 5 \times 5$ vb.)
$a(x, y)$: Pencere elemanları
$f(x, y)$: Sayısal görüntü
T	: Eşik değeri
G_x, G_y	: Gradyant vektörler
\oplus	: Genleşme (Dilation)
\square	: Aşınma (Erosion)
\circ	: Açma (Opening)
\bullet	: Kapama (Closing)
HAT	: Tepe-üstü (Top-hat)
∇	: Gradyant
mag	: Büyüklük (Magnitude)
α	: Yön

1. GİRİŞ

Görüntülerde köşe bulma, nesnelerin belirgin noktalarının bulunması işidir. Kenarlar ve köşeler nesnelerin belirlenmesi ve tanınması işlerinde ilk adımlardan biridir. Köşe, iki kenarın kesişimi olarak düşünülebildiği gibi kesintisiz bir kenar üzerindeki değişinti (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]:

1. Kenar bulma algoritmalarıyla, kenarların bulunmasından sonra, kenarların üzerinde dolaşarak köşelerin tespit edilmesi.
2. Gri yoğunluk esasına dayalı arama
3. Çeşitli açı ve büyüklükteki köşelerin şablon arama yöntemiyle bulunması

Ancak, hangi düşünce kullanılırsa kullanılsın, bazı durumlarda köşelerin belirlenmesi zor olabilir. Köşelerin bulunmasında karşılaşılan sorunlar şöyle özetlenebilir:

- Köşeler, piksel değerlerindeki yerel maksimum veya minimum noktalarla karıştırılabilir
- Çizgilerin son noktaları köşe olarak işaretlenebilir.
- Eğimli bir düzlemdeki bir eğri, bakış açısı nedeniyle yanlışlıkla köşe olarak değerlendirilebilir.

Görüntü üzerinde ne tür nesnelerin bulunduğu dair öngörü bulunmadığı durumlarda, köşe bulma algoritmaları beklenen sonuçları vermekten uzaktır. Bazı köşelerin bulunamaması veya köşe olmayan ilgi noktalarının köşe olarak tespit edilmesi, daha üst seviyelerde bulunan 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 [1].

- Köşeler iyi algılanmalıdır. Az sayıda köşe bulunması ve olması gerekenden daha fazla köşenin bulunması gibi durumlar en aza indirilmelidir.
- Köşenin yeri doğru şekilde bulunmalıdır.
- Bir kenara sadece bir yanıt verilmelidir.
- Köşe bulma algoritmaları daha üst seviyeli görüntü işleme görevlerinde kullanıldıkları için ve bunlara zaman bırakmak için hızlı çalışmalıdırlar.

Algoritmanın hızının artırılması, sonuçları ters bir şekilde etkileyebilir. Bu nedenle, ilk üç maddede belirtilen kriterlerin iyi olduğu durumlarda, algoritmanın hızının düşünülmemesi yerinde olacaktır. Zira görüntü işleme için kullanılan donanım hızları her geçen gün artmaktadır ve algoritmanın doğru çalışması hızından daha önemlidir.

1.1 Köşe Bulma Yöntemleri

Köşelerin yüksek seviyeli görüntü işleme ve nesne tanıma işlerindeki önemleri nedeniyle, köşe bulma konusunda çok sayıda araştırma gerçekleştirilmiştir. İlk araştırmalardan birini yapan Moravec [2] 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. Harris [3] Plessey algoritması olarak bilinen, Moravec algoritmasının daha gelişmiş bir türü olan bir yöntem önermiştir. Beaudet [4] köşelerin yakınlarında daha yüksek değer verebilen determinanta dayalı DET işlecini geliştirmiştir. Kitchen ve Rosenfeld [5] doğrudan gri seviyeli görüntülerde kullanılan altı yaklaşım önermiştir. Bu çalışmada önerilen yöntemler değişinti yönünün değişinti genliğine, kenar üzerinde yönün değişimi, birbirine en fazla benzeyen komşu piksellerin yönünün değişmesine ve yüzey oturarak dönüş miktarını hesaplama gibi yaklaşımlara dayanmaktadır. Tsai [6] yapay sinir ağlarına dayanan bir yöntem geliştirmiştir. Ji ve Haralick [7] kovaryans dağılımına dayanan bir teknik sunmuştur.

Kitchen ve Rosenfeld'in [5] önerdiği 6 köşe bulma algoritması içinden, gradyan açısının gradyan şiddetine bakılmasına dayanan yöntemeye dayalı bir tekniği, Fang ve Huang önermiştir [8]. Zhang ve Rockett, önerdikleri gri seviyeli köşe modeline dayanarak, köşeleri etiketlemeye yarayan Bayes karar dayalı bir teknik geliştirmişlerdir [9]. Wu ve Rosenfeld görüntülerin x ve y koordinatlarındaki izdüşümlerindeki süreksizliğin incelenerek köşelerin bulunabileceği yaklaşımında bulunmuşlardır [10]. Paler ve diğerleri [11] lokal piksel değerlerinin dağılımını kullanan bir yöntem geliştirmişlerdir. Rangarajan ve diğerleri [12] köşelerde tepkisi artan optimal bir fonksiyon önermiştir. Arrebola ve diğerleri [13] nesnelerin kenarlarından kurulan zincir kodların yerel histogramlarının köşe bulma için kullanılabileceğini önermişlerdir. Sohn ve diğerleri [14] ortalama alan soğuması (mean field annealing) yöntemini köşe bulma için önermişlerdir.

Kohlman [15] 2 boyutlu Hilbert dönüşümünü köşe bulmaya uygulamıştır Mehrotra ve diğerleri kenar ve köşe bulma için Gauss fonksiyonunun birinci ve ikinci yönlü türevlerine dayanan iki yöntem sunmuştur [16]. Davies, Hough dönüşümünü köşe bulmaya uygulamıştır [17]. Smith ve Brady, SUSAN ismini verdikleri yöntemle dairesel bir maske kullanarak ve türev hesaplamadan köşelerin bulunabileceği bir yöntem önermişlerdir [18]. Mokhtarian ve Suomela [1], yay evrimi düşüncesini kullanan ve eğrilerin değişik ölçeklerde değerlendirilmesine dayanan başarılı bir yöntem sunmuştur.

Yapılan çeşitli çalışmalar, önerilen farklı yaklaşımlar içinde Plessey köşe detektörü [3], Kitchen ve Rosenfeld'in [5] geliştirdiği detektör, SUSAN detektörünün [18] ve CSS (Eğri Ölçek Uzayı - Curve Scale Space) [1] temelli yöntemlerin diğerlerine göre çok daha başarılı olduğu söylenebilir. Bu nedenle bu tez çalışmasında bu yöntemler temel alınarak çözümler önerilmiştir.

1.2 SUSAN Yöntemi

Smith ve Brady [18], dairesel bir maske içinde parlaklık karşılaştırmasına dayanan, SUSAN adında kenar ve köşe bulma algoritmaları sunmuştur. Algoritma, değişinti (gradyan) hesaplaması gerektirmediği ve sadece ufak bir

komşuluktaki piksel değerlerine baktığından diğer kenar ve köşe bulma algoritmalarına göre daha hızlı olduğu bilinmektedir. Ancak, SUSAN yöntemi patentli olduğu için üzerinde değişiklik yapılamamakta ve patentli olması kullanımını ve uyarlanmasını engellemektedir.

SUSAN (Smalest Univalue Segment Assimilating Nucleus) kenar ve köşe bulma algoritmaları izotropik bir tepki verebilmek için dairesel bir maske kullanır. Smith ve Brady'nin makalelerinde kullandıkları dairenin yarı çapı 3.4 pikseldir. Bu yarıçaptaki bir dairesel maske içinde 37 piksel bulunmaktadır. Kullanılabilecek en küçük maske büyüklüğü 3×3 büyüklükte olabilmektedir.

Maske içindeki pikseller eğer bir nesneye bağlıysalar, piksel değerlerinin benzer olduğunu varsayar. Algoritma, dairesel maskenin merkezinde yer alan piksel ile ona benzer piksel değerlerine sahip piksellerin sayısını hesaplar ve ağırlık merkezini bulur. Benzer parlaklıktaki piksellere USAN (Univalue Segment Assimilating Nucleus) adı verilir. Benzer parlaklık değerlerini bulabilmek için kullanılan denklem şöyledir:

$$C(\vec{r}, \vec{r}_0) = \begin{cases} 1, & |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0, & |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases} \quad (1.1)$$

Formülde \vec{r}_0 maskenin merkez pikseli, \vec{r} ise maske içinde kalan diğer pikselleri ve $I(\vec{r})$ piksellerin değerlerini gösterir. Özellikle köşe bulma konusunda daha yararlı olan diğer bir benzerlik fonksiyonu da önerilmiştir:

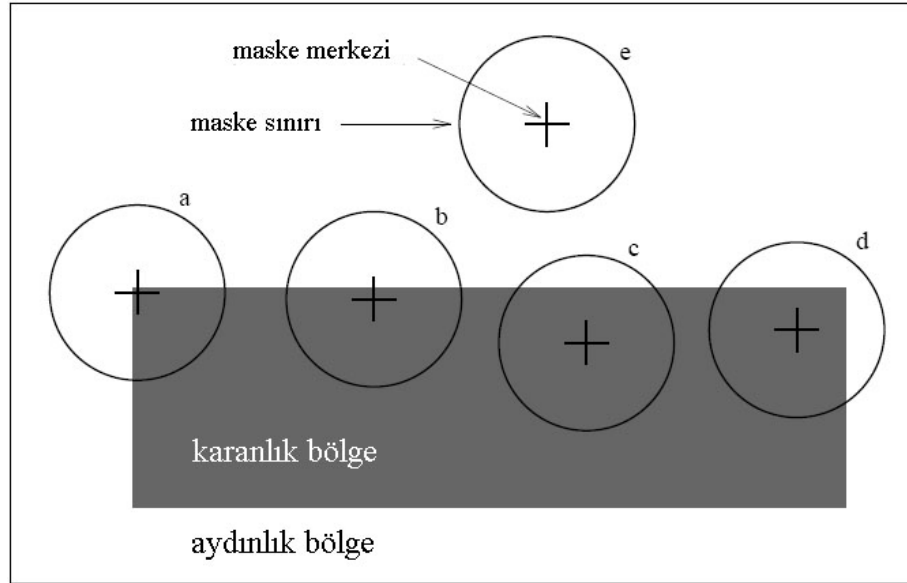
$$C(\vec{r}, \vec{r}_0) = \exp\left(-\frac{I(\vec{r}) - I(\vec{r}_0)}{t}\right)^6 \quad (1.2)$$

USAN'ın hesaplanabilmesi için maske içindeki piksellerin gri seviye yoğunluk farkı üzerine belirli bir t değerine göre eşikleme yapılır. Karşılaştırma sonucu da $C(\vec{r}, \vec{r}_0)$ çıktısı olur. C değerine göre USAN haritası oluşur. Smith ve Brady makalelerinde, t değeri olarak 27 değerini kullanmışlardır. Bu değerini deneme yanılma yöntemiyle bulunduğu ve sadece 8 bitlik, diğer bir deyişle 256 gri seviyeli görüntülerde geçerli olduğu anlaşılmaktadır.

$C(\vec{r}, \vec{r}_0)$ değeri maske içindeki bütün piksellere uygulandıktan sonra, orijine yakın olan ve orijin noktasına bağlı pikseller bulunmaktadır. USAN içinde yer alan ve merkeze bağlı olduğu $C(\vec{r}, \vec{r}_0)$ formülü yardımıyla belirlenmiş piksellerin sayısı aşağıdaki formül ile bulunabilir.

$$n(\vec{r}_0) = \sum_{\vec{r}} C(\vec{r}, \vec{r}_0) \quad (1.3)$$

$n(r_0)$, USAN haritasının kapladığı alanla birlikte USAN içinde yer alan piksellerin toplam sayısını göstermektedir. t eşik değerinin kullanılması sayesinde, $n(r_0)$ değerinin küçük olması ve bu şekilde USAN içinde zıtlık yaratılması sağlanmıştır. Şekil 1.1'de USAN yönteminin görüntüye uygulanması gösterilmiştir.



Şekil 1.1. Dairesel maskeler kullanarak USAN yönteminin bir görüntüye uygulanması

Bir sonraki aşamada, USAN'ın kapladığı alan bir eşik değeriyle, g , karşılaştırılmaktadır. Smith ve Brady kenar bulma algoritması için g 'nin değerini $3n_{max}/4$ olarak vermişlerdir. Burada n_{max} n 'nin alabileceği en büyük değerdir. Köşe bulma algoritmasında bu değer $n_{max}/2$ 'dan daha küçük olması gereklidir.

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{eger } n(\vec{r}_0) < g \\ 0 & \text{degilse} \end{cases} \quad (1.4)$$

$R(\vec{r}_0)$ SUSAN'ın kenar ve köşe tepkileridir. Buradaki formülden de anlaşılacağı gibi USAN'ın kapladığı alan küçüldükçe tepkisi de büyümektedir. Geometrik eşik değeri g 'nin kullanımı doğru kenarların ve köşelerin bulunmasını engellememektedir. Örneğin, adım kenar (step edge) için, n değeri her zaman için $n_{max}/2$ değerinden, en azından kenarın bir tarafı için küçük olacaktır. Geçerli kenarlar bu nedenle reddedilmeyecektir. Köşe bulunan durumlarda, adım kenarın verdiği tepkiden daha küçük bir tepki yaratacağından, g 'nin $n_{max}/2$ 'dan daha küçük kullanılması köşeyle karıştırılmasını engellemektedir. Eğer kenar adım

kenardan farklı ve daha yumuşak bir profile sahip olması durumunda, n 'nin değeri $n_{\max}/2$ değerinden de küçük olması beklenebilir. Bu nedenle, yumuşak profilli kenarların reddedilmesi riski adım kenarlara göre daha da düşüktür.

SUSAN kenar bulma algoritmasında kenarın yönünün bulunması da önemli bir aşamadır. Maksimum olmayan noktaların bastırılması gerekliyse, kenar yönünün doğru şekilde bulunması önemlidir. Ayrıca, kenarın piksel boyutunun da altında ölçülerde konumlandırılması (sub pixel localization) gerekebilir. Hatta bazı durumlarda kenarın yeriyile birlikte yönü ve şiddeti de önemli bilgi verebilir. SUSAN yöntemiyle değışinti değerlerine gerek duyulmaksızın, USAN'ın ağırlık merkezi ve maskenin çekirdeği arasındaki vektörün maske içindeki yerel kenarın yönüne dik olması sayesinde kenar yönü belirlenebilmektedir. USAN'ın ağırlık merkezi şöyle hesaplanabilir:

$$\bar{\vec{r}}(\vec{r}_0) = \frac{\sum_{\vec{r}} \vec{r} c(\vec{r}, \vec{r}_0)}{\sum_{\vec{r}} c(\vec{r}, \vec{r}_0)} \quad (1.5)$$

Burada, r USAN içindeki diğer bir pikseli işaret etmektedir. USAN kenar yönü boyunca ince bir çizgi oluşturmaktadır. Kenar yönü en uzun simetri eksenini bulunarak hesaplanabilir. Simetri eksenini aşağıdaki formüller kullanılarak hesaplanabilir:

$$d_x = \overline{(x - x_0)^2}(\vec{r}_0) = \sum_{\vec{r}} (x - x_0)^2 c(\vec{r}, \vec{r}_0) \quad (1.6)$$

$$d_y = \overline{(y - y_0)^2}(\vec{r}_0) = \sum_{\vec{r}} (y - y_0)^2 c(\vec{r}, \vec{r}_0) \quad (1.7)$$

$$d_{xy} = \overline{(x - x_0)(y - y_0)}(\vec{r}_0) = \sum_{\vec{r}} (x - x_0)(y - y_0) c(\vec{r}, \vec{r}_0) \quad (1.8)$$

Yukarıdaki formüller içinde d_x / d_y oranı kenarın yönünü göstermektedir. d_{xy} ise diyagonal kenarların pozitif veya negatif değışintilere sahip olmadığını belirlemek için kullanılmaktadır.

Köşe tespitinde bu işlemlerden sonra USAN haritası içinde yerel minimum (non-maximum suppression) bulur ve köşe noktaları tespit edilir. SUSAN yönteminde bazı köşe bulma yöntemlerinde olduğu gibi türev içeren işlemler kullanılmadığı için gürültüden etkilenmez. Gürültünün ortadan kaldırılmasına gerek yoktur ve hızlı bir yöntemdir. Bu özellikleri ile diğer köşe bulma yöntemlerine üstünlük sağlamaktadır.

1.3 Kitchen – Rosenfeld Yöntemi

Kitchen ve Rosenfeld [5] gri seviye görüntülerde köşesellik (cornerity) adı verilen değerleri ölçmek için altı yöntem sunmaktadır. Bu yöntemlerin zincir kod kullanan geleneksel yaklaşıma göre avantajı, bölütleme sürecine gerek duyulmadan doğrudan gri seviyeli görüntüler üzerinde uygulanabilmeleridir. Geleneksel yaklaşımda, görüntü bölütlendikten sonra elde edilen nesnelerin kenarlarında uygulanan zincir kodlamayla, köşesellik ve eğrilik değerleri bulunarak köşeler belirlenebilmektedir. Kitchen ve Rosenfeld'in önerdiği yöntemler görüntüleri bölütlemeye gerek kalmadan doğrudan gri seviye görüntülere uygulanabilen tekniklerdir.

Birinci yöntem değişinti yönünün değişintisinin genliğine dayanmaktadır. Bu yöntemde, ilk olarak gradyan yönleri ve gradyan büyüklükleri hesaplanır. Gradyan yönleri -180 ile +180 derece arasında alınıp, bunların mutlak değerleriyle gri-seviye bir resim olarak oluşturulursa, bu yön resmindeki parlaklık değişimleri orijinal resimdeki kenarların yön değişimlerini göstermektedir. Parlaklık değişimleri yön resminin gradyan büyüklüğü hesaplanarak bulunabilmektedir.

İkinci yöntemde kenar boyunca yön değişiminin izlenebileceği belirtilmiştir. Sadece kenar boyunca yön değişimini ölçmek, nesne sınırlarındaki ani dönüşleri saptayabilir. Orijinal görüntüde gradyan yönü hesaplandıktan sonra, sonuç görüntüye 3x3'lük bir operatör uygulanarak bu yöntem gerçekleştirilebilmektedir. Bu operatör her komşuluk içinde merkezden geçen ve merkez pikseldeki gradyan yönüne dik olan çizgiye yakın yerleşen karşılıklı pikselleri inceler. Operatörün sonucunda incelenen karşılıklı iki pikselin gradyan yönleri arasında fark bulunur. Eğer kayda değer bir fark bulunursa bu eğrinin yönünden ve büyüklüğünden bilgi

elde etmek mümkündür. Yine bu eğrinin değeri de gradyan büyüklüğü ile çarpılmalıdır. Kitchen ve Rosenfeld, bir önceki yönteme göre bu yöntemin daha iyi sonuç verdiğini belirlemiştir.

Üçüncü yöntemde birbirine çok benzeyen komşuların arasındaki açıdan yararlanılmıştır. Eğer bir kenar yerel bir komşuluğun merkezinden geçiyorsa, kenar boyunca yer alan pikseller merkez piksel ile benzer gri tonadırlar. Kenarın dışında kalan pikseller ya daha açık, ya da daha koyu olabilirler. 3×3 'lük bir komşulukta merkez piksele en yakın gri tondaki iki pikseli ele alalım. Merkez piksel C ve diğer pikseller A ve B olsun. AC ve AB vektörleri arasındaki yön farkı hesaplanır ve bu yön farkı eğrilik ölçüsü olarak kullanılır.

Dördüncü yöntem yerel komşuluğun merkezinde hesaplanan gradyan yönünün değişim oranının başka bir şekilde türetilmesine dayanmaktadır. Kare bir komşuluk merkezinden bir kenar kontur çizgisinin geçtiği düşünülerek aşağıdaki formül uygulanabilir.

$$I(x, y) = I(x, y) - I(0,0) = 0 \quad (1.9)$$

Denklemden y , x 'in bir fonksiyonu olarak düzenlenir ve x 'e göre y 'nin birinci ve ikinci türevleri hesaplanırsa, düzlemdeki eğrilik aşağıdaki formül ile elde edilir ve gradyan büyüklüğü ile çarpılarak tekrar k köşelilik değeri elde edilmiş olur.

$$\frac{d^2 y / d^2 x}{(1 + (dy / dx)^2)^{3/2}} \quad (1.10)$$

Beşinci yöntemde Beaudet'in [4] DET operatörü olarak isimlendirilen operatör, bir yüzeydeki köşeleri ve sırt noktalarını (saddle point) belirler. Bu operatörün anlamlı bir sonuç elde edebilmesi için gradyan büyüklüğü ile çarpılmasına gerek yoktur. Gerçekte tam olarak bir kenar üzerine konumlandığında, doğru sonuçlar vermeyebilir, fakat köşe yakınlarında konumlandığında kenarın her iki tarafı üzerinde doğru tepkiler verir. DET'in dezavantajı keskin kenarlı şekiller üzerinde kötü sonuçlar vermesidir.

Yukarıda açıklanan bütün köşe bulma dedektörleri büyük ölçekli komşuluklarda sapma gösterebilir. Köşelerin konumlarının daha iyi belirlenmesi, dedektörlerden alınan sonuçlar üzerinde piksellerin değişik boyuttaki dedektörler

ile çarpımı veya sonuçlara maksimum olmayanların bastırılması (non-maxima suppression) yöntemi uygulanarak yapılabilir. Gradyan büyüklüğü ile çarpım yapan yöntemlerde, şöyle bir problem ortaya çıkmaktadır: eğer bir kenar yakınlarında bir köşe varsa köşe dedektörü bu kenardan etkilenir ve köşeyle birlikte kenara da tepki verir. Bu problem de kenar gradyan büyüklüklerini çarpımda kullanmadan önce maksimum olmayanların bastırılması yöntemi uygulanarak çözülebilmektedir

Sunulan yöntemleri geleneksel bir köşe bulma dedektörü ile karşılaştırmak mümkündür. Geleneksel köşe bulma yöntemlerinde, öncelikle görüntü bir eşik değerine göre, içindeki nesnelere ve arka plan olmak üzere iki kısma ayrılır. Nesnelere ayırt edilebilmesi için her nesnenin 4'lü-komşuluk sınırları belirlenir. Sınır noktaları üzerinde P gibi bir pikselden m adım önce ve m adım sonra olmak üzere iki vektör alınır ve bu iki vektör arasındaki yön farkı eğimin büyüklüğünü verir. Geleneksel yöntemde kullanılan eşikleme değeri önemlidir, eğer uygun bir eşik değeri kullanılmazsa, hesaplanan eğimler güvenilir olmayabilir.

Kitchen ve Rosenfeld tarafından önerilen teknikler içerisinde en başarılısı dördüncü yöntem olarak görünmektedir. Başarısı üç nedene dayandırılabilir. İlk olarak, yüzey uygunlaştırılırken gürültünün onun üzerindeki etkileri yok edilmektedir. İkinci olarak, uygunlaştırılmış yüzey, komşuluk bölgesinin gri seviye örüntüsünden ilgililenen özellikleri yakalamak için yüksek yeterliliktedir. Üçüncü olarak, halihazırda bölütlenmiş şekildeki köşe kavramına uyan bir işleç kullanılmıştır.

1.4 Eğrilik Ölçek Uzayı Yöntemi

Mokhtarian ve Suomela [1], görüntüler için Curvature Scale Space (CSS) gösterimini temel alan bir köşe bulma yöntemi önermiştir. İlk olarak Canny kenar bulma yöntemi kullanılarak orijinal görüntüden kenarlar çıkartılır. Görüntüdeki kenarların mutlak eğriliklerinin maksimum değere sahip olduğu noktalar köşe olarak tanımlanır. Köşeler CSS'in yüksek ölçeği ile bulunur ve konumunu düzeltmek için birçok daha düşük ölçekler boyunca izlenir. Bu yöntem gürültülü

görüntülerde başarılı sonuçlar vermektedir ve var olan diğer köşe bulma yöntemlerinden de daha iyi çalıştığı gösterilmiştir.

Çalışmada ortaya çıkan diğer bir durum da, Canny kenar bulma yönteminin 45 ve 135 derecelik kenarlara verdiği sonuçların, diğer açılardaki kenarlara göre daha kalın olmasıdır. CSS yöntemi geleneksel köşelere ek olarak, kenarlar boyunca elde edilen eğrilik sıfır-kesişimlerinde fazladan nokta sağlayabilmektedir.

Eğri ölçek uzayı tekniği, düzlemsel bir eğrinin değişmeyen geometrik özelliklerini ortaya çıkarmak için kullanılabilir. Bu değeri hesaplayabilmek için, eğri, Γ 'nin yayın uzunluk parametresi olan u kullanılır:

$$\Gamma(u) = (x(u), y(u)) \quad (1.11)$$

Eğrinin geliştirilmiş hali şöyledir:

$$\Gamma_{\sigma}(u) = (X(u, \sigma), Y(u, \sigma)) \quad (1.12)$$

Burada, $x(u)$ standart sapması σ olan Gaussian fonksiyonuyla evriştirilmektedir:

$$\begin{aligned} X(u, \sigma) &= x(u) \otimes g(u, \sigma) \\ Y(u, \sigma) &= y(u) \otimes g(u, \sigma) \end{aligned} \quad (1.14)$$

Eğer, σ , 0 ile sonsuz arasında aldığı bir değere göre eğrinin farklı versiyonları ortaya çıkmaktadır. Bu şekilde eğrinin farklı yumuşaklıktaki versiyonları oluşturulabilir.

Eğrinin sıfır geçişlerini veya eğrinin yumuşatılmış versiyonlarındaki ekstrem noktaların bulabilmek için, eğrilik değerinin, Γ_{σ} versiyonlarının hesaplanması gereklidir.

Burada,

$$\kappa(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{1.5}} \quad (1.15)$$

$$X_u(u, \sigma) = \frac{\partial}{\partial u} (x(u) \otimes g(u, \sigma)) = x(u) \otimes g_u(u, \sigma) \quad (1.16)$$

$$X_{uu}(u, \sigma) = \frac{\partial^2}{\partial u^2} (x(u) \otimes g(u, \sigma)) = x(u) \otimes g_{uu}(u, \sigma)$$

$$\begin{aligned} Y_u(u, \sigma) &= y(u) \otimes g_u(u, \sigma) \\ Y_{uu}(u, \sigma) &= y(u) \otimes g_{uu}(u, \sigma) \end{aligned} \quad (1.17)$$

Bu fonksiyonla CSS görüntüsü olan Γ , $K(u,\sigma)=0$ olduğunda elde edilebilmektedir.

CSS operatörü uygulamada kenar özelliklerini kullandığı için, öncelikle kenar konturlarının ortaya çıkarılması gerekmektedir. Bu amaçla CSS operatörü kenarları belirlemek için Canny kenar detektörünü kullanır. Kenar belirlemede CSS operatöründe farklı kenar bulma detektörleri de kullanılabilir. Canny kenar detektörünün diğerlerine göre daha iyi sonuçlar verdiği belirlenmiştir. Ardından, kenar çizgileri içinde yer alan boşluklar doldurulur ve kesintisiz kenar çizgileri elde edilir. Doldurulan noktalar sayesinde T şeklindeki noktalar belirlenir. En büyük ölçekte elde edilen eğri üzerindeki mutlak maksimumun değeri bir eşik değerini geçtiğinde köşe olarak işaretlenir. Ölçek küçültülerek, köşenin daha iyi konumlandırılması sağlanır. T köşelerle bulunan köşeler karşılaştırılarak yakın köşeler saptanır.

1.5 Plessey Yöntemi

Köşe bulma gibi özellik bulma algoritmalarının başarısında, görüntü üzerine uygulanan kenar süzgecinin başarısı büyük önem taşır. Harris ve Stephens tarafından geliştirilen, Plessey yöntemi [3], yerel otokorelasyon işlemine dayanan bir köşe ve kenar bulucudur. Otokorelasyon fonksiyonunun istatistiksel bir işlem olması, yöntemin gürültüden etkilenen yerel piksel değerleri yerine incelenen komşuluğun istatistiksel özellikleri üzerine kurulu olmasını sağlamaktadır. Bu durum da, Plessey fonksiyonunun gürültüye karşı daha dayanıklı olmasını sağlamıştır.

Plessey yöntemi, Moravec'in [2] önerdiği köşe bulma algoritmasına benzer. Otokorelasyona işlemine benzer bir şekilde, görüntü üzerine yerel görüntü parçalarını dikkate alır ve bu görüntü parçalarının çeşitli yönlerde küçük miktarlarda kaydırılmasından dolayı görüntüde oluşan ortalama yoğunluk değişimlerini hesaplar. Moravec yönteminde dikkat edilmesi gereken üç durum vardır:

- Eğer görüntü parçası düz bir alan üzerinde ise, görüntü parçasının bütün yönlerde kaydırılması sadece küçük değişimlere neden olacaktır.

- Eğer görüntü parçası bir kenar üzerinde ise, görüntü parçasının o kenar boyunca kaydırılması küçük değişimler oluşturacaktır.
- Eğer görüntü parçası bir köşe veya bağımsız bir noktaya denk geldi ise, görüntü parçasının bütün yönlerde kaydırılması büyük değişimlere neden olacaktır. Böylece buradaki kaydırmalar sonucu elde edilecek en küçük değişim, bu nokta etrafında elde edilen en büyük değişim ise böylece burada bir köşe bulunmuştur.

Görüntü yoğunluğunu I ile, bir (x, y) ' de yapılan kaydırma işlemi sonucunda oluşan değişimi E ile ifade edersek;

$$E_{x,y} = \sum_{u,v} w_{u,v} |I_{x+u,y+v} - I_{u,v}|^2 \quad (1.18)$$

Burada W görüntü parçasının (pencerenin) konumunu ifade eder. Kaydırma işlemleri birim eksenler boyunca gerçekleştirilir. İşlemler $(1,0), (1,1), (0,1), (-1,1)$ kullanılarak yapılır. Moravec' in köşe bulma yöntemi köşeyi en sade biçimde, bazı eşik değerlerinin üzerinde olan $\min(E)$ değerleri içerisinde yerel maksimuma sahip olan nokta olarak tanımlar.

Moravec' in köşe bulma yönteminde karşılaşılan durumlar şunlardır:

- Sadece 45' er derecelik kaydırmalar dikkate alındığından sonuç yöne bağımlıdır (anisotropic).
- Gürültü içeren görüntüye düzeltme uygulanmadan sonuç elde edilir. Hiç düzeltme uygulanmaz.
- Her nokta için E değerlerinin minimumu dikkate alındığından, kenarlar da köşe olarak tespit edilebilir.

Plessey yöntemindeyse, E değişimi, küçük (x,y) kaydırmalar için kısaca şöyle yazılabilir.

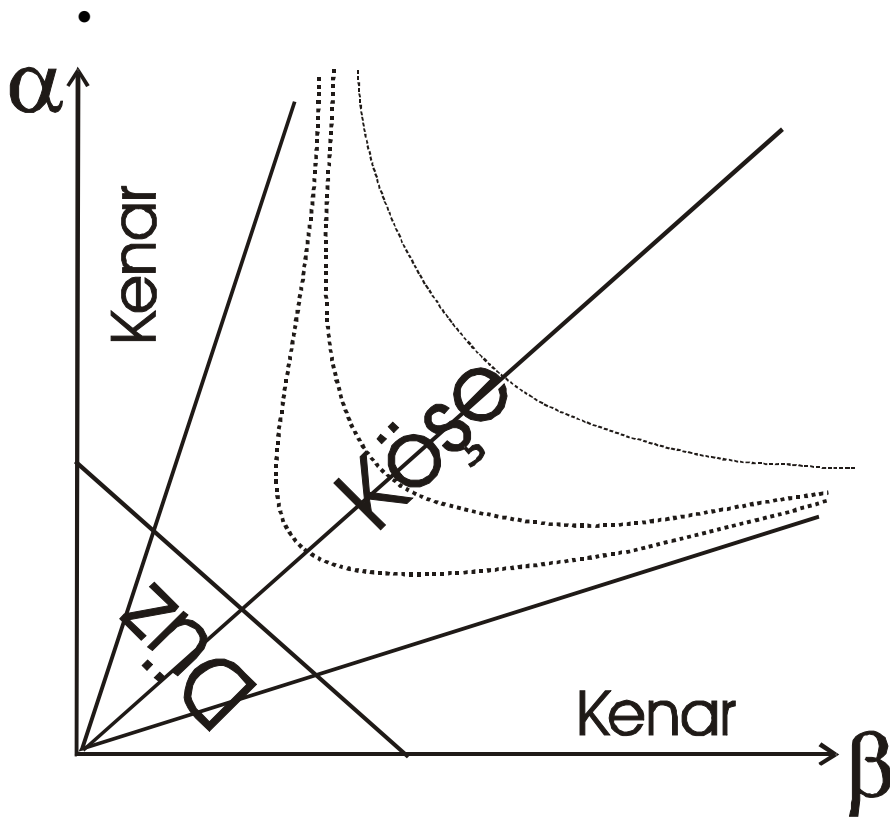
$$E(x, y) = (x, y)M(x, y)^T \quad (1.19)$$

Burada 2x2 lik simetrik matris M şöyle yazılabilir.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (1.20)$$

E işlevi bir otokorelasyon işlevi olarak düşünülebilir. Burada M merkezde yer alan otokorelasyon değeri olarak değerlendirilebilir. Böylece bu E matrisinin özdeğerleri olarak düşünülecek α , β bize ek bilgiler sunmaktadır (Şekil 1.2).

- Eğer yerel otokorelasyon işlevi düzse her iki özdeğer de küçükse, görüntü parçasının kapsadığı alan yaklaşık olarak sabit yoğunluktadır.
- Eğer yerel otokorelasyon işlevi sırt (ridge) görünümündeyse, diğer bir deyişle bir özdeğer yüksek ve diğeri düşükse, E 'de sadece sırt boyunca (yani kenar boyunca) yapılan kaydırmalar küçük değişimlere sebep olacaktır: Bu da bir kenarı ifade etmektedir.
- Eğer otokorelasyon işlevi keskin bir yükselme yaptığı için her iki özdeğer de yüksekse, E 'nin değeri herhangi bir yönde kaydırma işlemi yapıldığında artacaktır. Bu durum bir köşeyi ifade etmektedir.



Şekil 1.2. Plessey yönteminde, özdeğerlerin konumuna göre elde edilen özellikler

1.6 Değerlendirme

Bu çalışmada, bulanık topoloji kullanan bir köşe bulma algoritması geliştirilmiştir. Gri seviyeli görüntülerin, bulanık set olarak düşünülmesi Rosenfeld [19] tarafından önerilmiştir. Bu görüntülerde iki pikselin ne ölçüde bağlantılı olduklarının belirlenmesi de bulanık topolojinin alanına girmektedir. Yerel bir komşulukta piksellerin birbirleriyle olan ilişkilerinin, hem normal görüntüde hem de görüntünün tersinin değerlendirilmesiyle bölütleme konusunda çeşitli yöntemler önerilmiştir [20][21]. Daha düşük seviyeli bir işlem olan medyan süzme işleminde yerel bağlantılılıkların medyan filtrenin tepkisini olumlu yönde etkilediği Şenel [22] tarafından gösterilmiştir. Ayrıca, küçük bir gözlem penceresi içinde hesaplanabilen iki çeşit bağlantı haritaları arasındaki asimetrik durumu kullanan bir kenar bulma algoritması da geliştirilmiştir [23].

Görüntünün düzü ve tersi üzerinde yerel bağlantılılıkların kullanılmasıyla, bazı köşe bulma algoritmalarının geliştirilebileceği görülmüş ve bu çalışmada bunu gerçekleştiren bir yöntem geliştirilmiştir.

2. BULANIK TOPOLOJİ

Birbirine bağlı bileşenler bir görüntü içinde yer alan en temel bilgi parçalarıdır. Bağlantılılık, sayısal topolojinin temel kavramıdır. Sayısal topolojinin sadece iki piksel değerinden oluşan ikili görüntülerde uygulaması açıktır. Sayısal topoloji kullanılarak, ikili görüntülerde alanlar, nesnelerin sınırları ve kenarlar tanımlanabilir. Gri seviyeli görüntülerdeyse, sayısal topolojinin karşılığı bulanık topolojidir. Rosenfeld sayısal topolojiyi, çok seviyeli görüntüler için genişleterek bulanık topolojiyi tanımlamıştır [19]. Rosenfeld, gri seviyeli görüntülerdeki piksel değerleri $[0,0\dots1,0]$ arasına ölçeklendiğinde, görüntünün bulanık küme olarak düşünülebileceğini önermiştir. Böylece, ölçeklenmiş piksel değeri, o pikselin parlak nesnelere içindeki üyelik fonksiyonu olarak tanımlanmaktadır. Benzer şekilde, görüntünün negatifi alındığında elde edilen görüntüdeki ölçeklenmiş piksel değerleri, piksellerin koyu nesnelere kümesi içindeki üyeliklerinin ölçüsü olarak düşünülebilir.

2.1 Bulanık Topoloji

Σ sembolü tamsayı koordinatların oluşturduğu dikdörtgen şeklinde iki boyutlu bir diziyi tanımlasın. Koordinatları (x, y) olan herhangi bir piksel olan p , Σ 'nin elemanıdır. Σ 'nin bulanık alt kümesi olan W , $p \in \Sigma$ olmak şartıyla, sıralı ikililerden, $W = (p, \mu_w(p))$ oluşmaktadır. Buradaki $\mu_w(p)$, pikselin parlak noktalar içindeki üyelik derecesini, diğer bir deyişle ölçeklenmiş piksel değerini vermektedir.

W 'nin Σ 'nin bulanık alt kümesi olduğunu ve $\sigma: p = p_0, p_1, p_2, \dots, p_n = q$ dizisinin W içindeki herhangi iki noktayı birbirine kesiksiz olarak bağlayan bir yol olduğunu düşünelim. Yolun, diğer bir deyişle σ 'nin kuvveti, $S_w(\sigma)$, yol üzerindeki en zayıf bağlantıdır.

$$S_w(\sigma) \equiv \min_{0 \leq i \leq n} \mu_w(p_i) \quad \forall p_i \in \sigma \quad (2.1)$$

İkili (binary) görüntülerde, iki noktanın birbirine bağlı olduğu, iki noktayı bağlayan ve sadece parlak piksellerden oluşan bir yolun varlığıyla anlaşılabilir.

Diğer yandan, gri seviyeli görüntülerde iki pikselin bağlantılı olduğunu göstermek için farklı bir yaklaşım gerekir. Bu nedenle önerilen bulanık topoloji, gri seviyeli görüntülerdeki piksel değerlerinin ikiden fazla olması nedeniyle oluşan belirsizliği, bağlantılılık derecesi kavramıyla ölçülebilir hale getirmektedir. Görüntüdeki her iki piksel çiftine, (p, q) , sıfır ve bir aralığında bir sayı verilmektedir. Bu değer p ve q 'nin hangi dereceyle birbirine bağlı olduğunu göstermektedir. Pratikteki uygulamalarda ölçeklenmiş değerler yerine gerçek piksel değerler kullanılabilir. Gri seviyeli görüntülerde p ve q arasındaki bağlantı derecesi şöyle tanımlanmaktadır [22].

$$C_w(p, q) \equiv \max_{\sigma} S_w(\sigma) \quad \forall \sigma : p, \dots, q \quad (2.2)$$

İki pikselin, p ve q 'nin W içinde bağlı olabilmesi için, tanım itibariyle, $C_w(p, q) = \min(\mu_w(p), \mu_w(q))$ koşulunun sağlanması gereklidir. Bağlantı derecesi iki piksel değerinin en küçüğünden daha büyükse, bu iki nokta birbirine bağlıdır. Gri seviyeli görüntülerdeki bağlantılılık uygulamaları ve daha geniş tanımlar [24]'de verilmektedir.

Şenel çalışmasında, pikseller arasındaki bağlantı derecelerinin kullanılarak, küçük gözlem pencereleri için bağlantı derecesi haritasının (BDH) nasıl oluşturulacağını tanımlamıştır [22]. Bağlantı derecesi haritası, $o \in W$ pikselinin diğerleriyle nasıl bağlandığını şöyle göstermektedir

$$BDH_w(o) = C_w(o, p) \quad \forall p \in W. \quad (2.3)$$

Bu makalede önerilen yöntem BDH kavramını temel almaktadır.

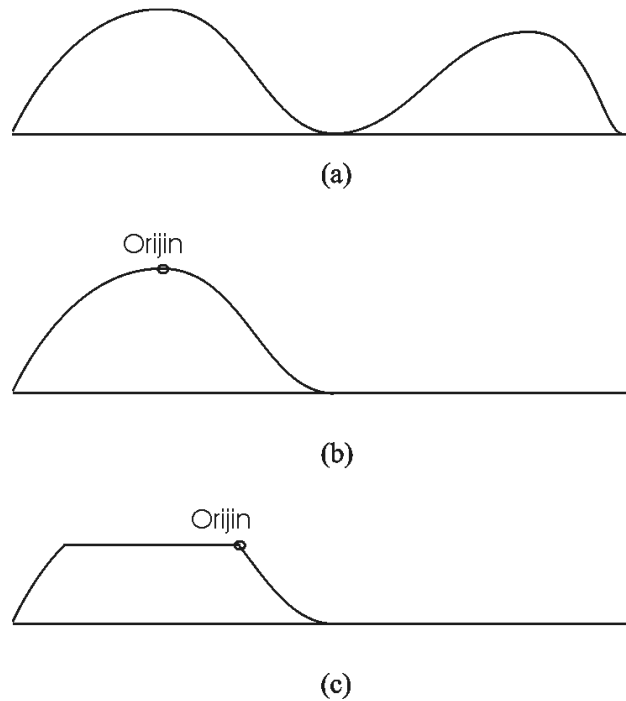
2.2 BDH'nin Özellikleri

Bir gözlem penceresi içinde yer alan piksel değerleri kullanılarak yeni bir pencere oluşturulmaktadır. Bunun içindeki değerler, merkez piksele olan bağlantı derecelerini göstermektedir. Görüntünün içindeki nesnelerin nasıl algılandığına bakılarak, iki farklı BDH oluşturulabilir. Zira görüntü içinde birbirine bağlı parlak pikseller ve bunların oluşturduğu nesnelere olabileceği gibi, koyu piksellerin birleşerek oluşturdukları koyu nesnelere de bulunabilir. Eğer görüntü olduğu gibi

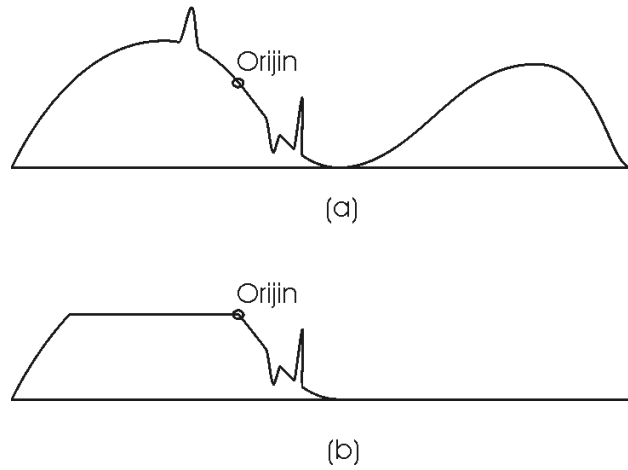
kullanılırsa, parlak pikseller için BDHp hesaplanabilir. En büyük piksel değerine göre görüntünün tersi üzerinde, koyu noktalar için BDHk elde edilir [22].

Bağlantı derecesi haritalarının nasıl çalıştığını gösterebilmek için, üzerinde birbirine yakın iki çizginin bulunduğu bir görüntü kullanılacaktır (Şekil 2.2.a). Bu görüntünün belirli bir yerinden kesit alınarak, bir boyuta taşınması tercih edilmiştir. İki boyutlu uzayda bağlantıların çok farklı yollar üzerinden gerçekleşebilmesine rağmen, BDH'nin nasıl çalıştığının gösterilebilmesi için kesit görüntüleri yeterlidir.

Şekil 2.1.b'de gösterilen orijin referans alınırsa ortaya çıkan BDH'de, sağdaki çizgi görülmemektedir. Görüntüde merkez piksele bağlı olmayan nesnelere ortadan kaldırılmıştır. Eğer, merkez soldaki nesnenin tepesi yerine, sağdaki nesneye yakın olan yamaçta yer alırsa, sağdaki nesne ortadan kalktığı gibi soldaki nesnenin bir bölümü de kesilmektedir (Şekil. 2.1.c). Diğer bir deyişle, orijine bağlantılı olmayan nesnelere bir miktar bozulmayla ortadan kaldırılmaktadır.



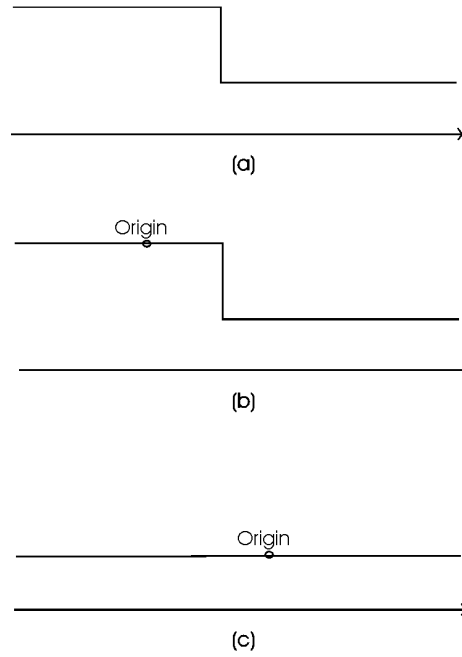
Şekil 2.1 a) İki yakın çizgi nesnesinin oluşturduğu görüntünün kesiti, b) Orijin nesnenin tepe noktasındayken BDH, c) orijinin yamaçta yer aldığı durumda BDH



Şekil 2.2. a) Gürültü eklenmiş görüntünün kesiti, b) Bağlantı derecesi haritası.

Şekil 2.1.a'da gösterilen kesitteki herhangi üç piksele gürültü eklenirse (Şekil 2.2.a) ve BDH oluşturulursa, Şekil 2.2.b elde edilir. Orijin pikselin değerinden yüksek olan ve gürültü içeren pikseller, soldaki nesnenin bir kısmıyla birlikte temizlenmektedir. Orijinin sağında yer alan gürültü içeren pikseller deformasyon yaratarak, sağdaki nesneyle birlikte yok edilmektedir. Kısacası, BDH'de gürültü bileşenleri eğer orijine bağlı değilse, bağlı olmayan diğer nesnelere birlikte temizlenmektedir.

İdeal adım kenar, Şekil 2.3.a'da gösterilmektedir. Şekil 2.3.b, orijin soldaki yüksek bölüme konulursa elde edilen BDH'yi göstermektedir. Şekilde görüleceği gibi, adım kenar bozulma olmadan BDH'ye taşınmaktadır. Diğer yandan, orijin sağdaki düşük piksel değerli bölüme konulursa, BDH'de düz bir grafik elde edilmektedir (Şekil 2.3.c). Bu durum, gözlem penceresindeki en düşük değerli pikselin, kendi piksel değerine eş değer derecede diğerlerine bağlanmasından ileri gelmektedir [22]. Çünkü en düşük değerli piksel her zaman diğerlerine bağlıdır.



Şekil 2.3. a) Adım kenarın kesiti, b) Orijin soldaki yüksek bölümde yer alırsa elde edilen BDH, c) Orijin sağdaki

Orijin kenarın parlak tarafına konumlanırsa, BDH’de parlak ve koyu pikseller görülmektedir. Eğer orijin koyu noktalardan birinin üzerine konulursa, sadece koyu pikseller görülmektedir. Gözlenen bu asimetric durumdan, bu makalede anlatılan yöntemin geliştirilmesinde yararlanılmaktadır. Yukarıda da belirtildiği gibi, bağlantı derecesi haritası görüntünün kendisi ve negatifinden olmak üzere iki şekilde elde edilmektedir. Her iki harita birbirini tamamlayan bilgilere sahiptir.

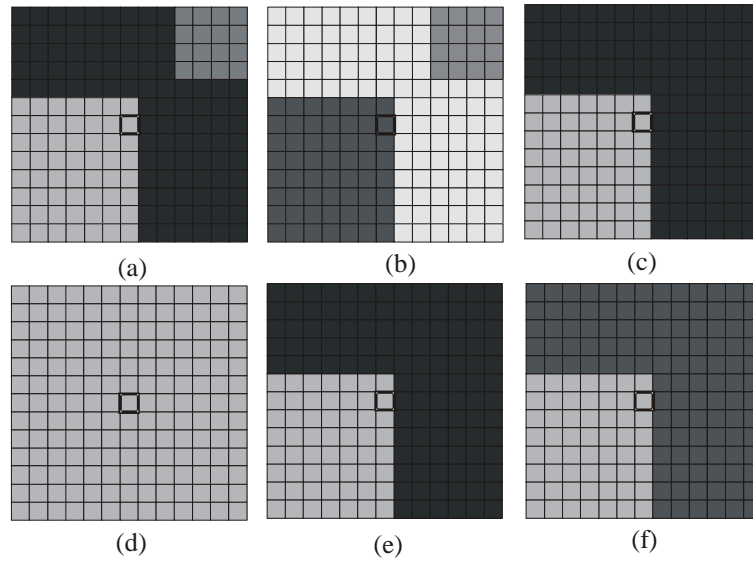
BDH oluşturmak için boyutları 3×3 ’ten daha büyük olan görüntü parçaları kullanılmalıdır. Çünkü 3×3 görüntü parçasında, 8’lik bağlantı kullanılırsa bütün pikseller birbirine bağlı durumdadır. Benzer mantıkla 5×5 parçalar da yeterli olmayabilir. Genellikle, BDH’nin uygulanabilmesi için 5×5 ’lik gözlem pencerelerinden büyük parçalar seçilmelidir.

3. BDH ÖZELLİKLERİ

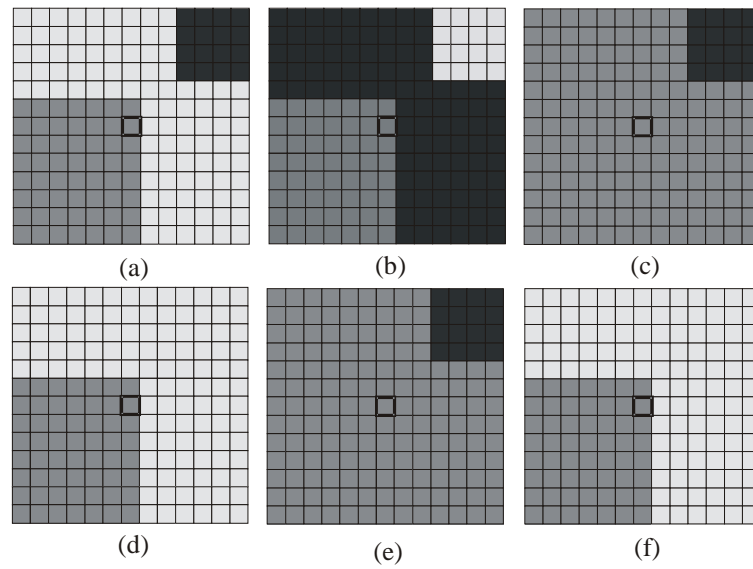
Bağlantı derecesi haritalarının (BDH) çeşitli özellikleri, kenar ve köşe gibi ilgi noktalarının bulunmasını kolaylaştırmaktadır. Bölüm 2.1’de BDH’nin gürültüyü ortadan kaldıran özellikleri, parlak ve koyu nesnelere olan asimetrik tepkileri gösterilmişti. BDHp ve BDHk görüntülerinin beraber kullanımı yerine ayrı kullanıldıklarında, koyu ve parlak nesnelere eşit olarak tepki vermemektedir. Ancak, BDHp ve BDHk görüntülerinin birbirlerini tamamlayıcı özellikleri bulunmaktadır.

Şekil 3.1’de 13x13’lük bir görüntü parçası (Şekil 3.1.a) üzerinde uygulanmış BDHp (Şekil 3.1.c) ve BDHk (Şekil 3.1.d) görüntüleri görülmektedir. Şekilde yer alan görüntüde, sol alt köşede orijine bağlı şekilde bir nesne görülmekte ve sağ üst köşede orijine bağlı olmayan bir nesne daha bulunmaktadır. Kenar ve köşe bulma işlerinde orijine bağlı olan nesnenin görülmesi, diğer küçük nesnenin de süzülmesi gereklidir. Şekil 3.1.e’de $\min(\text{BDHp}, 255-\text{BDHk})$ hesaplamasının sonucu gösterilmektedir. Şekil 3.1.f’deyse $\max(\text{BDHp}, 255-\text{BDHk})$ sonucu bulunmaktadır. Şekil 3.1.e ve 3.1.f’de orijinde bulunan nesnenin korunduğu görülmektedir.

Benzer şekilde, Şekil 3.2’de 13x13’lük bir görüntü parçası (Şekil 3.2.a) üzerinde uygulanmış BDHp (Şekil 3.2.c) ve BDHk (Şekil 3.2.d) görüntüleri görülmektedir. Şekilde yer alan görüntüde, sol alt köşede orijine bağlı şekilde koyu bir nesne görülmekte ve sağ üst köşede orijine bağlı olmayan daha koyu bir nesne daha bulunmaktadır. Bu iki nesnenin arasında parlak bir bölüm yer almaktadır. Kenar ve köşe bulma işlerinde orijine bağlı olan koyu nesnenin görülmesi, diğer küçük olan koyu nesnenin de süzülmesi gereklidir. Şekil 3.2.e’de $\min(\text{BDHp}, 255-\text{BDHk})$ hesaplamasının sonucu gösterilmektedir. Şekil 3.2.f’deyse $\max(\text{BDHp}, 255-\text{BDHk})$ sonucu bulunmaktadır. Şekil 3.2.f’de orijinde bulunan nesne korunurken, Şekil 3.2.e’de koyu renkteki küçük nesne görülmektedir.

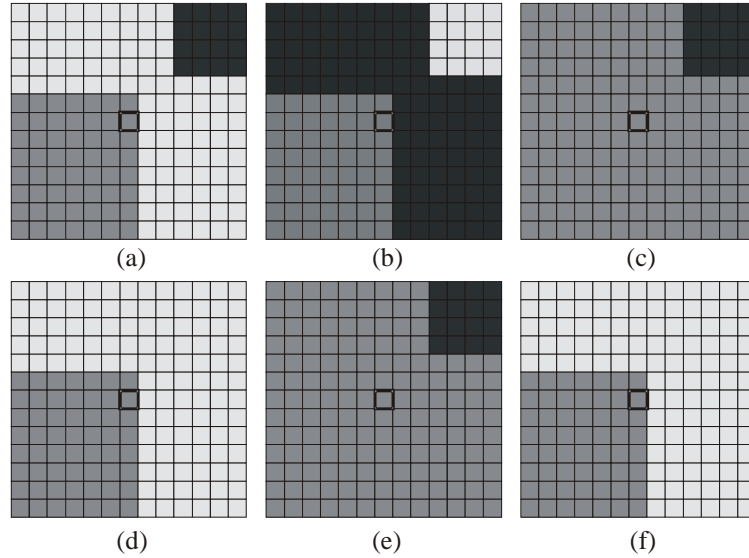


Şekil 3.1. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$



Şekil 3.2. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp, d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

Şekil 3.3’de iki koyu nesneyi ayıran parlak bir alan görülmektedir. Eğer orijin koyu nesnelere birinde yer alırsa, 3.3.e’de sadece en koyu nesne tutulmakta ve orijinin üzerinde olduğu koyu nesne ortadan kalkmaktadır. Diğer yandan, Şekil 3.3.f’de orijin üzerinde bulunduğu koyu nesne korunmaktadır.

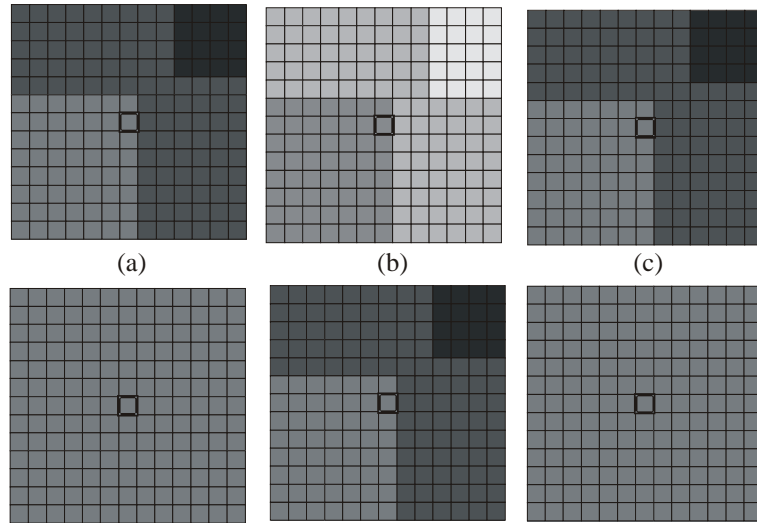


Şekil 3.3. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk’nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

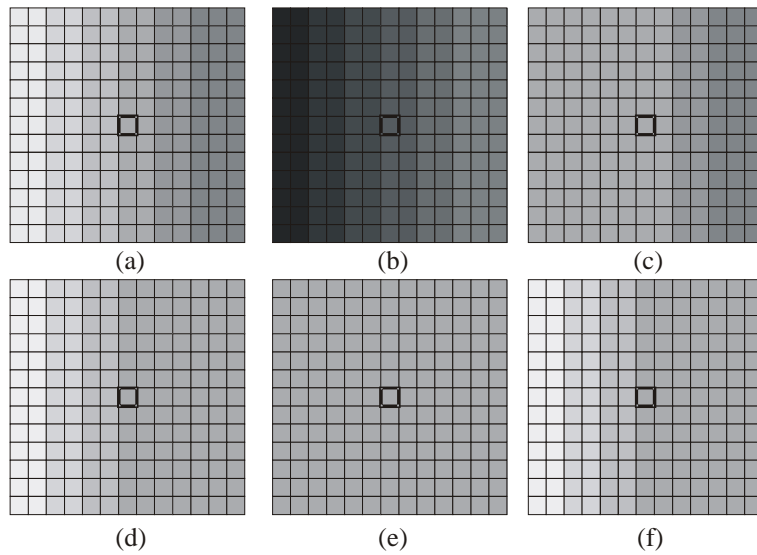
Şekil 3.4.a’da daha açık ve koyu renkli iki nesneyi ayıran daha koyu bir alan bulunmaktadır. Orijin daha açık renkli nesnede bulunduğu anda, Şekilde 3.4.e orijinal görüntünün bir kopyasıdır. Diğer yanda, Şekil 3.4.f orijinin bulunduğu nesnenin tonunu içeren sabit alan görülmektedir.

Şekil 3.5.a’da, görüntünün sol tarafındaki daha parlak alandan sağdaki koyu alana kadar inen bir merdiven kenar görülmektedir. Şekilde 3.5.e’de sadece orijinin bulunduğu pikselin değerinde sabit bir alan görülmektedir. Şekilde 3.5.f, orijinal görüntünün orijin piksel seviyesinde kesilmiş hali olarak görülmektedir.

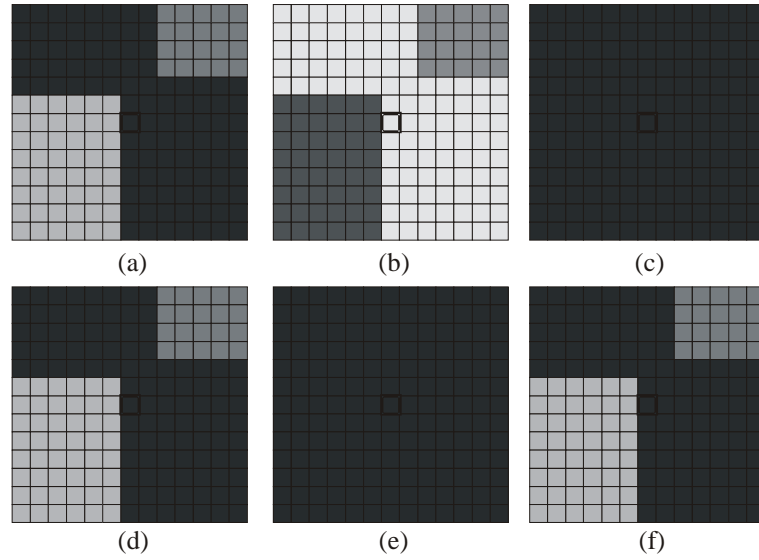
Şekil 3.6.a’da parlak iki noktayı ayıran daha koyu renkli bir alan görülmektedir. Orijin koyu alandadır. Şekil 3.6.e, orijin pikselin değerinde düz bir alan göstermektedir. Diğer yandan, Şekil 3.6.f, orijinal görüntünün aynısıdır.



Şekil 3.4. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

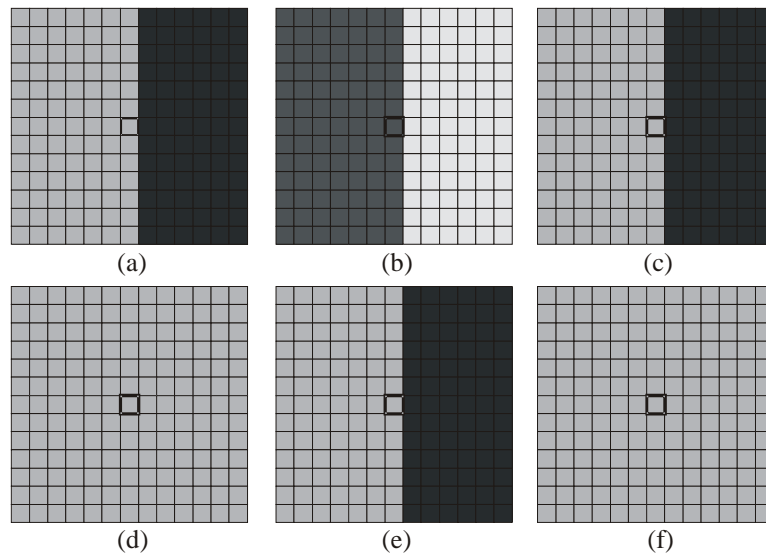


Şekil 3.5. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$



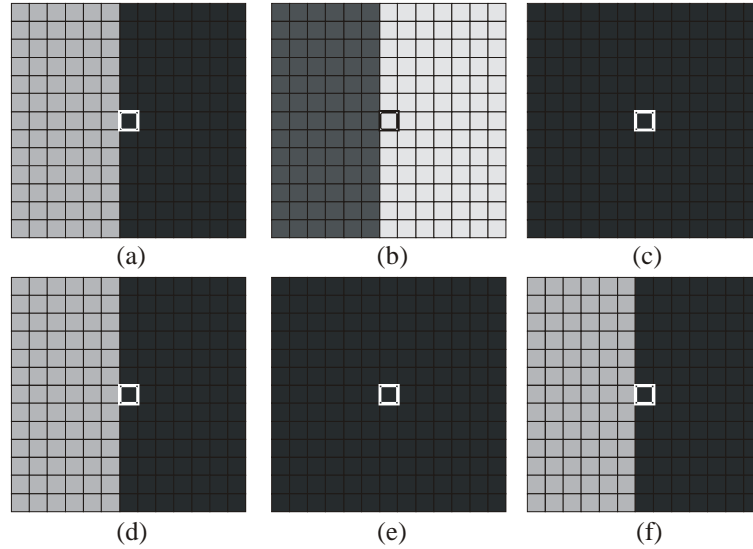
Şekil 3.6. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

Şekilde 3.7.a'da ideal bir kenar görülmektedir. Orijin kenarın parlak ola tarafında görülürse, BDHp'nin (Şekil 3.7.c) kenarı koruduğu ama BDHk'da kenarın görülmediği anlaşılmaktadır. Şekil 3.7.e'de kenarın görüldüğü ama Şekil 3.7.f'de kenar yerine sabit bir alan bulunduğu görülmektedir.



Şekil 3.7. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

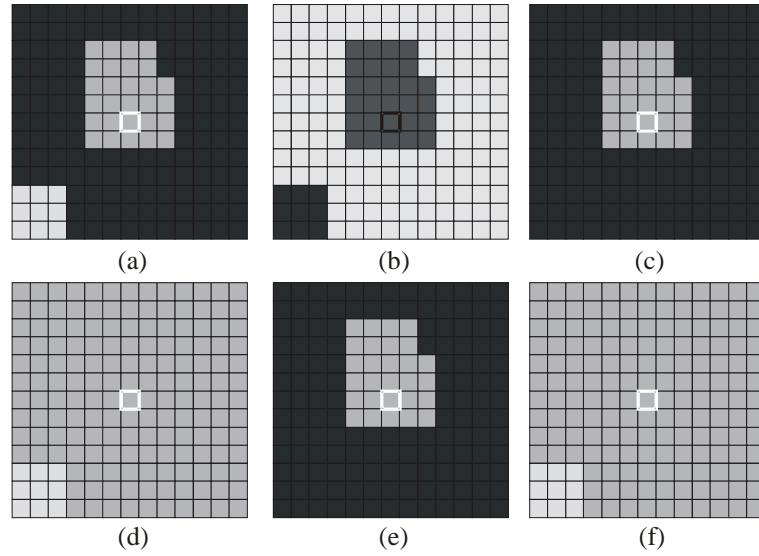
Şekilde 3.8'de orijini koyu bölümde yer alan bir kenar verilmiştir. BDHp orijinin piksel değerinde düz koyu bir alanı gösterirken, 1-BDHk'de kenar eksiksiz komşuluk içinde bulunmaktadır. Şekil 3.8.d'de yer alan şekil BDHp'nin aynısıyken, Şekil 3.8.e'de kenar eksiksiz görülmektedir.



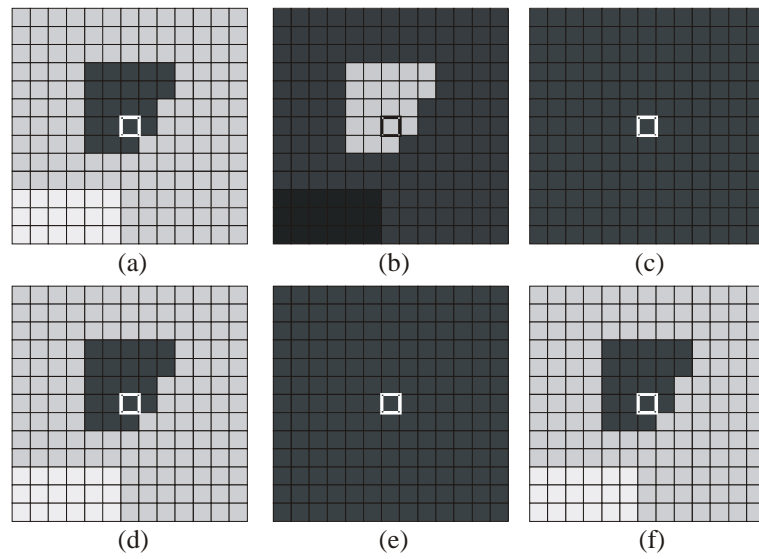
Şekil 3.8. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

Şekil 3.9.a'da orijinin bağlı bulunduğu küçük parlak bir nesne ve sol alt köşede parlak küçük bir nesne bulunmaktadır. BDHp'de sadece küçük nesne gösterilmiş ve 1-BDHk'de genel olarak nesne ortadan kalkmıştır. Şekil 3.9.d'de nesnenin görüldüğü ve Şekil 3.9.e'de düz bir sathın bulunduğu görülmektedir.

Şekil 3.10.a'da koyu bir nesnenin düz bir sathın üzerinde bulunduğu bir komşuluk görülmektedir. Şekil 3.10.e'de gösterilen $\min(\text{BDHp}, 1-\text{BDHk})$ işlemi, düz bir sathı üretmekte ama Şekil 3.10.e'de gösterilen $\max(\text{BDHp}, 1-\text{BDHk})$ işlemi de Şekilde 3.10.a komşuluğunu vermektedir.

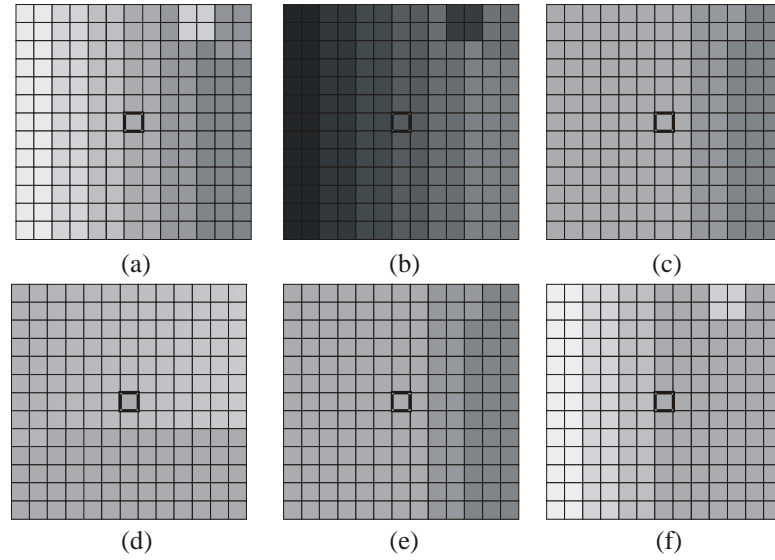


Şekil 3.9. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$



Şekil 3.10. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

Şekilde 3.11.a'da yokuş kenar ve komşuluğun sağ üst tarafında parlak bir nesne gösterilmektedir. BDHp orijin seviyesinde kesilmiş ve sağ taraftaki koyu bölümün korunduğu bir resim üretmektedir. 1-BDHk'de ise, sol taraftaki parlak bölümün korunduğu ama sağ taraftaki koyu kısmın orijinin piksel seviyesinde kesilmiş bir resim görülmektedir. Şekil 3.11.e ve Şekil 3.11.f'ye bakıldığında her iki komşuluğun orijinal resmin birbirini tamamlar parçaları olduğunu görebiliriz.



Şekil 3.11. a) 13x13 görüntü, b) Görüntünün tersi, c) BDHp d) BDHk'nin tersi (1-BDHk), e) $\min(\text{BDHp}, 1-\text{BDHk})$, f) $\max(\text{BDHp}, 1-\text{BDHk})$

On bir resimden oluşan inceleme sonrasında, $\min(\text{BDHp}, 1-\text{BDHk})$ görüntüsünün, eğer orijin resimdeki baskın parlak nesnenin üzerindeyse arzu edilen bir tepki ürettiği görülmektedir. Diğer taraftan $\max(\text{BDHp}, 1-\text{BDHk})$ görüntüsü, orijin baskın nesnenin koyu tarafından yer aldığı anda anlamlı bir resim ürettiği anlaşılmıştır. Bu çalışmada, her iki formül kullanılmıştır.

4. TOPOLOJİK KÖŞE BULMA

Bağlantı derecesi haritası (BDH), kenar bulma, gürültü temizleme gibi düşük seviyeli görüntü işleme görevlerinde başarıyla kullanılmıştır [22][23]. BDH'nin gürültüyü temizleyen özellikleri, BDHp ve BDHk'nin birbirini tamamlayan ama asimetrik çıktıları bu yöntemlerdeki temel yaklaşım tarzıdır. BDH kavramının köşe bulma konusunda kullanılması konusunda şu ana kadar literatürde görülen herhangi bir çalışma yapılmamıştır. Bölüm 3'de yer alan çeşitli durumlar için üretilen BDH haritalarının, köşe bulma için kullanılabilir yararlı özellikleri olduğu görülmektedir.

Bölüm 1'de sunulan literatür çalışması sonucunda ortaya çıkan durumda, en başarılı köşe bulma algoritmalarının SUSAN, CSS, Plessey ve Kitchen Rosenfeld teknikleri olduğu görülmektedir. Özellikle SUSAN yöntemi [18], yüksek hız özellikleri, gürültüyü engelleyen yaklaşımı ve gradyan değerlerinin hesaplanmasına gerek duymaması nedeniyle diğerlerine göre artıları olan bir algoritmadır. SUSAN yönteminin en büyük sıkıntısı, bulunduğu yanlış köşelerin sayısının diğerlerine göre daha fazla olmasıdır. SUSAN yönteminin daha başarılı hale getirilmesi için bu tez çalışmasında topolojik bir yaklaşım önerilmektedir. Yeni algoritmanın SUSAN'ın bazı sorunlarının ortadan kaldıracağı düşünülmüştür.

4.1 Topolojik SUSAN Yöntemi

SUSAN (Smallest Univalued Segment Assimilating Nucleus) kenar ve köşe bulma algoritmaları izotropik bir tepki verebilmek için dairesel bir maske kullanır. Smith ve Brady'nin makalelerinde kullandıkları dairenin yarı çapı 3.4 pikseldir. Bu çaptaki bir dairesel maske içinde 37 piksel bulunmaktadır. Ancak topolojik olarak bu kadar küçük yarıçap içindeki komşulukta bütün pikseller birbirine bağlıdır. Şenel [22] topolojik özellikleri kullanabilmek için en küçük 7x7'lik bir gözlem penceresinin kullanılması gerektiğini belirtmiştir. Ayrıca, 3.4 piksel yarıçaplı dairesel maskelerin kullanılması, daha yüksek çözünürlükte yakalanmış görüntüler için maskenin küçük kalması nedeniyle sorun yaratabilmektedir. Bu

nedenlerden ötürü, BDH'lerin hesaplanabilmesi için 9x9'luk gözlem penceresi kullanılmasına karar verilmiştir.

Bilindiği gibi benzer parlaklıktaki piksellere USAN (Univalve Segment Assimilating Nucleus) adı verilir. Parlak nesne köşeleri için Topolojik USAN, 9x9'luk gözlem penceresi içinde bulunan $\min(BDHp, 1-BDHk)$ görüntüsü üzerinde hesaplanmaktadır (bkz. Şekil 3.1). Koyu nesne köşeleri için Topolojik USAN, 9x9'luk gözlem penceresi içinde bulunan $\max(BDHp, 1-BDHk)$ görüntüsü üzerinde hesaplanmaktadır (Şekil 3.2).

Bu şekilde, $\min(BDHp, 1-BDHk)$ ve $\max(BDHp, 1-BDHk)$ işlemleriyle elde edilen görüntüler üzerine uygulandığında, dış bükey ve iç bükey kenarlar için C_p ve C_k , iki ayrı USAN değeri hesaplanacaktır.

$$\begin{aligned} C_p(\vec{r}, \vec{r}_0) &= \exp\left(-\frac{I_p(\vec{r}) - I_p(\vec{r}_0)}{t}\right)^6 \\ C_k(\vec{r}, \vec{r}_0) &= \exp\left(-\frac{I_k(\vec{r}) - I_k(\vec{r}_0)}{t}\right)^6 \end{aligned} \quad (4.1)$$

Burada eşik değeri olan t 'nin değeri, SUSAN algoritmasındaki gibi 27 olarak uygulanmıştır. I_p ve I_k değerleri parlak ve koyu nesnelere oluşan $\min(BDHp, 1-BDHk)$ ve $\max(BDHp, 1-BDHk)$ işlemleriyle elde edilen görüntülerdeki piksel değerlerini ifade etmektedir.

$C(\vec{r}, \vec{r}_0)$ değeri, maske içindeki bütün piksellere uygulandıktan sonra, yakın olan ve orijin noktasına bağlı pikseller bulunmaktadır. USAN içinde yer alan ve merkeze bağlı olduğu $C(\vec{r}, \vec{r}_0)$ formülüyle belirlenmiş piksellerin sayısı şöyle bulunabilir:

$$n(\vec{r}_0) = \sum_{\vec{r}} C(\vec{r}, \vec{r}_0) \quad (4.2)$$

$n(\vec{r}_0)$, USAN haritasının kapladığı alanla birlikte USAN içinde yer alan piksellerin toplam sayısını göstermektedir.

Bir sonraki aşamada, USAN'ın kapladığı alan bir eşik değeriyle, g , karşılaştırılmaktadır. Smith ve Brady köşe bulma algoritması için g 'nin değerini $n_{max}/2$ olarak vermişlerdir. Burada n_{max} n 'nin alabileceği en büyük değerdir. Parlak ve koyu nesnelere için iki farklı hesaplama yapılmaktadır.

$$\begin{aligned}
R_p(\vec{r}_0) &= \begin{cases} g - n_p(\vec{r}_0) & n_p(\vec{r}_0) < g \\ 0 & n_p(\vec{r}_0) \geq g \end{cases} \\
R_k(\vec{r}_0) &= \begin{cases} g - n_k(\vec{r}_0) & n_k(\vec{r}_0) < g \\ 0 & n_k(\vec{r}_0) \geq g \end{cases}
\end{aligned} \tag{4.3}$$

Parlak ve koyu nesnelere için USAN'ın ağırlık merkezleri şöyle hesaplanabilir:

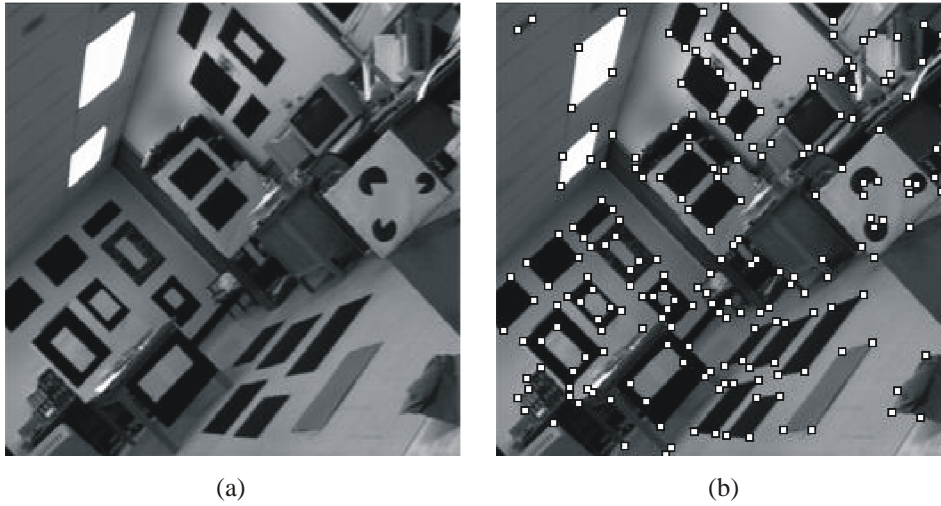
$$\begin{aligned}
\bar{r}_p(\vec{r}_0) &= \frac{\sum_{\vec{r}} \vec{r} c(\vec{r}, \vec{r}_0)}{\sum_{\vec{r}} c(\vec{r}, \vec{r}_0)} \\
\bar{r}_k(\vec{r}_0) &= \frac{\sum_{\vec{r}} \vec{r} c(\vec{r}, \vec{r}_0)}{\sum_{\vec{r}} c(\vec{r}, \vec{r}_0)}
\end{aligned} \tag{4.4}$$

Parlak ve koyu nesnelere için hazırlanan iki ayrı hesaplamada birbirini tamamlayan sonuçların elde edilmesi beklenebilir. Her iki yaklaşımın tepki verdiği noktalarda köşelerin kesinlikle bulunduğu görülebilir. Ancak, parlak ve koyu nesnelere için yapılan hesaplamalardan birinin tepki verdiği noktalar olabileceği düşünüldüğünde, önerilen yöntemin yanlış köşelerin sayısını artıracak açıktır. Diğer yandan, her ikisinin de tepki vermediği anlarda kesinlikle köşenin orada bulunmadığı sonucuna varılabilir.

4.2 Köşe Bulma Yöntemlerinin Tepkileri

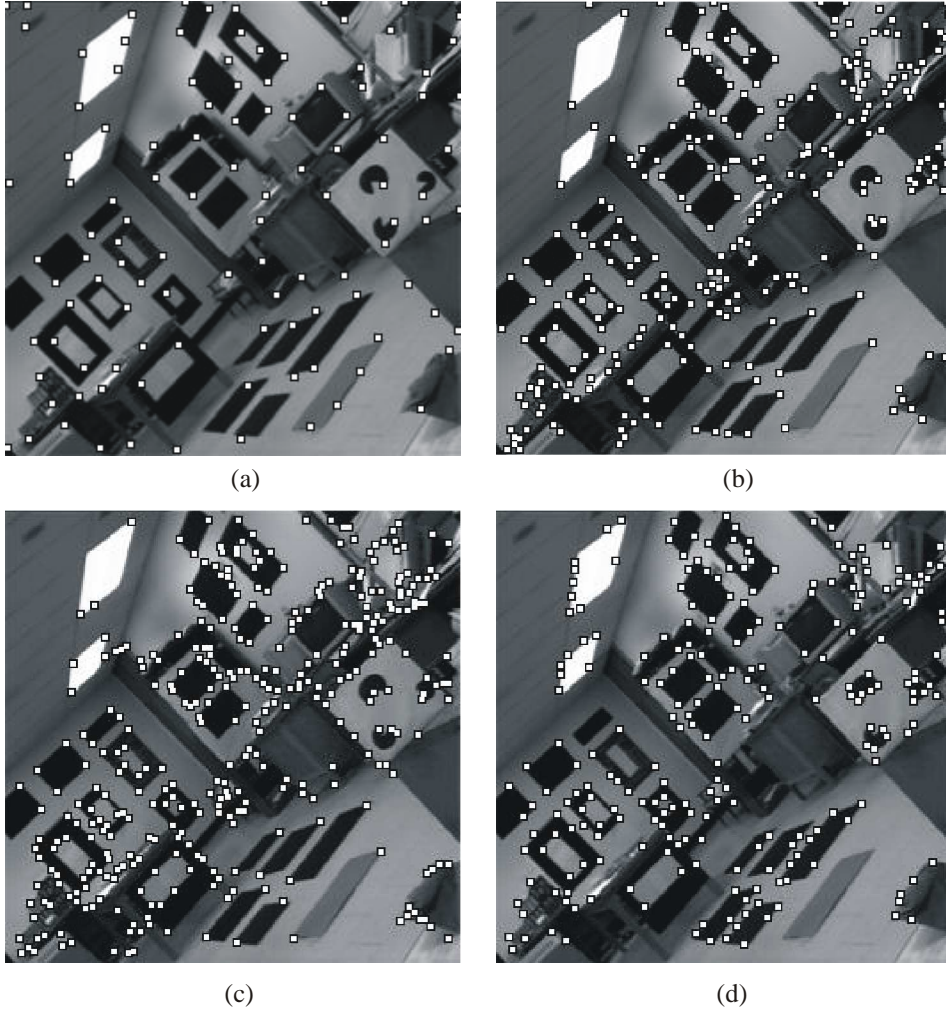
Bu bölümde, SUSAN, CSS, Plessey ve Kitchen-Rosenfeld yöntemlerinin tepkileri incelenecektir. Test görüntüleri olarak, çeşitli makalelerde yer alan standart test görüntüleri kullanılmıştır. Test görüntülerinin, hem normal halleri hem de 0.01 yoğunluklu salt & paper gürültüsü eklenerek elde edilmiş gürültülü halleri kullanılmıştır. Böylelikle algoritmaların gürültülü ve gürültüsüz ortamdaki başarıları test edilmiştir. Her test görüntüsü, belirli bir yöntemde en iyi sonuçları verirken, diğer yöntemlerde başarısız sonuçlar üretmektedir. Bu nedenle çok sayıda test görüntüsüyle çalışmaların yapılması gerekmektedir.

Şekil 4.1’de laboratuvar adı test görüntüsünün ve 213 köşenin yerleri gösterilmektedir. Şekil 4.2.a’da Kitchen Rosenfeld sonuçları, 4.2.b’de Plessey, 4.2.c’de SUSAN, 4.2.d’de ise CSS sonuçları gösterilmiştir. Kitchen Rosenfeld 86 köşeyi bulmakta ve diğerlerini bulamamaktadır. Yanlış olarak 25 noktayı işaretlemektedir. Diğer yandan, Plessey yönteminde doğru köşe bulma oranı 183/213’ken hatalı işaretleme sayısı 19’dur. SUSAN yönteminde doğru köşe bulma oranı ve hatalı işaretleme sayıları sırasıyla 185/213 ve 47’dir. CSS yöntemi bu oranları 159/213 ve 24 olarak vermektedir. Bu test görüntüsünde en başarılı yöntem doğru köşe bulma oranında SUSAN’dır. Buna rağmen, hatalı köşe işaretleme konusunda diğerlerine göre başarılı değildir.

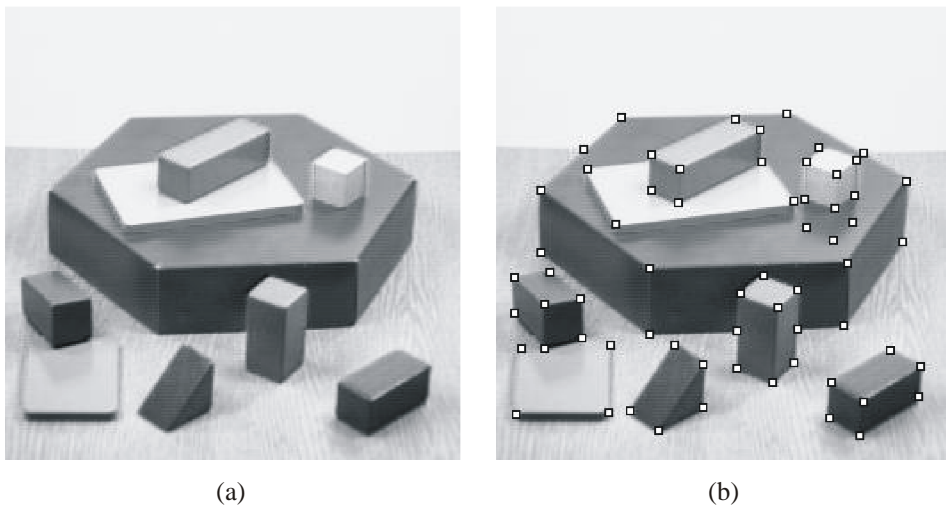


Şekil 4.1. a) 256x256 lab görüntüsü, b) gerçek köşeler

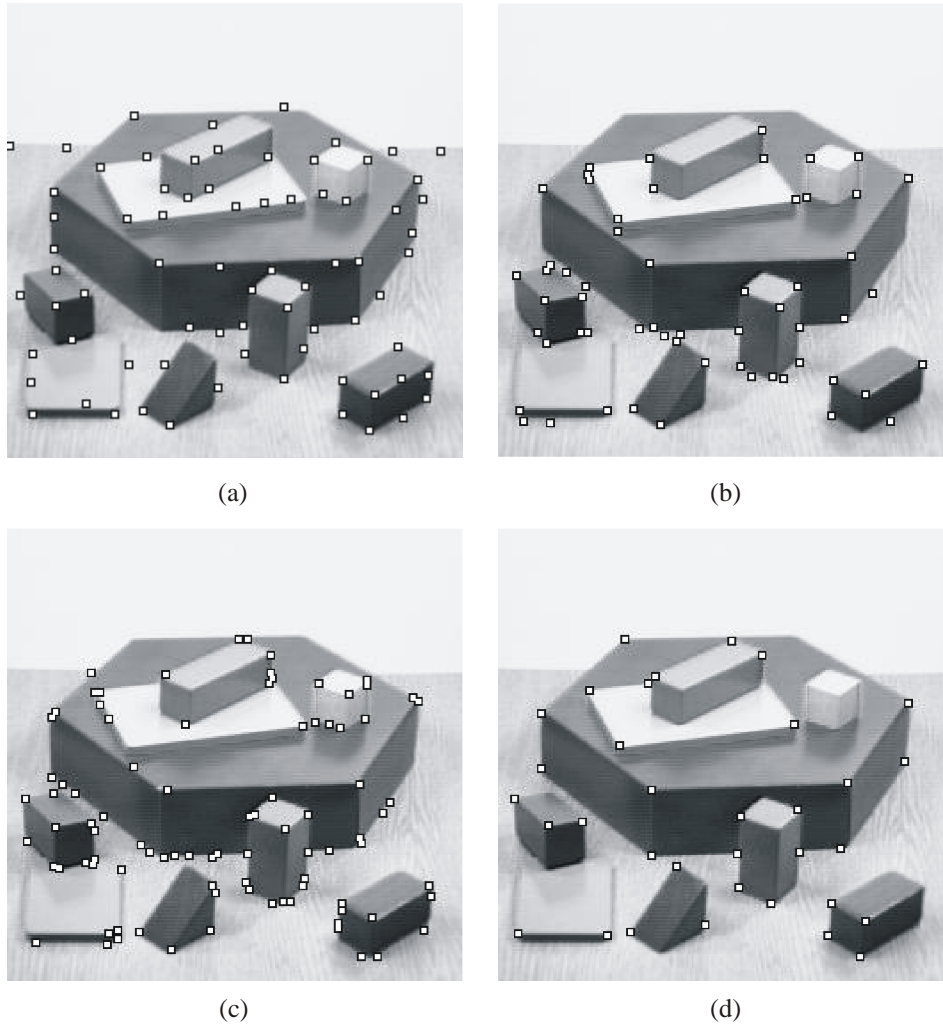
Şekil 4.3’de 256x256 büyüklükte olan blocks test görüntüsü bulunmaktadır. Blocks test görüntüsünün özelliği birbirine girmiş şekilde bulunan tahta bloklardır. Şekilde 4.4’de Kitchen-Rosenfeld, Plessey, SUSAN ve CSS yöntemleri için sonuçlar verilmektedir. Kitchen Rosenfeld (Şekil 4.4.a) 64 köşeden sadece 46’sını bulabilmekte ve 4 adet yanlış sonuç vermektedir. Plessey 42 doğru sonuç verirken, 3 yanlış bulunmaktadır. SUSAN’ın sonuçları diğerlerine göre daha başarılıdır ve 52 köşeyi bulabilmektedir. Ancak 9 tane yanlış ortaya çıkmaktadır. CSS yöntemi de 33 köşeyi doğru şekilde tespit ederken yanlış sonuç vermemektedir.



Şekil 4.2. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS



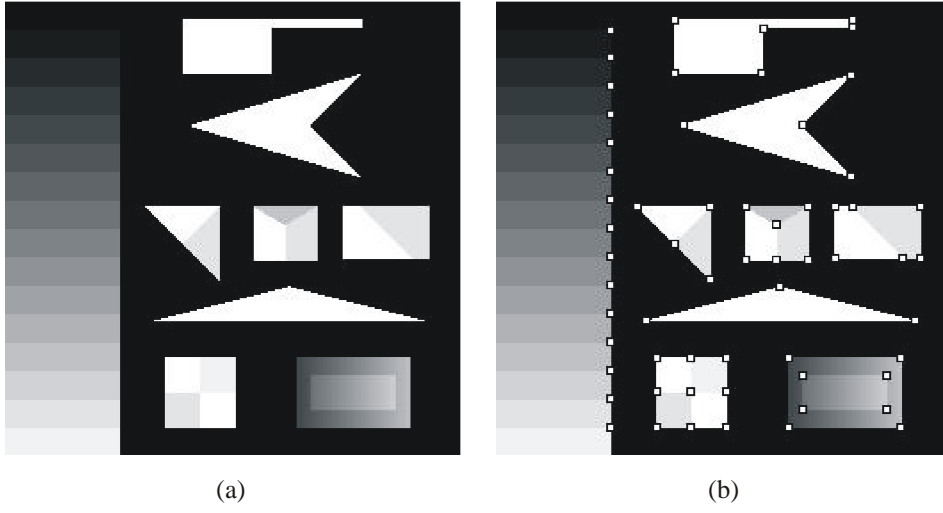
Şekil 4.3. a) 256x256 blocks görüntüsü, b) gerçek köşeler



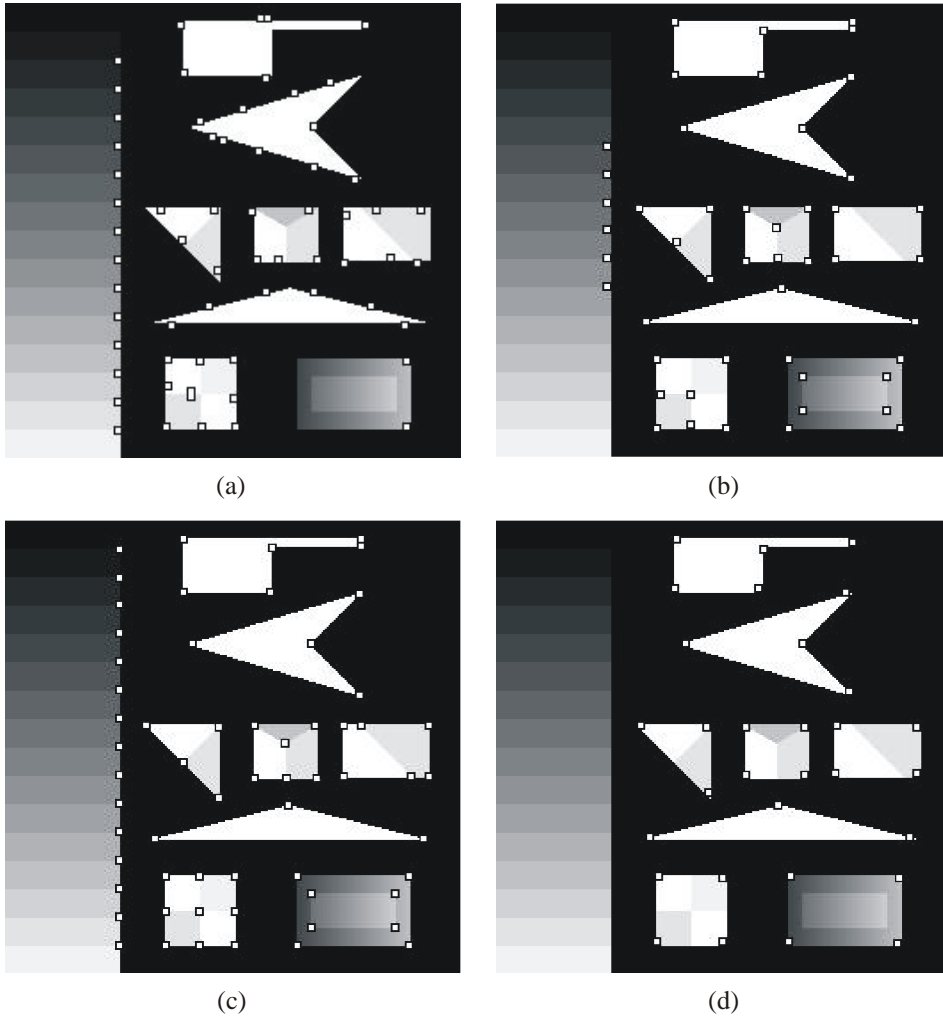
Şekil 4.4. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS

Şekil 4.5, Smith ve Brady'nin SUSAN makalesinde yer alan test görüntüsüdür. Bu görüntü kullanıldığında, SUSAN algoritması bütün köşeleri hata yapmaksızın bulmaktadır (Şekil 4.6.c). Kitchen-Rosenfeld ancak 61 köşeden 47 tanesini başarıyla bulmakta ve 14 yanlış köşe tespiti yapmaktadır. Plessey ise, 48 doğru köşeye karşın hiç hata yapmamaktadır. CSS algoritması 31 köşeyi hata olmaksızın tespit edebilmiştir.

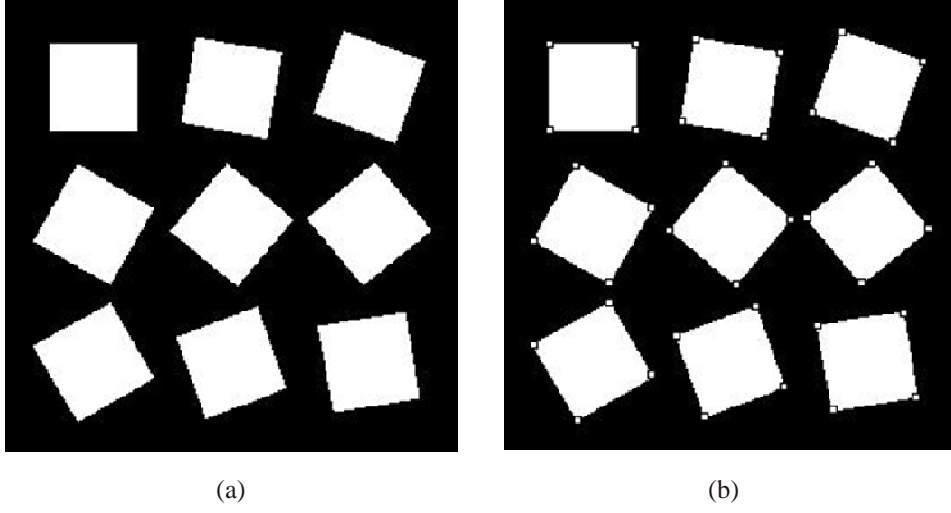
Şekil 4.7'de kare şeklindeki bir nesnenin döndürülmesi ile oluşan çeşitli açılardaki köşelerin bulunduğu test görüntüsü vardır. Bu görüntüde Kitchen-Rosenfeld dışındaki yöntemler tüm köşeleri hatasız olarak bulmuşlardır. Kitchen-Rosenfeld yöntemi ise 30 köşeyi doğru tespit etmiş fakat bunun yanında 42 tane yanlış köşe bulmuştur(Şekil 4.8.a).



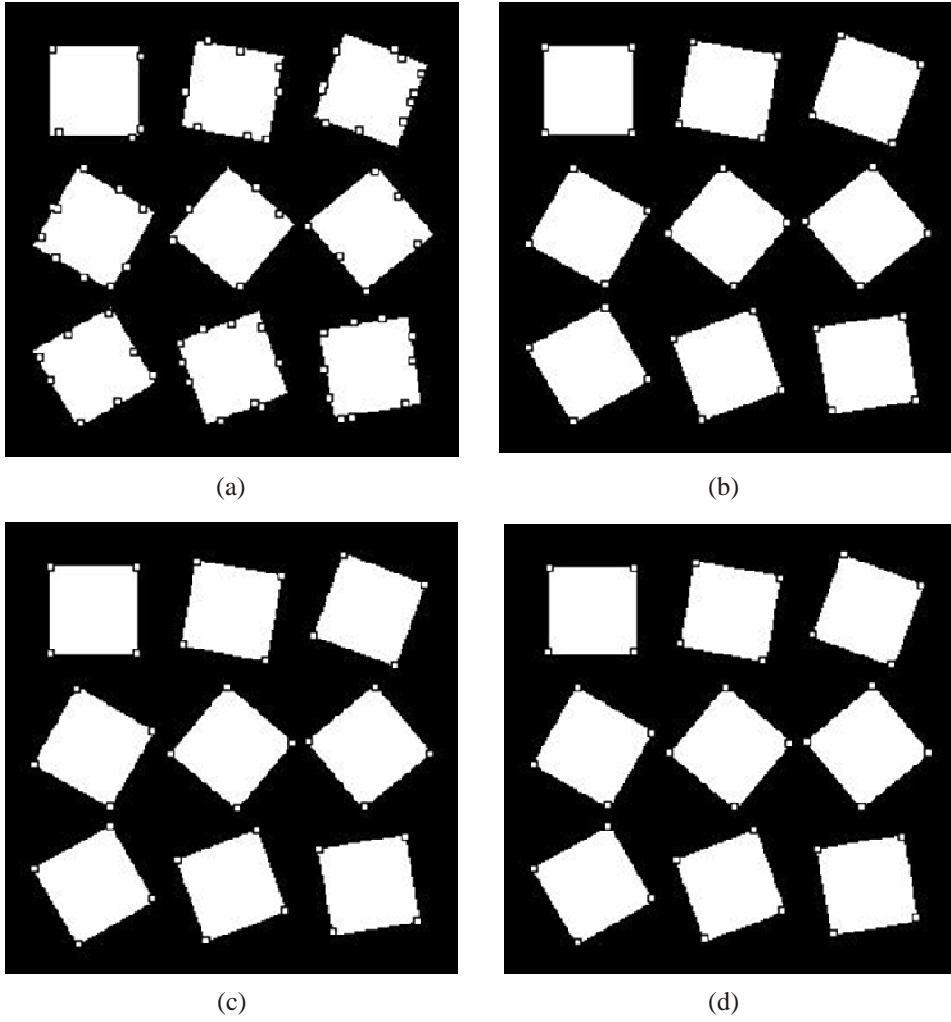
Şekil 4.5. a) 256x256 SUSAN test görüntüsü, b) gerçek köşeler



Şekil 4.6. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS

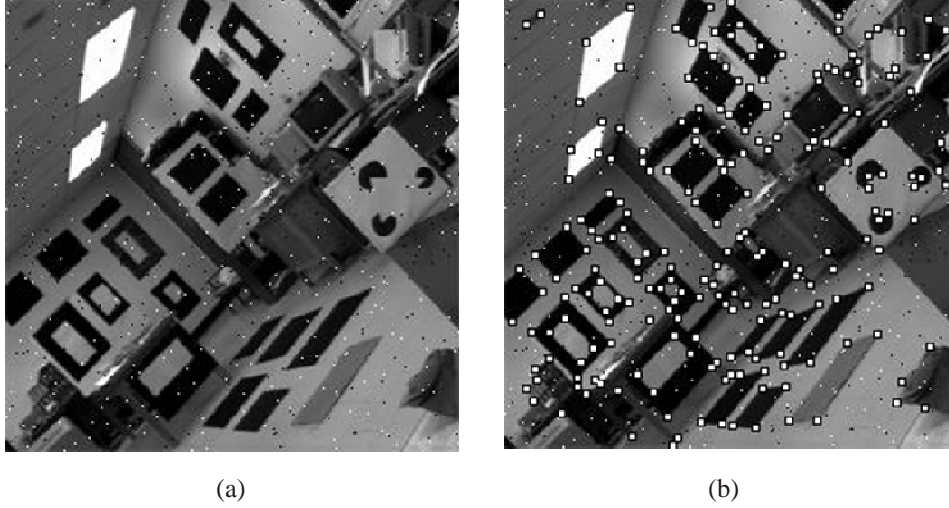


Şekil 4.7. a) 256x256 rotating squares görüntüsü, b) gerçek köşeler



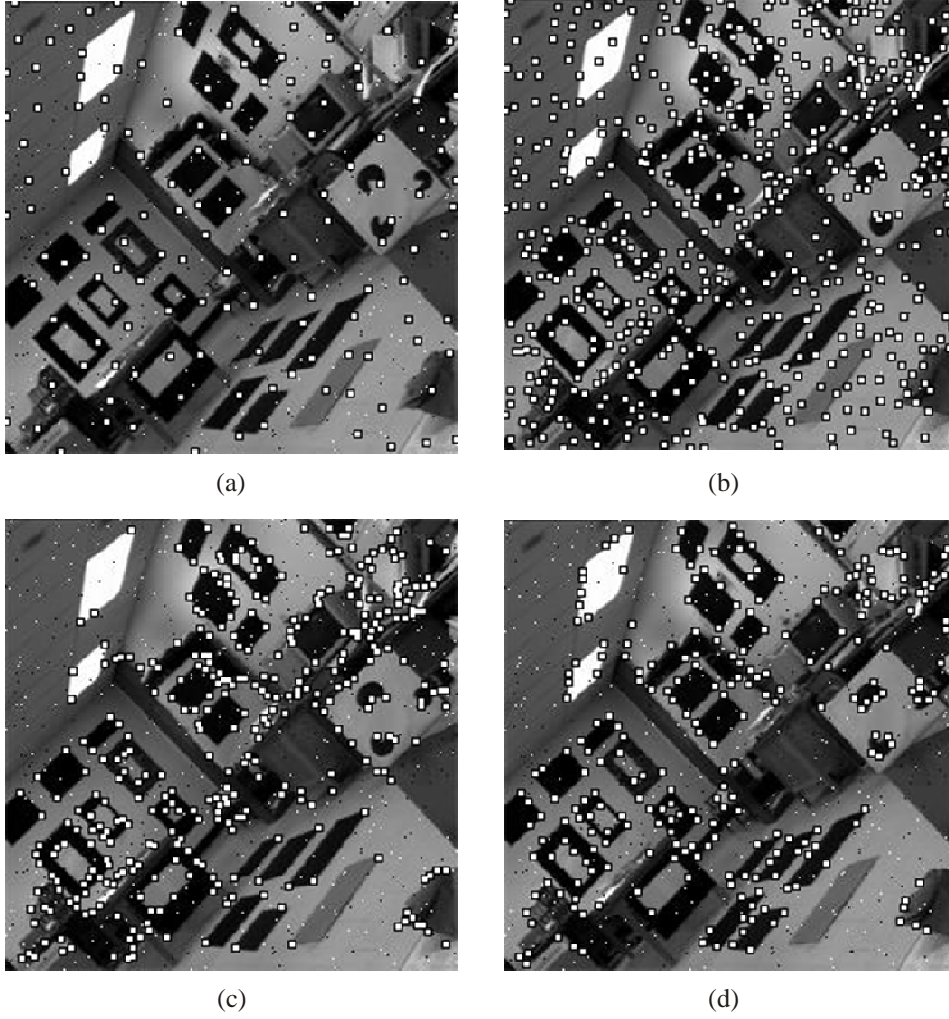
Şekil 4.8. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS

Şekil 4.9'dan itibaren yukarıda değerlendirdiğimiz dört adet test görüntüsüne 0.01 yoğunluklu salt&paper gürültüsü eklenerek elde edilmiş yeni görüntülerden oluşan sonuçlar yer almaktadır.



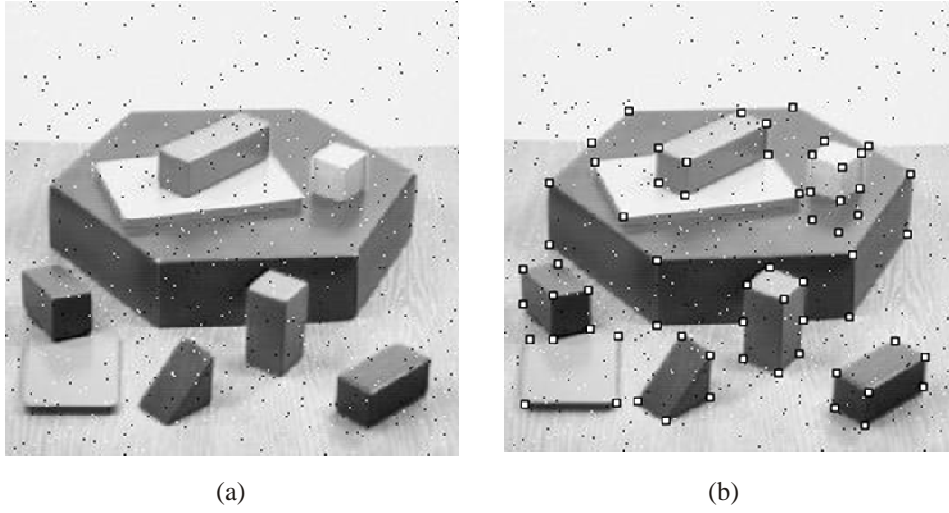
Şekil 4.9. a) 0.01 yoğunluklu salt&paper gürültüsü eklenmiş 256x256 laboratuvar görüntüsü, b) gerçek köşeler

Şekil 4.10.b'deki görüntüden anlaşılacağı gibi Plessey yöntemi gürültüye duyarlı bir yöntemdir. 202/213 doğru köşe tespiti ile en yüksek oran Plessey'e ait olmasına rağmen 211 hatalı köşe tespit ederek bu konuda da en yüksek orana sahiptir. Kitchen-Rosenfeld ise 80/213 doğru köşe tespit oranı ve 61 yanlış köşe tespiti ile Plessey'e oranla daha başarılıdır. SUSAN ve CSS yöntemleri ise gürültüden en az etkilenen yöntemler olmuştur. Doğru köşe tespit oranları sırasıyla 188/213, 168/213 iken hatalı köşe bulma sayıları sırasıyla 47 ve 24'dür.

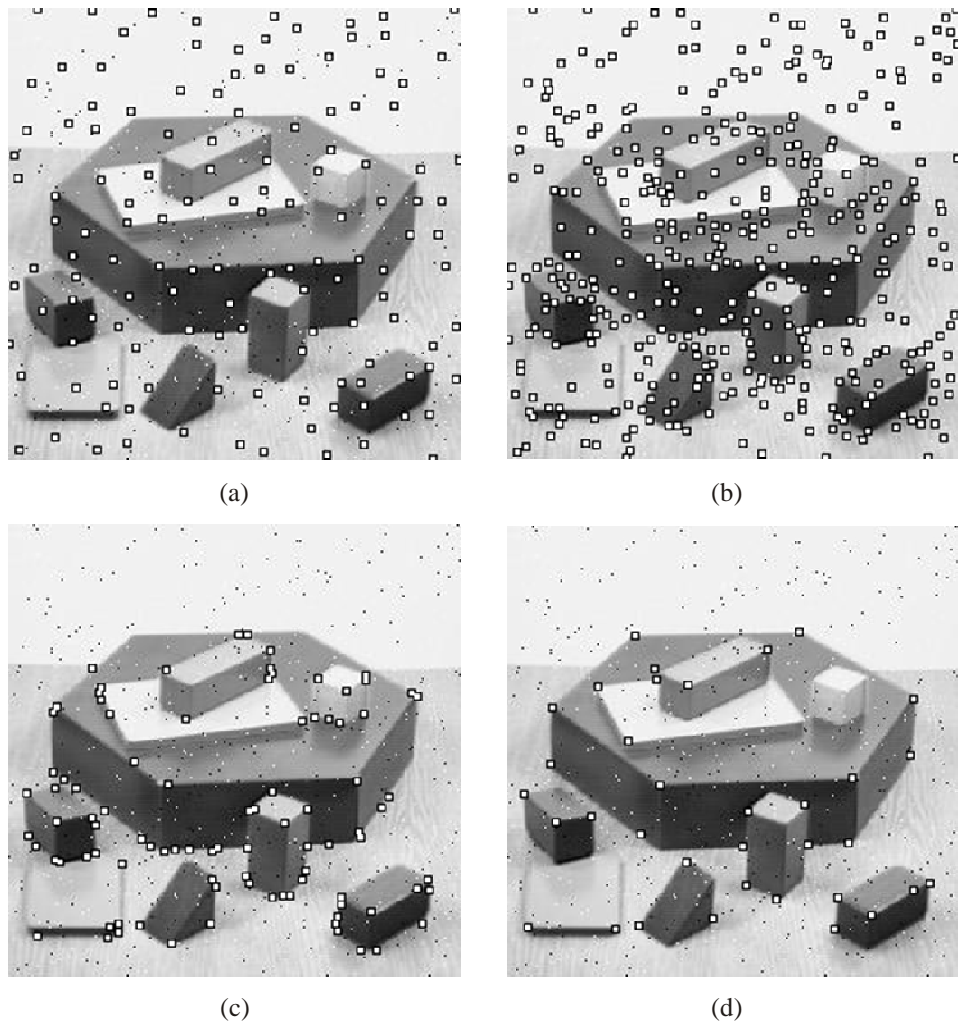


Şekil 4.10. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS

Şekil 4.12'deki sonuçlara bakıldığında CSS yöntemi bir tane hatalı köşe tesbit etmesine rağmen 64 köşeden sadece 32'sini tespit edebilmiştir. Plessey yöntemi 244 hatalı köşe ve 62/64 doğru köşe oranına, Kitchen-Rosenfeld ise 96 hatalı köşe tespiti ve 35/64 doğru köşe oranına sahipken SUSAN istikrarlı bir şekilde 46 hatalı köşe ve 52 doğru köşe tespit etmiştir. Böylece gürültü eklenmiş block görüntüsünde (Şekil 4.11.a) SUSAN en başarılı yöntem olarak öne çıkmaktadır.

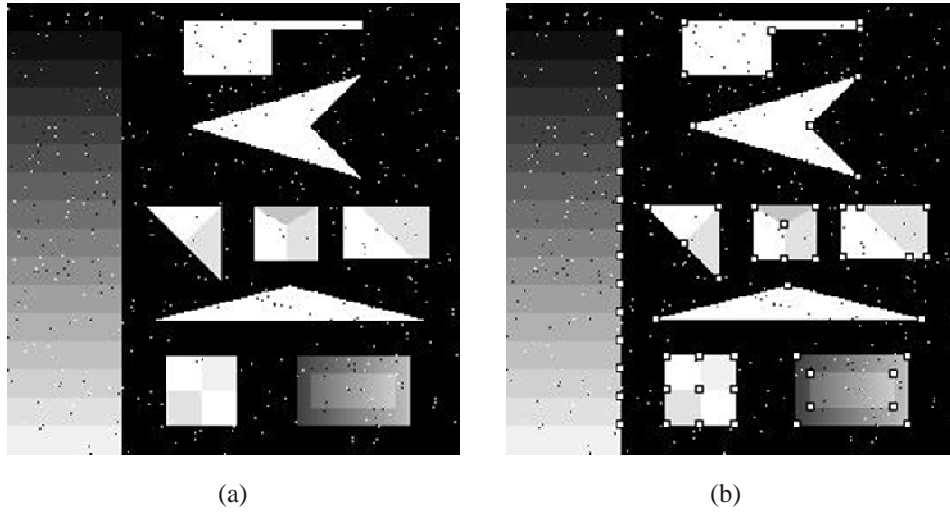


Şekil 4.11. a) 0.01 yoğunluklu salt&paper gürültüsü eklenmiş 256x256 blocks görüntüsü, b) gerçek köşeler



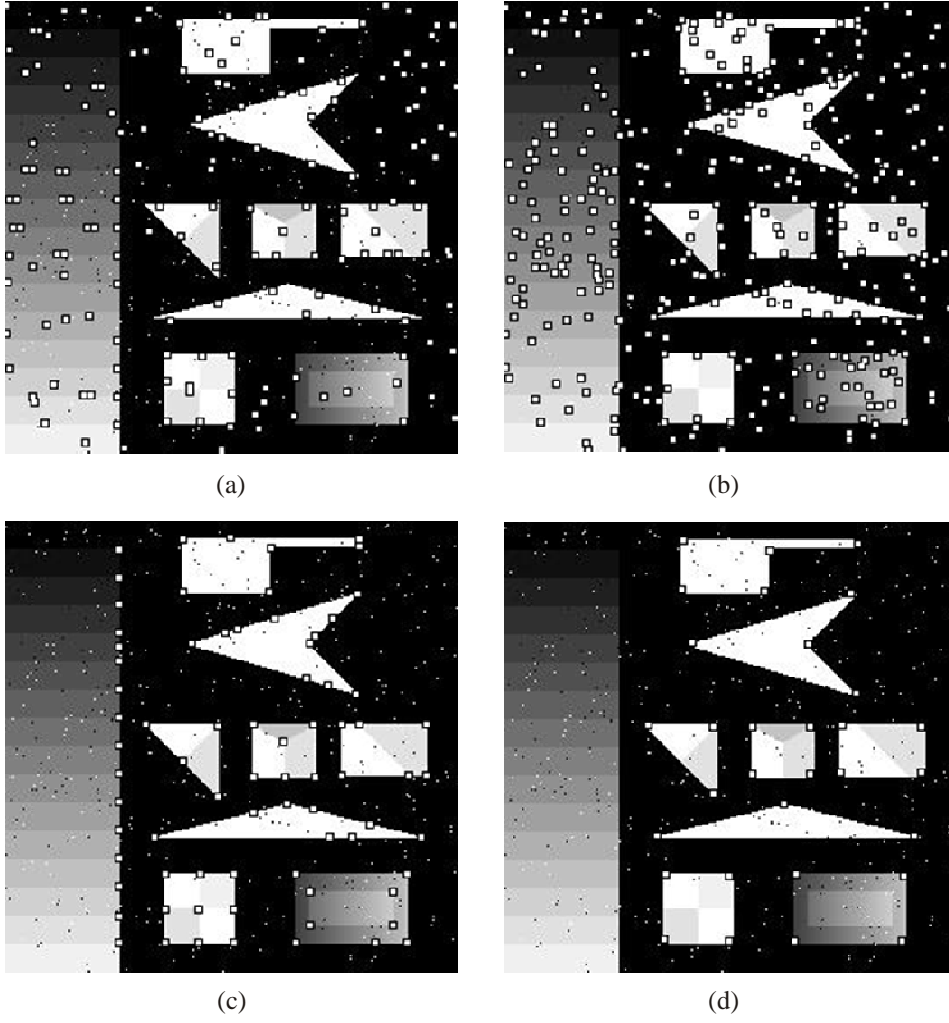
Şekil 4.12. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS

Şekil 4.13'deki görüntüye uygulanan köşe bulma yöntemleri arasında SUSAN yöntemi yine gürültüden çok az etkilenecek 13 hata ve 61/61 doğru köşe tespiti ile en başarılı yöntem olmuştur (Şekil 4.14.c). CSS yöntemi hiç hatalı köşe bulmamasına karşı 61 köşeden sadece 31 tanesini tespit ederek başarısız olmuştur. Plessey yöntemi gürültüye duyarlı olduğundan bu görüntüde de 56 doğru köşe tespitine karşılık 255 hatalı köşe tespit ederek en çok hata yapan yöntem olmuştur. Kitchen-Rosenfeld ise 116 hataya karşılık 51 doğru köşe tespit etmiştir.

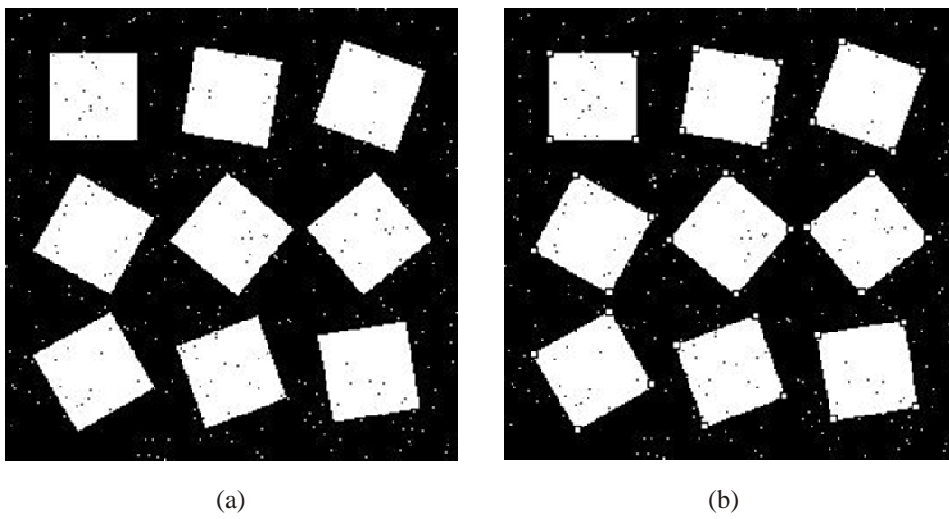


Şekil 4.13. a) 0.01 yoğunluklu salt&paper gürültüsü eklenmiş 256x256 SUSAN test görüntüsü, b) gerçek köşeler

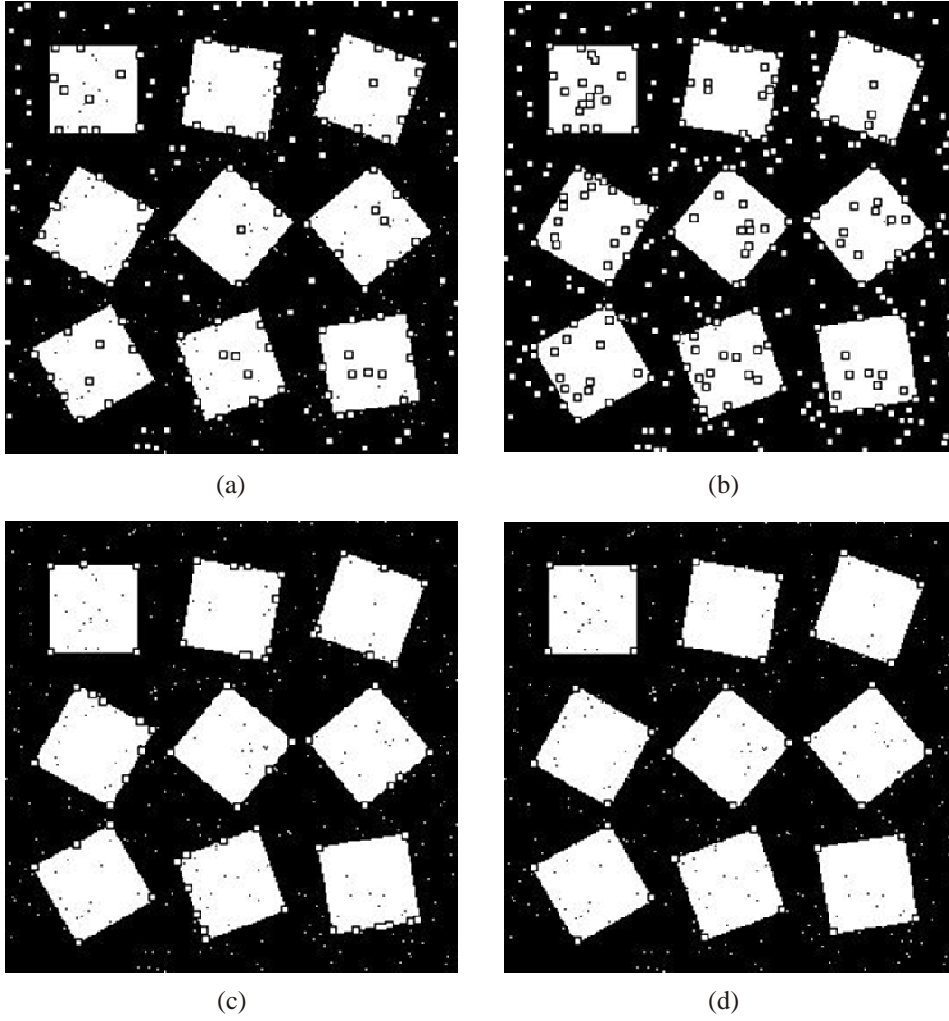
Şekil 4.15'deki görüntüden elde edilen sonuçlara (Şekil 4.16) bakıldığında gürültüden az etkilenen CSS yöntemi hata yapmaksızın tüm doğru köşeleri tespit edebilmiştir. SUSAN yöntemi ise 23 hatalı köşe ve 36/36 doğru köşe tespit etmiştir. Kitchen-Rosenfeld ile Plessey yöntemleri ise sırasıyla 116 ve 255 hatalı köşe tespit etmişler. Buna karşılık yine sırasıyla 30/36 ve 36/36 doğru köşe tespiti yapabilmişlerdir.



Şekil 4.14. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS



Şekil 4.15. a) 0.01 yoğunluklu salt&paper gürültüsü eklenmiş 256x256 rotating squares görüntüsü, b) gerçek köşeler



Şekil 4.16. Sonuçlar a) Kitchen Rosenfeld, b) Plessey, c) SUSAN, d) CSS

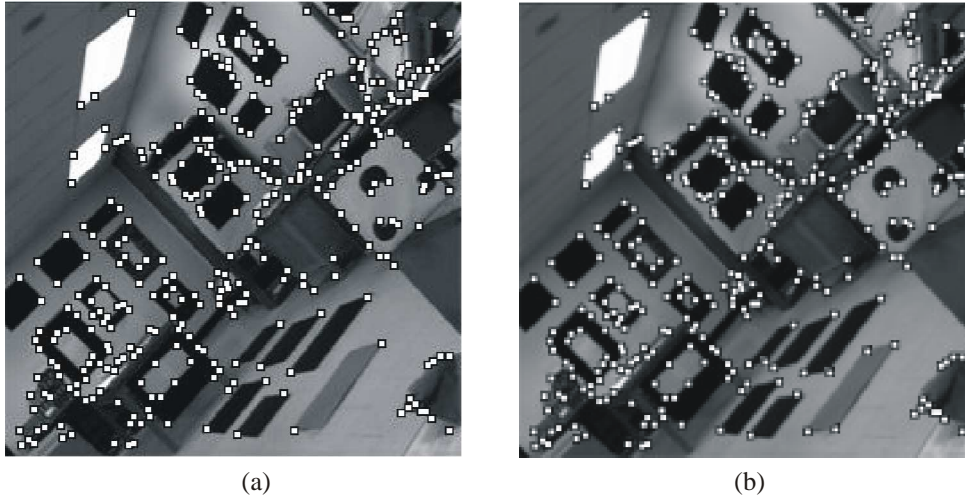
İncelenen dört test görüntüsü ve dört köşe bulma yönteminin sonuçlarına bakıldığında SUSAN yönteminin gürültülü ve gürültüsüz görüntülerdeki genel başarısı nedeniyle geliştirilen yöntemin SUSAN yöntemine uygulanmasına karar verilmiştir.

4.3 Topolojik SUSAN Yönteminin Sonuçları

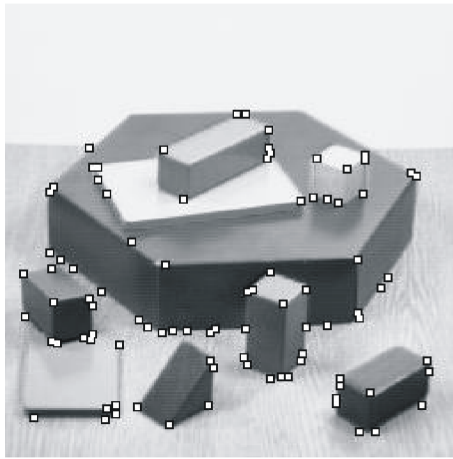
Bu bölümde, parlak ve koyu nesnelere esas alınarak oluşturulan SUSAN temelli köşe görüntüleriyle elde edilen sonuçlar, diğer köşe bulma algoritmalarıyla karşılaştırılmıştır. Maskenin 9x9 haline getirilmesi ve BDHp ve BDHk görüntülerinin kullanılmasıyla, topolojik SUSAN yönteminin klasik SUSAN'a göre yanlış köşe oranında azalma görülmüştür.

Şekil 4.17.a ve 4.17.b'de, Laboratuvar test görüntüsü üzerinde geleneksel SUSAN ve topolojik SUSAN yöntemlerinin sonuçları gösterilmektedir. Topolojik SUSAN yöntemi bu görüntüde, 213 köşeden 189'unu doğru olarak bulurken, 33 köşede hata yapmaktadır. Diğer yandan, geleneksel SUSAN 185 köşeyi bulurken, 47 yanlış köşe bulmuştur. Topolojik yaklaşımın, geleneksel SUSAN'a göre yanlış köşe bulma oranını azalttığı anlaşılmaktadır.

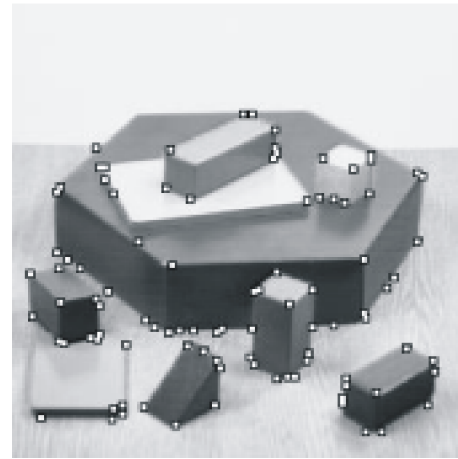
Blocks isimli test görüntüsü üzerinde uygulanan geleneksel SUSAN ve topolojik SUSAN yöntemlerinin sonuçları Şekil 4.18.a ve 4.18.b'de gösterilmektedir. Geleneksel SUSAN'da başarı oranı, 52/64'ken topolojik SUSAN'da 57/64 olmaktadır. Geleneksel yöntemde hata sayısı 9'ken topolojik olanda bu sayı 5'e inmektedir. SUSAN yönteminin önerildiği makalede kullanılan test görüntüsünde (Şekil 4.5) ve rotating squares görüntüsünde (Şekil 4.7), geleneksel ve topolojik SUSAN yöntemlerinin sonuçları birbirinin aynısıdır (Şekil 4.19, 4.20).



Şekil 4.17. a) SUSAN, b) topolojik SUSAN

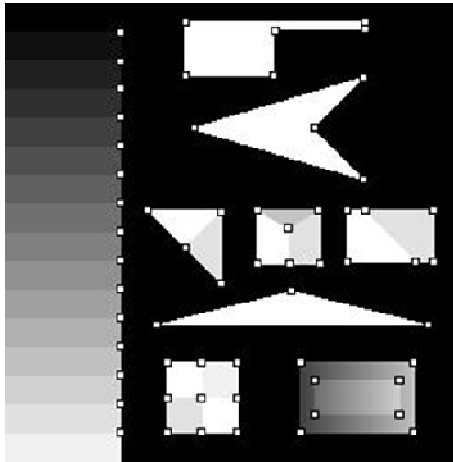


(a)

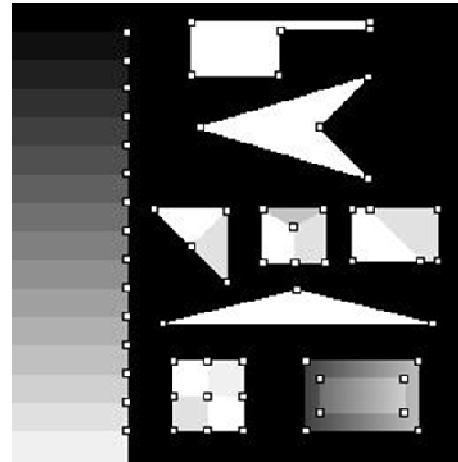


(b)

Şekil 4.18. a) SUSAN, b) Topolojik SUSAN

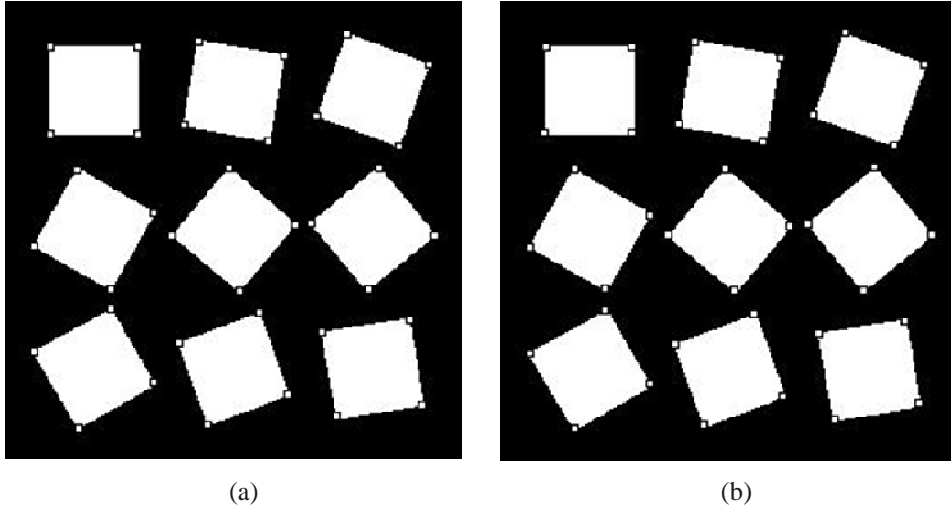


(a)



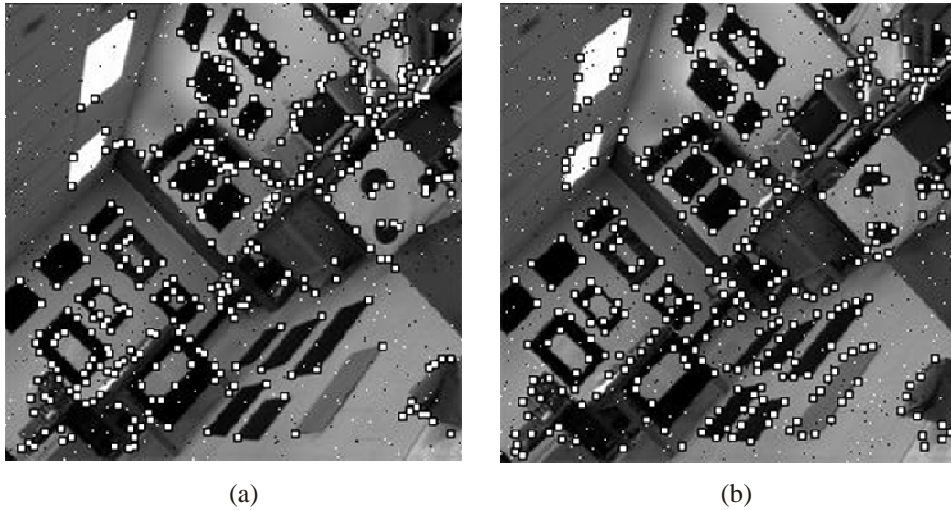
(b)

Şekil 4.19. a) SUSAN, b) Topolojik SUSAN



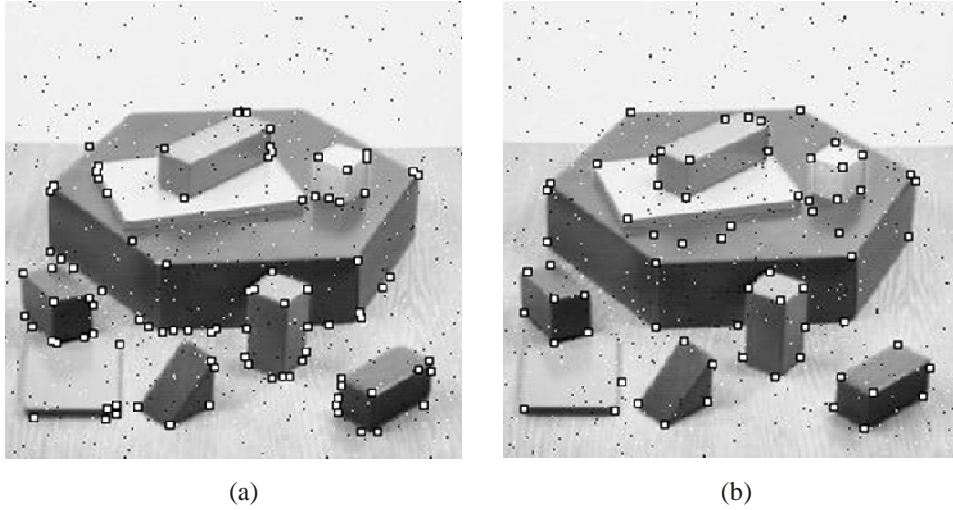
Şekil 4.20. a) SUSAN, b) Topolojik SUSAN

Şekil 4.21'den sonraki gürültülü görüntülerde de gürültüsüz görüntülerde olduğu gibi Klasik SUSAN yöntemine göre gelişme sağlanmıştır. Gürültülü görüntüler, önceki görüntülere 0.01 yoğunluklu salt&paper gürültüsü eklenerek elde edilmiştir. Laboratuvar görüntüsünde (Şekil 4.21) klasik SUSAN yöntemi 213 köşeden 188 doğru köşeyi tespit ederken topolojik SUSAN ise 195 doğru köşe tespit etmiştir. Aynı şekilde hata sayısında da 46'dan 44'e bir düşüş gözlenmiştir.



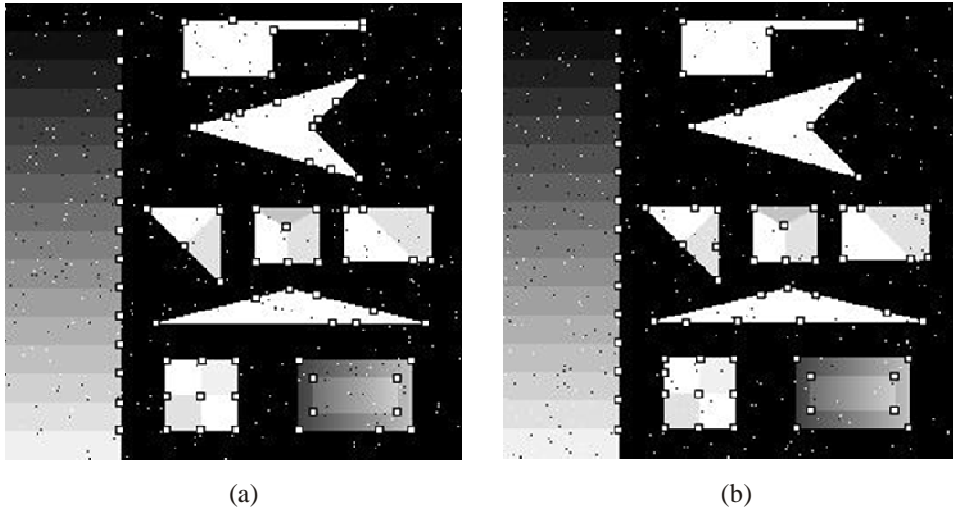
Şekil 4.21. a) SUSAN, b) Topolojik SUSAN

Şekil 4.22'deki blocks test görüntüsünde ise Topolojik SUSAN 64 köşeden 59'unu, klasik SUSAN ise 52'sini tespit etmiştir. Hata sayısında ise 10'dan 6'a yine düşüş gözlenmiştir.



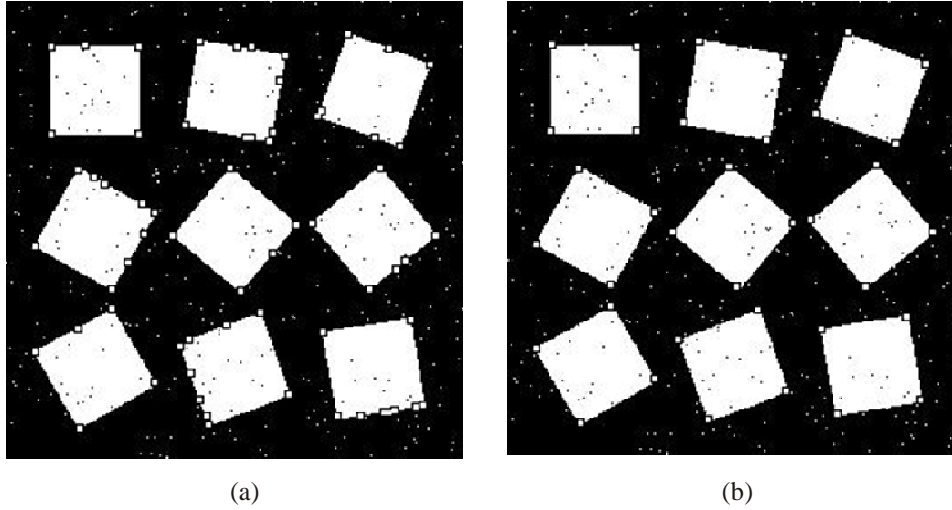
Şekil 4.22. a) SUSAN, b) Topolojik SUSAN

Şekil 4.23’de klasik SUSAN ile Topolojik SUSAN yöntemleri tüm köşeleri tespit etmişlerdir. Fakat klasik SUSAN 13 tane yanlış köşe tespit ederken Topolojik SUSAN 9 hatalı köşe tespit etmiştir.



Şekil 4.23. a) SUSAN, b) Topolojik SUSAN

Şekil 4.24’de çeşitli açılardaki köşelerin bulunduğu gürültü eklenmiş rotating squares test görüntüsü üzerinde klasik SUSAN ve Topolojik SUSAN ile elde edilen sonuçları bulunmaktadır. Bu görüntüde klasik SUSAN yöntemi tüm köşeleri bulmasına rağmen 23 taen hatalı köşe bulmuştur. Topolojik SUSAN ise hata yapmaksızın tüm köşeleri doğru olarak tespit edebilmiştir.



Şekil 4.24. a) SUSAN, b) Topolojik SUSAN

Çizelgelerde Topolojik SUSAN yöntemi ile diğer dört yöntemin köşe bulma süreleri (Çizelge 4.1, 4.4), köşe bulma başarıları (Çizelge 4.2, 4.5) ve köşe bulurken ürettikleri hatalı köşe sayılarına göre (Çizelge 4.3, 4.6) karşılaştırması yer almaktadır. Köşe bulma süresinde diğer yöntemlere göre daha kötü dereceler alınmış olsa da elde edilen doğru köşe sayısı ve düşük hata sayısı ile tatmin edici bir iyileştirme gerçekleştirilmiştir.

Çizelge 4.1 Yöntemlerin köşe bulma sürelerine göre karşılaştırması.

Görüntü/Yöntem	Kitch.	Plessey	SUSAN	CSS	T.SUSAN
Laboratuar	1.136sn	0.169sn	1.044sn	1.067sn	6.473sn
Blocks	1.103sn	0.161sn	0.995sn	0.726sn	5.921sn
SUSAN Test	0.888sn	0.152sn	0.977sn	0.723sn	6.057sn
Rotating Squares	1.018sn	0.472sn	0.123sn	0.787sn	0.763sn

Çizelge 4.2 Yöntemlerin köşe bulma başarılarına göre karşılaştırması.

Görüntü/Yöntem	Kitch.	Plessey	SUSAN	CSS	T.SUSAN
Laboratuar	86/213	183/213	185/213	159/213	189/213
Blocks	47/64	42/64	52/64	33/64	57/64
SUSAN Test	47/61	48/61	61/61	31/61	61/61
Rotating Squares	30/36	36/36	36/36	36/36	36/36

Çizelge 4.3 Yöntemlerin köşe bulmada ürettikleri hataya göre karşılaştırması.

Görüntü/Yöntem	Kitch.	Plessey	SUSAN	CSS	T.SUSAN
Laboratuvar	25	45	47	24	33
Blocks	26	3	9	0	5
SUSAN Test	14	0	0	0	0
Rotating Squares	42	0	0	0	0

Çizelge 4.4 Gürültü eklenmiş görüntülerdeki yöntemlerin köşe bulma sürelerine göre karşılaştırması.

Görüntü/Yöntem	Kitch.	Plessey	SUSAN	CSS	T.SUSAN
Laboratuvar	1.116sn	0.163sn	1.065sn	1.116sn	6.603sn
Blocks	1.104sn	0.163sn	1.038sn	0.746sn	6.421sn
SUSAN Test	0.932sn	0.157sn	0.983sn	0.743sn	6.095sn
Rotating Squares	0.954sn	0.157sn	0.981sn	0.674sn	6.082sn

Çizelge 4.5 Gürültü eklenmiş görüntülerdeki yöntemlerin köşe bulma başarılarına göre karşılaştırması.

Görüntü/Yöntem	Kitch.	Plessey	SUSAN	CSS	T.SUSAN
Laboratuvar	80/213	202/213	188/213	168/213	195/213
Blocks	35/64	62/64	52/64	32/64	59/64
SUSAN Test	51/61	56/61	61/61	31/61	61/61
Rotating Squares	30/36	36/36	36/36	36/36	36/36

Çizelge 4.6 Gürültü eklenmiş görüntülerdeki yöntemlerin köşe bulmada ürettikleri hataya göre karşılaştırması.

Görüntü/Yöntem	Kitch.	Plessey	SUSAN	CSS	T.SUSAN
Laboratuvar	61	211	46	28	52
Blocks	96	244	10	1	6
SUSAN Test	116	255	13	0	9
Rotating Squares	131	262	23	0	0

5. SONUÇ

Dört farklı test görüntüsü üzerinde yapılan çalışmalar sonucunda, topolojik SUSAN'ın geleneksel olana göre daha fazla köşeyi bulduğu anlaşılmaktadır. Ayrıca, hata oranlarında da ciddi bir düşüş sağlamaktadır. Bunun temel nedenlerinden biri, köşelerin parlak nesnelere içindeki ve koyu nesnelere göre durumları ayrıca incelenip köşeler bulunduğundan, algoritmanın bu şekilde daha iyi çalıştığı anlaşılmaktadır.

Bununla birlikte, bağlantı derecesi haritalarının hazırlanması algoritma için ek işlem gücü gerektirmekte olduğundan, geleneksel yöntemlere göre daha yavaş çalışmaktadır. Ancak bilgisayarların her geçen gün hızlanması artan işlem zamanını telafi edebilmektedir.

Bu tez çalışmasında sadece SUSAN yönteminin topolojik olarak geliştirilmesi yapılmasına rağmen, bulanık topolojik kavramların diğer köşe bulma yöntemlerine de eklenebileceği açıktır. Özellikle BDHp ve BDHk haritalarının, yerel komşuluklar kullanan diğer yöntemlerde de ek bilgi sağlayacağı ve bunlardan elde edilen sonuçlarla köşe bulma başarımının artırılması beklenebilir.

KAYNAKLAR

- [1] F. Mokhtarian, and R. Suomela, "Robust image corner detection through curvature scale space," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, **20**(12), pp. 1376-1381, Dec, 1998.
- [2] H.P. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Proc. Int'l Joint Conf. Artificial Intelligence*, p. 584, 1977.
- [3] C. Harris, and M. Stephens, "Combined Edge and Corner Detection," *Fourth Alvey Vision Conference*, pp. 147-151, 1988.
- [4] P. R. Beaudet, "Rotational invariant image operators," in *4th International Conference on Pattern Recognition*, Tokyo, Japan, 1978, pp. 579-583.
- [5] L. Kitchen, and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, **1**(1), pp. 8, Dec, 1982.
- [6] D. M. Tsai, "Boundary-based corner detection using neural networks," *Pattern Recognition*, **30**(1), pp. 85-97, Jan, 1997.
- [7] 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.
- [8] J. Q. Fang, and T. S. Huang, "A Corner Finding Algorithm for Image Analysis and Registration," in *AAAI Conference*, 1982, pp. 46-49.
- [9] 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.
- [10] 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] R. Mehrotra, S. Nichani, and N. Ranganathan, "Corner Detection," *Pattern Recognition*, **23**(11), pp. 1223-1233, 1990.
- [17] 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.
- [18] 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.
- [19] A. Rosenfeld, "Fuzzy Digital Topology," *Information and Control*, vol. 1, pp. 76-87, January 1979, 1979.
- [20] S. Dellepiane, and F. Fontana, "Extraction of intensity connectedness for image-processing," *Pattern Recognition Letters*, **16**(3), pp. 313-324, Mar, 1995.
- [21] J. K. Udupa, and S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *Graphical Models and Image Processing*, **58**(3), pp. 246-261, May, 1996.
- [22] H. G. Senel, R. A. Peters, and B. Dawant, "Topological median filters," *Ieee Transactions on Image Processing*, **11**(2), pp. 89-104, Feb, 2002.
- [23] H. G. Senel, "Gradient Estimation Using Wide Support Operators", *IEEE Trans. on Image Processing*, yayın için kabul edildi.
- [24] U. Braga-Neto, and J. Goutsias, "A Theoretical Tour of Connectivity in Image Processing and Analysis," *Journal of Mathematical Imaging and Vision*, **19**, pp. 5-31, 2003.