

**BULANIK TOPOLOJİYE DAYALI
KÖŞE BULMA ALGORİTMASI**

Aysun TOK
Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı
Kasım-2007

JÜRİ VE ENSTİTÜ ONAYI

Aysun Tok'un **Bulanık Topolojiye Dayalı Köşe Bulma Algoritması** başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Yüksek Lisans tezi 19.10.2007 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. YUSUF OYSAL
Üye (2. Tez Danışmanı)	:Yard. Doç. Dr. HAKAN G. ŞENEL
Üye	:Doç. Dr. AHMAD BABANLI
Üye	:Yard. Doç. Dr. HÜSEYİN POLAT
Üye	:Yard. Doç. Dr. EMİN GERMEN

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

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

BULANIK TOPOLOJİYE DAYALI KÖŞE BULMA ALGORİTMASI

Aysun TOK

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Yusuf OYSAL

İkinci Danışman: Yard. Doç. Dr. Hakan Güray ŞENEL

2007, 51 sayfa

Köşeler, görüntü işleme tekniklerinde çok kullanışlı olan özelliklerdendir. Nesne tanıma, hareket izleme, stereo eşleştirme gibi yöntemler içinde sıklıkla kullanılmaktadır. Görüntüde yer alan nesnelere hakkında gerekli olan birçok bilgiyi verir. Nesnenin tanımlanması, ayırt edilmesi ve belirlenebilmesi için köşelerden faydalanmak mümkündür. Gri-seviye görüntü içerisinde köşelerin sezimini gerçekleştiren yöntemler, kenar bulma tabanlı yöntemler ve yoğunluk tabanlı yöntemler olarak gruplandırılabilir. Bu tez çalışmasında bulanık topolojiye dayanan bir köşe bulma algoritmasının iyileştirilmesi ve diğer yöntemlere göre performansının değerlendirilmesi gerçekleştirilmiştir.

Anahtar Kelimeler: Görüntü işleme, Köşe bulma, Topolojik Ortanca Süzgeçler

ABSTRACT

Master of Science Thesis

FUZZY TOPOLOGY BASED CORNER DETECTION ALGORITHM

Aysun TOK

**Anadolu University
Graduate School of Natural and Applied Science
Computer Engineering Program**

Supervisor: Assist. Prof. Dr. Yusuf OYSAL

Second Supervisor: Assist. Prof. Dr. Hakan Güray ŞENEL

2007, 51 pages

Corners are very useful features in image processing and computer vision. They frequently employed in the object recognition, motion detection and segmentation. Corners in images represent a lot of useful information and they play an important role in describing object features for recognition and identification. The methods that detects corners in images can be categorized as edge-detection based methods and intensity based methods. In this thesis we propose to improve a fuzzy topology based corner detection algorithm and to evaluate it's performance with the other methods.

Keywords: Image Processing, Corner Detection, Topological Median Filters

TEŐEKKÜR

Tezin her aŐamasında bilgisini, yardımını ve desteęini esirgemeyen yardımcı danıŐmanım Yard. Doę. Dr. Hakan G¼ray ŐENEL'e ok teŐekk¼r ederim. DanıŐmanım Yard. Doę. Dr. Yusuf OYSAL'a, dięer j¼ri ¼yelerime ve Ahmet Okan ONARCAN'a ayrıca teŐekk¼r ederim.

Aysun TOK

Kasım - 2007

İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	viii
1. GİRİŞ	1
1.1 Kenar Bulma Tabanlı Yöntemler	3
1.2 Yoğunluk Tabanlı Yöntemleri	4
2. KÖŞE BULMA YÖNTEMLERİ	5
2.1 Kitchen ve Rosenfeld Yöntemi	5
2.2 Harris Yöntemi.....	8
2.3 SUSAN Yöntemi.....	11
2.4 CSS Yöntemi.....	14
2.5 Laganiere Yöntemi.....	17
3. DİJİTAL TOPOLOJİ ve BULANIK DİJİTAL TOPOLOJİ	19
3.1 Bağlılık.....	19
3.2 Bulanık Küme Teorisi.....	20
3.3 Bulanık Topolojinin Tanımı ve Özellikleri	21
3.4 α -Bağlantılılık Kavramı ve Özellikleri	23
3.5 Bağlantı Haritası (The Connectivity Map)	24
3.6 Bağlantı Haritasının Özellikleri	26
3.7 Birbirine Bağlı Bileşenlerin İşaretlenmesi (Connected Component Labeling-CCL)	29

4. TOPOLOJİK ORTANCA SÜZGEÇLER.....	31
4.1 Topolojik Ortanca Süzgeçlerin Etkileri	35
4.2 Topolojik Ortanca Yöntemiyle Köşe Bulma	35
5. TOPOLOJİK ORTANCA YÖNTEMİYLE KÖŞE BULMA ALGORİTMASININ İYİLEŞTİRİLMESİ.....	39
5.1 İyileştirilmiş Algoritmanın Karşılaştırma Sonuçları.....	41
5.2 İyileştirilmiş Algoritmanın Diğer Köşe Bulma Yöntemleri ile Karşılaştırma Sonuçları.....	42
6. SONUÇ.....	48
KAYNAKLAR	49

ŞEKİLLER DİZİNİ

1.1 Farklı şekillerdeki köşe çeşitleri	2
2.1 Plessey işlecinin karşılaştığı farklı durumlar	9
2.2 Özdeğer uzayının farklı bölgelere ayrılması	10
2.3 Dairesel maskeler kullanarak USAN yönteminin bir görüntüye uygulanması	12
2.4 CSS'nin karşılaştığı durumlar	15
2.5 Laganiere'in kullandığı 4 farklı yapı elemanı	18
3.1 4'lü komşuluk ve 8'li komşuluk grafikleri	20
3.2 p ve q piksellerini birbirine bağlayan iki farklı yoldaki yoğunluk değerleri	23
3.3 7x7'lik pencere içindeki görüntünün bağlantı haritası	25
3.4 İki yakın çizgi nesnesinin oluşturduğu görüntü kesiti	26
3.5 Gürültü eklenmiş görüntü kesiti ve bağlantı haritası	27
3.6 Adım kenarın kesiti	28
3.7 CCL yönteminin uygulanması ve CCL yöntemiyle işaretlenmiş görüntü kesiti	30
4.1 3x3 görüntü kesitine klasik ortanca yöntemi TL ve TD uygulanması	32
4.2 TL ve TD uygulanmış görüntüler	33
4.3 TLD ve TDL uygulanmış test görüntüleri	34
4.4 Topolojik Ortanca Yöntemi uygulanarak köşeleri tespit edilen test görüntüleri	38
5.1 Topolojik ortanca yöntemiyle köşe bulma algoritmasının sonuçları	40
5.2 "Lab" görüntüsünün bir kesiti üzerinde önceki algoritmanın sonuçları	41
5.3 "Lab" görüntüsünün bir kesiti üzerinde iyileştirilen algoritmanın sonuçları	42
5.4 Yapay Test Görüntüsünün Sonuçları	46
5.5 Blok Görüntüsünün Sonuçları	47

ÇİZELGELER DİZİNİ

5.1. Şekil 5.4 için köşe bulma sonuçları.....	44
5.2. Şekil 5.5 için köşe bulma sonuçları.....	44

SİMGELER ve KISALTMALAR DİZİNİ

Σ	: Kesikli uzayda görüntü matrisi
$\mu_A(p)$: p'nin üyelik fonksiyonu
\mathcal{V}	: p ve q pikselleri arasındaki yol
$S_E(\rho)$: Yolun bağlantı derecesi
DOM	: Üyelik derecesi
DOC	: Bağlantılılık derecesi
BDH	: Bağlantılılık derecesi haritası
I	: İki boyutlu görüntü matrisi
TL	: Topolojik aşınma operatörü
TD	: Topolojik genişleme operatörü
TLD	: Topolojik kapama operatörü
TDL	: Topolojik açma operatörü
TOS	: Topolojik ortanca süzgeçler
KOS	: Klasik ortanca süzgeç

1. GİRİŞ

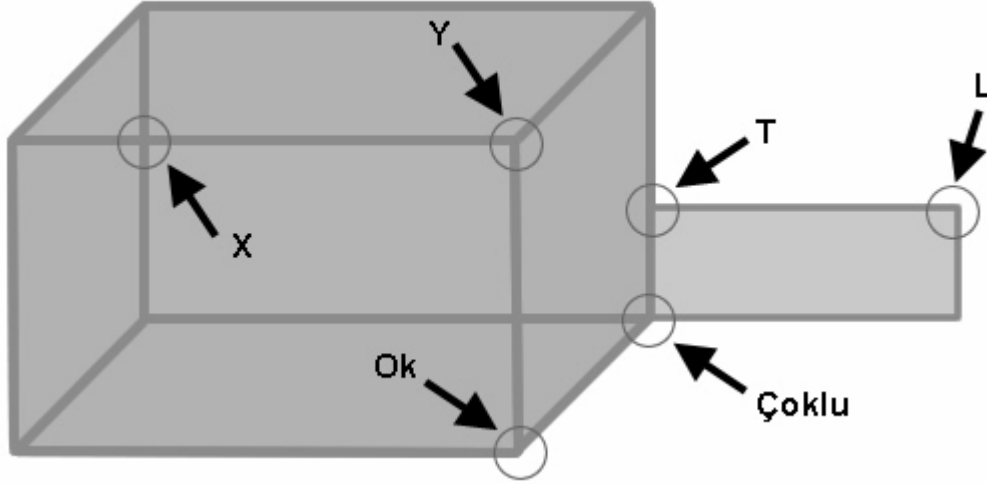
Görüntü işleminin en önemli araştırma konularından biri nesne tanımadır. Bir görüntüdeki nesnelere görüntünün en göze çarpan ve en anlamlı öğeleridir. Nesnelere görüntüden çıkarabilmek ya da nesnenin yerini tespit edebilmek için nesnenin konumsal veya şekilsel özelliklerinin belirlenmesi gerekir. Genellikle nesneyi görüntüden çıkarabilmek için kenar çizgileri kullanılır. Ancak özellikle video görüntülerinde şekilleri, cisimleri takip edebilmek için kenarlara nazaran daha güvenilir, anlamlı ve bağımsız olan noktasal özellikler kullanılır. Bu noktalar ilgi noktaları ya da köşelerdir.

Görüntü işlemede geniş bir araştırma konusu olmasına rağmen aslında kesin bir “köşe” kavramı tanımı yapılamamıştır. Köşe; iki doğrunun kesiştiği nokta olarak veya yerel bir komşulukta iki farklı baskın kenarın kesişimi olarak tanımlanabilir. İlgi noktası ise görüntüde konum bakımından yeri doğru olarak tanımlanmış ve sabit bir şekilde beliren, nesnenin özelliğini gösteren bir noktadır. İlgi noktası bir köşe ya da bir eğrinin zirve noktası, bir çizginin sonu veya yerel yoğunluğu maksimum ya da minimum olan herhangi bir nokta da olabilir [20]. Bu sebeple bulunan ilgi noktası gerçekte bir köşe olabilir ya da olmayabilir.

Köşe, iki doğrunun kesiştiği nokta olmakla birlikte, görüntü içindeki köşeler; şekillerin kesiştiği noktalar, nesne sınırlarının kesilerek yön değiştirdiği noktalar veya kenar sınır eğrisinin belirli bir eşik değeri üzerindeki değerleri köşe nokta olarak tanımlanabilir [5]. Örneğin bir nesnenin kenar çizgilerinin kesiştiği noktaların yanı sıra, iki nesnenin bir düzlemde kesiştiği doğrunun bitiş noktaları da köşe olarak belirlenebilir.

Köşeler görüntü içindeki nesnelere önemli birer özelliğidir ve nesnenin tanımlanması, ayırt edilebilmesi ve yerinin belirlenebilmesi için yararlı olan bilgiyi sağlar [7]. İnsan gözü şekilleri ayırt edebilmek için köşelerden faydalanır. Nesne tanımanın yanı sıra, hareket izleme, stereo eşleştirme gibi köşeleri kullanan yöntemler için gerekli olan işlemlerden biri olmasından dolayı çok sayıda köşe bulma yöntemi geliştirilmiştir [1-15].

Köşeler, görüntü içinde farklı şekillerde bulunabilirler. Kenarların birleşimleri, iki veya daha fazla parçanın buluşması, kesişmesi veya çakışması ile oluşabilirler. Şekil 1.1’de görüntülerde oluşabilecek çeşitli köşeler gösterilmiştir. İki doğru parçasının buluşması veya iki boyutlu dörtgensel geometrik şekillerin kenarları L-birleşim bir köşe oluşturur. Y-birleşim daha çok 3-boyutlu dikdörtgensel nesnelerin 3 farklı yüzünün oluşturduğu perspektif görüntüde bulunur. Ok-birleşim yine 3 boyutlu dikdörtgensel cisimlerin alt tabanı ve yan yüzlerinin birleşerek oluşturduğu köşelerdir. T-birleşim 3 boyutlu bir nesneyle 2 boyutlu bir şeklin birleşimiyle, X-birleşim ise iki doğrunun veya nesne kenarlarının birbirini kesmesiyle oluşur. Çoklu-birleşimde 3 veya daha fazla kenarın çakışmasıyla oluşabilir.



Şekil 1.1 Farklı şekillerdeki köşe çeşitleri- X-Birleşim, Y-Birleşim, T-Birleşim, L-Birleşim, Çoklu-Birleşim, Ok-Birleşim

Görüntülerde farklı şekillerde köşeler oluştuğu için standart bir köşe bulma yöntemi yoktur. Bu nedenle geliştirilen bir köşe bulma algoritmasının bütün köşe çeşitleri üzerinde doğru sonuçlar vermesi çoğu zaman sıkıntılıdır [14]. İyi bir köşe bulma algoritmasının aşağıdaki şartları sağlaması gerekir [29]:

- Bütün köşeler doğru olarak tespit edilmelidir.
- Köşe olmayan noktalar köşe olarak tespit edilmemelidir.
- Köşeler tam olarak ayırt edilebilmelidir.
- Köşe bulma algoritması gürültülü resimlerde bile doğru sonuçlar vermelidir.

- Bütün bu koşulları sağlarken hızlı ve verimli bir şekilde çalışmalıdır.

Literatürde köşe bulmak için farklı yaklaşımlar kullanan çok sayıda yöntem bulunmaktadır. Gri-seviye görüntü içerisinden köşelerin sezimini gerçekleştiren yöntemler, kenar bulma tabanlı yöntemler, yoğunluk tabanlı yöntemler olarak iki gruba ayrılabilir [20]. Köşe seziminde daha çok kenar tabanlı yöntemler kullanılmasının yanı sıra, gri seviye görüntülerde morfolojik açma (opening) ve kapama (closing) işlemlerini kullanan az sayıda köşe bulma algoritması bulunmaktadır. Bu çalışmada morfoloji kullanan algoritmalarından biri olan topolojik bir köşe bulma [4] algoritmasının iyileştirilmesi ve diğer köşe bulma yöntemlerine göre performansının değerlendirilmesi gerçekleştirilmiştir.

Bu tez çalışmasında, karşılaştırmada kullanılacak olan çeşitli köşe bulma algoritmaları ikinci bölümde incelenmiştir. Üçüncü bölümde kullanılan yöntemin temel tanımlamaları açıklanmış, dördüncü bölümde ise bu tanımlamalara dayanarak sunulan filtrelere yer verilmiş ve kullanılan “Bulanık Topolojiye Dayalı Köşe Bulma” algoritması açıklanmıştır. Beşinci bölümde ise kullanılan algortmada yapılan değişiklikler ve görüntüler üzerindeki uygulama sonuçları yer almaktadır. Beşinci ve son bölüm bu algoritma hakkındaki sonuç fikri vermektedir.

1.1 Kenar Bulma Tabanlı Yöntemler

Kenar bulma tabanlı yöntemler, köşeleri belirleyebilmek için öncelikle nesnelerin sınırlarını belirleyen kenarları, çeşitli kenar bulma yöntemlerini uygulayarak bulmaktadır. Kenarlar belirlendikten sonra kenar çizgilerinin kesiştiği noktalar bulunarak veya eğriliğin maksimum olduğu noktalar belirlenerek köşelerin yerleri tespit edilmektedir.

Kenar bulmaya dayanan köşe bulma yöntemleri köşeleri belirleyebilmek için yöntemi kenarları belirlenmiş bir görüntü üzerinde uygulamaktadırlar. Bu yöntemlerin en önemli dezavantajı başlangıç olarak kenar bilgisine ihtiyaç duymaları sebebiyle kenar bulma aşamasında oluşabilecek sapmalardan dolayı köşelerde kayba neden olabilirler. Kenar belirlemede ortaya çıkan hatalar veya

sapmalar da köşeleri belirlemede hataya sebep olacaktır, bu bakımdan yeterince güvenilir değildirler.

1.2 Yoğunluk Tabanlı Yöntemler

Köşe bulma yöntemlerinin bir kısmı da gri-seviye görüntü üzerinde köşe belirleme yöntemini kullanmaktadır. Bu yöntemler genellikle renkli veya gri seviye görüntülerde birinci veya ikinci dereceden türeve dayalı olarak büyüklük ve yön hesaplamasına dayanır. Ayrıca türevden farklı olarak gri seviye görüntüler üzerinde morfolojik işlemler uygulayarak köşe belirleyen yöntemler de vardır [2,3]. Kenar bulma tabanlı yöntemler gibi daha önce başka bir işleme gerek duymaksızın köşeleri belirledikleri için güvenilirlikleri daha yüksektir. Çünkü doğrudan görüntünün gri seviye yoğunluk değerleri üzerine uygulanırlar ve ilk işlem olarak başka bir yöntem uygulanmadığı için ondan kaynaklanabilecek hatalar oluşması söz konusu değildir. Bu tezde incelenen yöntemlerin birçoğu gri seviye görüntülerde yoğunluk değerlerini kullanarak köşe tespitini gerçekleştirmektedir.

2. KÖŞE BULMA YÖNTEMLERİ

Literatürde çok sayıda köşe bulma yöntemi bulunmaktadır [1-15]. Ancak bunların birçoğu daha önce var olan bir yöntemin geliştirilmesine dayanır. Bu tezde birçok köşe bulma yöntemine temel olan ve genellikle çok iyi sonuçlar üreten köşe bulma yöntemleri incelenmiş ve bu yöntemler ile iyileştirilen yöntemin sonuçları karşılaştırılmıştır.

2.1 Kitchen ve Rosenfeld Yöntemi

Kitchen ve Rosenfeld [1] 1982 yılında sundukları çalışmada gri-seviye şemaları kullanarak kenar bulma tabanlı teknikleri daha da iyileştirmişlerdir. Kenar sınırı boyunca gradyan büyüklüğünü ve gradyan yönünün değişim oranını kullanarak gerçekleştirilen bir kenar bulma algoritması sunmuşlardır. Kullanılan teknikte, yerel bir işleç gri seviye görüntünün komşuluklarına paralel olarak uygulanır. Her komşulukta işleç, o komşuluğun içinden geçen bir kenarın eğimini hesaplar. Ayrıca kenarın yerel büyüklüğünü de hesaba katmak gerekir, çünkü yüksek değerler bir şeklin kenarını gösterir. İşlecin uygulanması sonucunda sadece belirgin bir köşeden dönen keskin kenarlar yüksek değerler alır. Gradyan yönünü hesaplamak için Denklem (2.1) kullanılır. Gradyanın x ve y bileşenlerinin hesaplanması için, yerel gradyan değeri, yatay ve dikey Prewitt [28] işlecinin uygulanmasıyla hesaplanır (Denklem (2.2) ve (2.3)).

$$\tan(\theta(x, y)) = \frac{I_y(x, y)}{I_x(x, y)} \quad (2.1)$$

$$\theta_x = \frac{I_{xy}I_y - I_{xx}I_y}{I_x^2 + I_y^2} \quad (2.2)$$

$$\theta_y = \frac{I_{yy}I_x - I_{xy}I_y}{I_x^2 + I_y^2} \quad (2.3)$$

Kitchen ve Rosenfeld köşe bulmak için birkaç farklı yöntem kullanmışlardır. İlk olarak gradyan yönleri ve gradyan büyüklükleri hesaplama işlemi yapılmıştır. Gradyan yönleri -180 ile +180 derece arasında alınıp, onların mutlak değerleri gri-seviye bir resim olarak oluşturulursa, bu yön resmindeki

parlaklık deęişimleri orijinal resimdeki kenarların yön deęişimlerini gösterir. Parlaklık deęişimleri yön resminin gradyan büyüklüęü hesaplanarak bulunabilir. Gerçek köşelilik (cornerness) ölçüsünü hesaplayabilmek için bu sonucun orijinal resmin gradyan büyüklüęü ile çarpılması önemlidir [1].

$$K = \frac{I_x \theta_y - I_y \theta_x}{I_x^2 + I_y^2} = \frac{I_{xx} I_y^2 + I_{yy} I_x^2 - 2I_{xy} I_x I_y}{I_x^2 + I_y^2} \quad (2.4)$$

İkinci olarak kenarlar boyunca yön deęişimi hesaplanmıştır. Sadece kenar boyunca yön deęişimini ölçmek, nesne sınırlarındaki ani dönüşlere büyük ölçüde benzer. Orijinal görüntüde gradyan yönü hesaplandıktan sonra, sonuç görüntüye 3x3'lük bir işleç uygulanarak bu yöntem gerçekleştirilebilir. Bu işleç her komşuluk içinde merkezden geçen ve merkez pikseldeki gradyan yönüne dik olan çizgiye yakın yerleşen karşılıklı pikselleri inceler. İşlemin sonucunda incelenen karşılıklı iki pikselin gradyan yönleri farkı bulunur. Eğer kayda değer bir fark bulunursa bu eğrinin yönünden ve büyüklüğünden bilgi elde etmek mümkündür. Yine bu eğrinin değeri de gradyan büyüklüęü ile çarpılmalıdır. Bir önceki yönteme nazaran bu yöntem daha iyi sonuç vermektedir.

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 tondadırlar. Kenarın dışında kalan pikseller ya daha açık, ya da daha koyu olabilirler. 3x3'lük bir komşulukta merkez piksele en yakın gri tondadaki 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 yöntemde yerel komşuluğun merkezinde hesaplanan gradyan yönünün deęişim oranı başka bir şekilde türetilmiştir. Kare bir komşuluk merkezinden bir kenarın kontur çizgisinin geçtięi düşünülerek Denklem (2.5) uygulanır.

$$I(x, y) = I(x, y) - I(0, 0) = 0 \quad (2.5)$$

Denklemden y , x 'in bir fonksiyonu olarak düzenlenir ve x 'e göre y 'nin birinci ve ikinci türevleri hesaplanırsa, düzlemdeki eğrilik Denklem (2.6) ile elde edilir ve gradyan büyüklüęü ile çarpılarak tekrar k köşelilik değeri elde edilmiş olur.

$$\frac{d^2y/d^2x}{(1+(dy/dx)^2)^{3/2}} \quad (2.6)$$

Beşinci yöntemde Beaudet'in [24] DET işleci olarak isimlendirilen, işleci bir yüzeydeki köşeleri ve sırt noktalarını (saddle point) belirler. Bu işlecin 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 özellikle keskin kenarlı şekiller üzerinde kötü sonuçlar vermesidir.

Yukarıda açıklanan bütün köşe bulma yöntemleri büyük ölçekli komşuluklarda sapma gösterebilir. Köşelerin konumlarının daha iyi belirlenmesi, yöntemlerden alınan sonuçlar üzerinde piksellerin değişik boyuttaki işlemler ile çarpımı veya sonuçlara yerel 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 işleci bu kenardan etkilenir ve köşeyle birlikte kenara da tepki verir. Bu problemde kenar gradyan büyüklüklerini çarpımda kullanmadan önce maksimum olmayanların bastırılması yöntemi uygulanarak çözülebilir.

Sunulan yöntemleri geleneksel bir köşe bulma yöntemi 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.

Sonuç olarak geleneksel köşe bulma işlemleriyle sunulan yöntemler karşılaştırıldığında geleneksel yöntem görüntüdeki şekilleri ayırdıktan sonra köşe bulmaktayken, sunulan yöntemlerde görüntü içindeki şekiller ayırt edilmeden, köşeler bulunmaktadır. Uygulanan eşik değerine göre iki yöntem benzer sonuçlar verebilir. Ayrıca işlem hızı bakımından, geleneksel yöntemler sıralı işlemler

yaptığı için daha kısa sürede cevap vermektedir. Sunulan yöntemlerde paralel programlama uygulanırsa, işlem zamanı farkı ortadan kaldırılabılır [1].

2.2 Harris Yöntemi: Plessey Operatörü

Harris ve Stephens [8] 1988 yılında videodan alınan görüntü kesitleri içinde nesne hareketlerini izleyebilmek için nesne köşelerini belirleyebilmek amacıyla bir köşe bulma işleci sunmuşlardır. Literatürde Plessey adıyla bilinen bu köşe bulma yöntemi ilk olarak Moravec [25] tarafından sunulan bir başka köşe bulma işlecinin eksikliklerini belirleyip, geliştirilerek ortaya konmuştur. Sonuçta geniş bir kullanım alanı bulan, Moravec işlecine göre daha az hesaplama karmaşıklığı içeren ve doğruluk oranı yüksek bir yöntem geliştirilmiştir.

Moravec işleci sadece sekiz yönde yoğunluk değişimini hesapladığı için anisotropik bir sonuç vermektedir. Yöntemin bu yetersizliğini gidermek için yoğunluk değişiminin mümkün olabilen her yönde hesaplanabilmesi gerekmektedir. Plessey işleci belirli bir pencere matrisinin merkezindeki pikselin mümkün olan tüm yönlerde yoğunluk değişimini hesaplar ve bu yöntemin bir otokorelasyon matrisi olarak ifade edilmesinin mümkün olduğunu gösterir. Bu matris, $I(x,y)$ görüntüsünün gri seviye yoğunluk fonksiyonunda birinci dereceden türevi alınarak hesaplanmıştır.

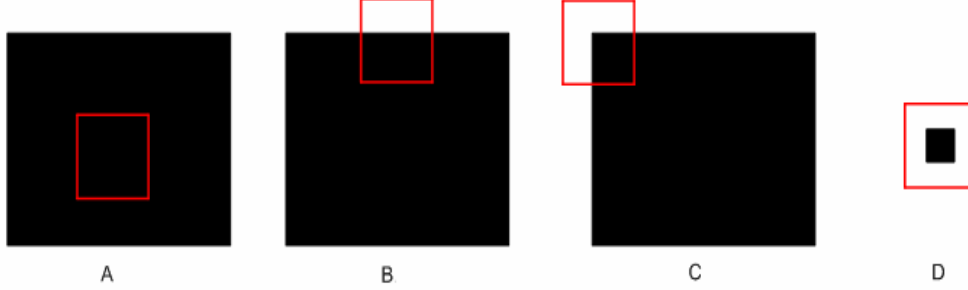
Otokorelasyon matrisi M:

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right) \left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right) \left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \quad (2.7)$$

M matrisinin özdeğerleri yüzey eğrilik değerleri ile orantılıdır ve M'yi yönden bağımsız olarak ifade eder. Oysaki M matrisinin bileşenleri sadece yatay ve dikey gradyanları hesaplayabildiği için yönden bağımsız değildir.

Belirli bir gözlem penceresinin görüntü üzerinde kaydırılması sonucu dört farklı durum ortaya çıkar. Bu durumlar Şekil 2.1'de gösterilmiştir. A durumunda gözlem penceresi görüntüde bir nesnenin iç bölgesinde veya arka plan üzerinde yer almaktadır ve gözlem penceresi içindeki görüntünün yoğunluk değişimi

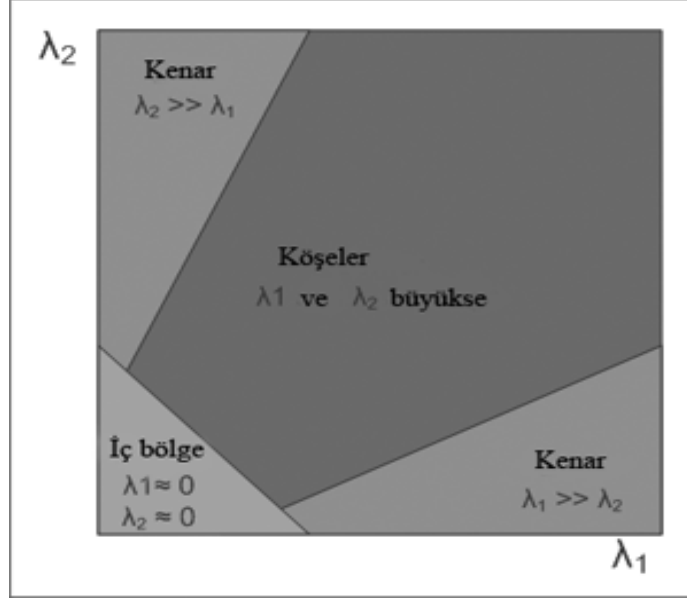
nispeten sabittir. Pencere içinde hesaplanan yüzey eğrilik miktarı çok fazla değişim göstermemektedir. Ayrıca hesaplanan özdeğerler çok yüksek değerlere sahip değildir.



Şekil 2.1 Plessey operatörünün karşılaştığı farklı durumlar

B durumunda gözlem penceresi bir kenar üzerinde yer almaktadır. Bu durumda pencere içindeki yüzey eğrilik miktarı kenar boyunca fazla değişim göstermezken kenara dik olarak alındığında eğimi daha büyük değerler almaktadır. Bu iki durumdan birinde hesaplanan özdeğer diğerinden büyük olur. C ve D durumlarında ise pencere bir köşeyi veya bağlantısız bir pikseli içine almaktadır. Bu durumda hem yüzey eğrilik miktarı hem de özdeğerlerin değeri her yönde artış göstermektedir.

M matrisinden elde edilen öz değerler λ_1 ve λ_2 olsun. Hesaplanan özdeğerlerden biri diğerinden değer olarak büyük bir farka sahipse bu durumda gözlem penceresi bir kenarı içine almaktadır. Her özdeğerlerin değeri oldukça küçük ve sıfıra yakın ise gözlem penceresi fazla yoğunluk geçişleri olmayan bir arkaplan veya zemin üzerindedir. Eğer özdeğerler belirli bir eşik değerinden oldukça büyük ise gözlem penceresi bir köşeyi içine almaktadır. Bu durumlara göre λ_1 ve λ_2 değerleri özdeğer uzayını Şekil 2.2 de görüldüğü gibi 3 bölgeye ayırır.



Şekil 2.2 Özdeğer uzayının farklı bölgelere ayrılması

Hesaplanan özdeğerlerin bu bölgelerden hangisinde bulunduğu göz önüne alınarak köşeler tespit edilir. Her bir piksele ayrı etiketler atamak yerine bir köşelilik değeri hesaplanır ve köşelerin veya kenarların yerel maksimumu alınarak yerlerinin doğru olarak tespiti gerçekleştirilmiş olur. Harris ve Stephens [8] tarafından önerilen köşelilik fonksiyonu:

$$R = \det M - k (\text{trace } M)^2 \quad (2.8)$$

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

$$k = \text{sabit}$$

Harris ve Stephens k parametresinin uygulama testlerinde en doğru sonuçlar verdiği değer aralığını 0.04 ile 0.06 olarak önermiştir. Köşelilik fonksiyonu sonucunda negatif değerlerin kenarları, pozitif değerlerin ise köşe ve iç bölgeleri gösterdiği görülmektedir. Köşe ve iç bölgeleri birbirinden ayırt etmek ve köşeleri tespit edebilmek için belirli bir değere göre eşikleme yapılması gerekir, çünkü belirli bir değerden daha küçük köşelilik değerleri iç bölgelere karşılık gelmektedir. Bu yüzden eşik değeri köşe olmayan noktaları köşe olarak belirlemeyecek kadar büyük bir değer olmalıdır. Köşeler, köşelilik fonksiyonunun yerel maksimumu olarak belirlenir. Köşelerin gürültüden etkilenmemesi için

görüntüyü Gaussian filtresi kullanarak düzgünleştirmek mümkündür. Bu filtre doğrudan görüntünün kendi üzerinde değil de, görüntüde türevinin karesini içeren

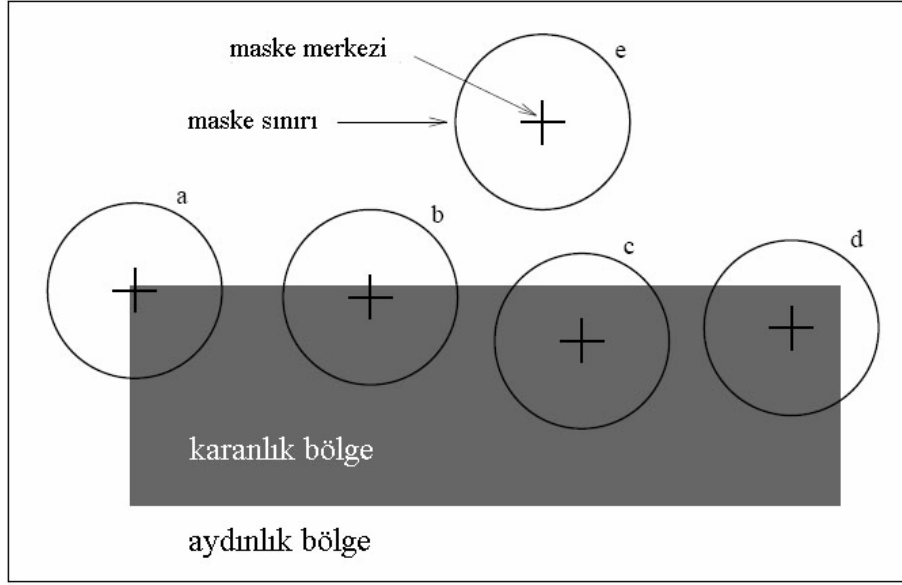
fonksiyonlara uygulanır $\left(\left(\frac{\partial I}{\partial x} \right)^2, \left(\frac{\partial I}{\partial y} \right)^2, \left(\frac{\partial I}{\partial x} \right) \left(\frac{\partial I}{\partial y} \right) \right)$.

Plessey işleci genellikle doğru köşeleri bulan en iyi işleç olarak kabul edilir, ancak işlem yoğunluğu fazla ve köşelerinin yerlerinin doğru tespiti konusunda eksiklikleri bulunmaktadır [23].

2.3 SUSAN Yöntemi

Smith ve Brady [11], dairesel bir maske içinde parlaklık karşılaştırmasına dayanan yeni bir köşe bulma algoritması sunmuştur. SUSAN (Smallest Univalve Segment Assimilating Nucleus), dairesel bir maske içindeki pikseller eğer bir nesneye ait iseler nispeten aynı tarzda parlak olduklarını varsayar. Algoritma dairesel maskenin merkezinde yer alan piksel ile benzer parlaklığa sahip piksellerin sayısını hesaplar ve maskenin ağırlık merkezini bulur. Benzer parlaklıktaki bu pikseller USAN (Univalve Segment Assimilating Nucleus) olarak isimlendirilir.

Şekil 2.3'de USAN'ın görüntüde oluşabileceği durumlar gösterilmektedir. Maskenin merkezi yoğunluk farkı olmayan düzgün bir görüntü bölümü üzerinde yer aldığıda USAN en geniş alana ulaşır. Merkez bir kenar çizgisine yakın yer aldığıda USAN dairesel maskenin yaklaşık olarak yarısını kaplar ve USAN'ın alanı maskenin yarısından daha az yer kapladığıda ise merkez piksel bir köşe yakınlarında bulunur. USAN'ın büyüklüğü, içindeki piksel sayısı, ikinci momenti hesaplanarak kenarlar ve köşeler bulunabilir. USAN haritasının yerel minimumu görüntünün köşe noktalarını gösterir [11].



Şekil 2.3 Dairesel maskeler kullanarak USAN yönteminin bir görüntüye uygulanması

Görüntünün bütün pikselleri üzerinde dairesel maske uygulanarak merkez piksel ile maske içindeki pikseller karşılaştırılır ve benzer yoğunluktaki pikseller bulunarak USAN haritası oluşturulur:

$$C(r, r_0) = \begin{cases} 1, & |I(r) - I(r_0)| \leq t \\ 0, & |I(r) - I(r_0)| > t \end{cases} \quad (2.9)$$

USAN haritası Denklem (2.9) uygulanarak bulunur. r_0 Maskenin merkez pikseli, r ise maske içindeki diğer pikselleri ve $I(r)$ piksellerin yoğunluk değerini gösterir. USAN'ın hesaplanabilmesi için maske içindeki piksellerin gri seviye yoğunluk farkı üzerine belirli bir t değerine göre eşikleme yapılır. Karşılaştırma sonucunda $C(r, r_0)$ çıktısı yani USAN haritası oluşur. Bu işlem tüm pikseller için uygulanır ve her maskede USAN içinde yer alan piksellerin sayısı hesaplanır:

$$n(r_0) = \sum_{\bar{r}} c(r, r_0) \quad (2.10)$$

n , USAN haritası içinde yer alan piksellerin toplam sayısını gösterir ve aynı zamanda USAN'ın kapladığı alanı belirtmektedir. t eşik değerinin uygulanması sayesinde görüntüde maksimum gürültü etkisi bertaraf edilir, minimum karışıklık ortaya çıkar.

Bir sonraki aşamada n değeri geometrik eşik değeri g ile karşılaştırılır. Bu değer başlangıç olarak kenar bulmak için $3n_{max}/4$ olarak belirlenir. Burada n_{max} n 'nin alabileceği en büyük değerdir.

$$R(r_0) = \begin{cases} g - n(r_0) & \text{eğer } n(r_0) < g \\ 0 & \text{değilse} \end{cases} \quad (2.11)$$

SUSAN hem kenar hem de köşe bulmak için geliştirilmiş bir görüntü işleme yöntemidir. Görüntülerde köşe ve kenarlar gibi görüntü özelliklerini ortaya çıkaran formül, Denklem (2.11)'de gösterilmiştir. Bu formül SUSAN yönteminin temel prensibini basit olarak açıklamaktadır. Formülde g eşik değerine başlangıç değerden daha küçük değerler atayarak köşeleri de tespit etmek mümkündür. Köşe tespitinde, eğer maskenin merkez noktası bir köşenin üzerine denk geliyorsa USAN'ın kapladığı alan maske alanının yarısından daha küçük olur. Bu yüzden n değeri n_{max} değerinin yarısından küçük olmalıdır. Ayrıca USAN alanı küçüldükçe köşe tepkisi de artmaktadır [11].

Doğru sonuçlar elde edebilmek için kullanılan t ve g eşik değerlerinin görüntülere uygun olarak seçilmesi gerekir, uygun olmayan seçimler sonuçları büyük ölçüde etkiler. t eşik değeri fazla küçük seçilirse, çok sayıda aday köşe belirlenir, bunlar içinde gerçek köşelerin tespit edilmesi gerekir. t değeri fazla büyük belirlenirse bazı gerçek köşeler köşe olarak belirlenmeyebilir. g eşik değerinin de n_{max} değerinin yarısı olarak belirlenmesi daha doğru sonuçlar verecektir.

Köşe tespitinde köşe olmayan yerlerin köşe olarak bulunmasını önlemek için USAN'ın ağırlık merkezi bulunur. Eğer maske merkezi köşe yakınlarında ise USAN'ın ağırlık merkezi ve maske merkezine uzaktır. Ayrıca USAN içinde bulunan köşelerin doğruluğundan emin olmak için maske merkezinden USAN'ın ağırlık merkezine doğru olan doğruların USAN haritası içinde olup olmadığına bakılır. Eklenen bu işlemler yanlış köşeler bulmayı önleyecektir. Son aşamada USAN haritası içinde yerel minimum (non-maximum suppression) hesaplanarak 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 bu bakımdan daha hızlı uygulanabilen bir

yöntemdir. Bu özellikleri ile diğer köşe bulma yöntemlerine üstünlük sağlamaktadır.

2.4 Curvature Scale Space (CSS) Yöntemi

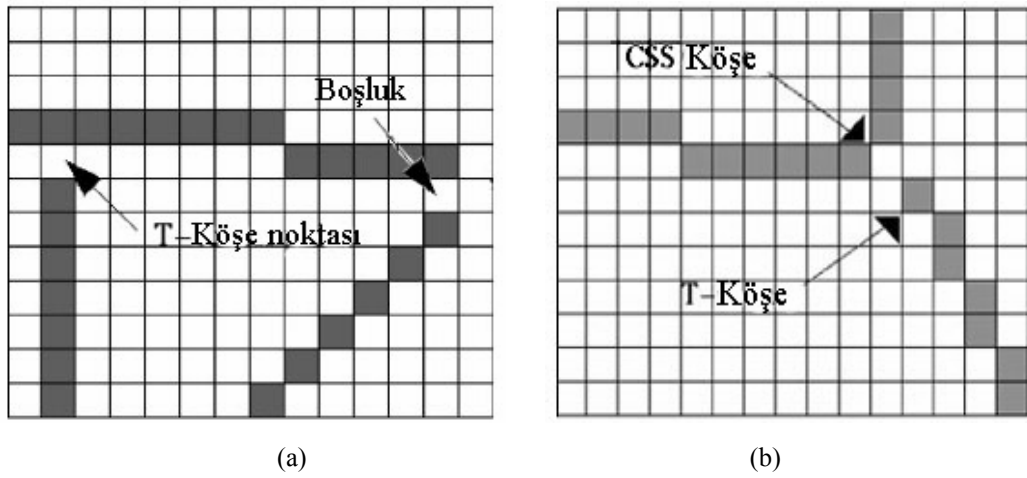
Mokhtarian ve Suomela [10] tarafından Curvature Scale Space (CSS) tekniğini kullanılarak köşeleri tespit etmeyi sağlayan bir köşe bulma yöntemi sunulmuştur. Bu yöntem köşelerin “mutlak eğriliğin yerel maksimumu” olarak tanımlanmasına dayanmaktadır. CSS tekniği farklı ölçeklerdeki konturların (contour) eğrilik özelliklerini ortaya çıkarmak için uygun bir yöntemdir. Bu köşe bulma yöntemi kenar kontur çizgileri ortaya çıkarılmış bir görüntü üzerinde uygulanır.

CSS yönteminin uygulama adımları şöyle sıralanmıştır [7]:

- Canny [26] veya başka bir kenar bulma dedektörü kullanarak görüntüdeki kenar konturlarının belirlenmesi.
- Kenar konturları üzerinde boşluklar varsa bu boşlukların doldurulması.
- Boşluklar doldurulduğunda T-birleşim oluşturan yerlerin T-köşe olarak işaretlenmesi.
- Yüksek bir ölçekte kenar konturları üzerinde eğriliğin hesaplanması
- Belirli bir eşik değeri üzerindeki mutlak eğriliğin en yüksek olduğu noktaların köşe olarak belirlenmesi
- Köşe olarak belirlenen noktaların pozisyonlarının doğru olup olmadığının kontrol edilmesi ve yerinin doğru tespit edilmesi.
- CSS tekniği kullanılarak bulunan köşelerin, T-köşeler ile karşılaştırılması ve birbirine çok yakın köşelerin teke indirilmesi.

CSS operatörü uygulamada kenar özelliklerini kullandığı için, öncelikle kenar konturlarının ortaya çıkarılması gerekmektedir. Bu amaçla CSS köşe bulma yöntemi kenar belirlemek için Canny [26] kenar işlecini kullanmaktadır. Kenar belirlemede CSS operatöründe farklı kenar bulma işleçleri de kullanılabilir. Canny kenar işlecinin daha iyi sonuçlar verdiği gözlemlendiği için bu kenar bulma yöntemi tercih edilmiştir [10].

Kenar konturların belirlenmesi sırasında görüntüdeki nesnelerin oluşturduğu T-birleşim noktalarda boşluklar oluşabilmektedir. Kesikli kenar konturları köşelerin belirlenmesine engel olduğu için bu boşlukların doldurulması gerekir. Bir kenar çizgisinin bittiği nokta, başka bir kenar çizgisinin bittiği noktaya ya da bir kenar çizgisine çok yakınsa aradaki boşluk doldurularak birleştirilir. Şekil 2.4 (a)'da kenar konturlarında oluşabilen bu iki tipteki boşluklar gösterilmiştir. Boşluklar birleştirildiğinde T-birleşim oluşturan noktalar T-köşe olarak işaretlenir.



Şekil 2.4 CSS'nin karşılaştığı durumlar (a) Kenar konturları üzerinde yer alan iki farklı boşluk türü. (b) Bir köşenin birden fazla kez köşe olarak belirlenmesi .

Görüntüde yer alan kenar konturlar oluşturulduktan sonraki aşamada kenar çizgilerinin mutlak eğrilik değeri başlangıç olarak yüksek bir ölçekte hesaplanır. Mutlak eğrilik değerinin yerel maksimum olduğu noktalar aday köşeler olarak belirlenir. Mutlak eğriliğin maksimum olduğu noktalar, gerçek köşeler ya da gürültünün etkisiyle ortaya çıkan tepe noktaları olabilir. Gerçekte gürültüye bağlı olarak ortaya çıkan tepe noktaları gerçek birer köşe değildirler. Gerçek köşelerin mutlak eğrilik değeri gürültünün sebep olduğu tepe noktasına göre daha yüksektir. Aday köşeleri ve gerçek köşeleri birbirinden ayırt edebilmek için aday köşeler komşuluğundaki yerel minimum değer ile karşılaştırılır. Ayrıca gerçek köşelerin eğimi ile gürültünün sebep olduğu tepe noktalarının eğrilik değeri arasında açık bir fark bulunduğu için belirli bir t değerine göre eşikleme yapılır.

Başlangıçta aday köşeler belirlendikten sonra bu köşeler tekrar ele alınarak yeniden gözden geçirilir. Köşeler başlangıçta yüksek bir ölçekle belirlendiği için pozisyonları doğru tespit edilmemiş olabilir. Eğriliği daha düşük ölçeklerde hesaplayarak yakın komşulukta bulunan aday köşeler ile önceki hesaplamada bulunan köşeler karşılaştırılır, gerekirse köşelerin yerleri güncellenir. Bu izleme ölçek en düşük seviyeye varıncaya kadar yapılır. İzleme sırasında köşelerin sayısı değişmez yalnızca yerlerinin daha doğru belirlenmesi sağlanır.

Bu yöntemde köşeler iki ayrı şekilde belirlenmiştir. Birincisi boşluk doldurma esnasında işaretlenen T-köşeler, ikincisi mutlak eğrilik hesabı ile bulunan köşelerdir. Son aşamada birbirine çok yakın yerlerde bulunan iki kez işaretlenen ve aslında aynı köşeyi gösteren aday köşelerin teke indirilmesi işlemi yapılır. Şekil 2.4(b)'de bir köşe hem T-köşe olarak işaretlenmiş hem de CSS ile bulunmuştur.

CSS tekniği kullanan köşe bulma yönteminde mutlak eğriliğin yüksek bir ölçekte hesaplanması görüntüdeki gürültü etkisini azaltmakta ancak gerçek köşelerin bulunma olasılığını düşürmektedir. Eğriliğin çok düşük bir ölçekte hesaplanması ise gürültüye bağlı olarak oluşan tepe noktaları köşe olarak belirlemektedir. Ölçeğin yüksekten alçağa doğru düşürülerek hesaplanması yalnızca ilk aşamada tespit edilen köşelerin yerlerini daha doğru belirleyebilmek için yapılmaktadır. Karmaşık bir görüntüde bu yöntem kullanılarak köşeler belirlenmek istendiğinde gerçek köşeleri belirleyemediği ancak köşe olmayan noktaları köşe olarak belirlediği görülmektedir [7]. Ayrıca her görüntüye genel bir t eşik değeri uygulanması da beklenmeyen sonuçlar üretebilmektedir. Bu eksikliklerin giderilmesi için CSS köşe bulma yöntemi Mokhtarian ve Mohanna [27] tarafından ve ayrıca uyarlamalı eşik değeri fikrine dayanarak He ve Yung [7] tarafından geliştirilmiştir.

2.5 Laganiere Yöntemi

Laganiere 1998’de morfolojik işlemlere dayalı “Asimetrik Kapatma” (Asymmetric Closing) [2] adını verdiği bir köşe bulma algoritması sunmuştur. Bu yöntemde artı, çarpı, eşkenar dörtgen ve kare şeklinde dört farklı yapı elemanı kullanarak morfolojik aşınma-genleşme ve açma-kapama yöntemlerini uygulamıştır.

Küme teorisine dayanan matematiksel morfoloji, nesne şekillerini değişik yapı elemanlarıyla modelleyerek bulur. Temel morfolojik işlemler aşınma, genleşme, açma ve kapama işlemleridir. Bir görüntü kümesi A’yı, bir yapı elemanı B ile morfolojik genleşme işlemi şöyle tanımlanır:

$$A \oplus B = \{ a + b \mid \exists a \in A \text{ ve } b \in B \} \quad (2.12)$$

A görüntü kümesinin, B yapı elemanı ile morfolojik aşınma işlemi:

$$A \ominus B = \{ a - b \mid \exists a \in A \text{ ve } b \in B \} \quad (2.13)$$

olarak tanımlanmıştır. Bileşik morfolojik işlemler olan açma ve kapama işlemleri şunlardır:

Morfolojik açma işlemi:

$$A \circ B = (A \ominus B) \oplus B \quad (2.14)$$

Morfolojik kapama işlemi:

$$A \bullet B = (A \oplus B) \ominus B \quad (2.15)$$

Asimetrik kapama yönteminde Laganiere, önce bir yapı elemanı ile genleşme işlemi uygulamış, daha sonra başka bir yapı elemanı ile aşınma işlemi uygulamıştır. Bu iki işlemin birbirini tamamlayıcı biçimde kullanılmasının sebebi, farklı tipteki köşeleri yakalayabilmektir. Bu işlemde kullanılan yapı elemanları, artı “+” ve “◇” eşkenar dörtgendir (Şekil 2.5). A görüntüsünün “+” ve “◇” yapı elemanları ile asimetrik kapama işlemi Denklem (2.16)’daki gibidir:

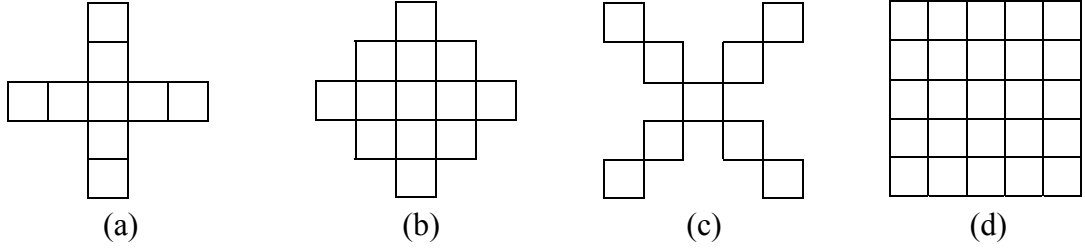
$$A_{+, \diamond}^c = (A \oplus +) \ominus \diamond \quad (2.16)$$

Görüntüdeki köşelerin bir bölümünü aşağıdaki Denklem (2.17) ile bulmak mümkündür:

$$C_+(A) = | A - A_{+, \diamond}^c | \quad (2.17)$$

Dört farklı yöndeki köşeleri bulmak için kullanılan yapı elemanları Şekil 2.5’de gösterilmiştir. “+” ve “◊” yapı elemanlarının 45° açı ile döndürülmüş hali olan “×” ve “□” yapı elemanları kullanılarak yapılan asimetrik kapama işlemi uygulanmıştır. Denklem (2.18) ile diğer iki yöndeki köşelerde bulunur.

$$C_{\times}(A) = |A - A_{\times, \square}^c| \quad (2.18)$$



Şekil 2.5 Laganieri’in kullandığı 4 farklı yapı elemanı (a) +, (b) ◊ , (c) × ve (d) □.

Tanımlanan bu iki farklı köşe dedektörü farklı yönlerdeki köşeleri ayrı ayrı bulabilmek için neredeyse yeterlidir. Sonuç olarak bu iki dedektörün birleştirilmesiyle 4 farklı yönde köşelerin bulunmasını ve çok küçük boyutlu şekillerin yok edilmemesi sağlanır. İki dedektör Denklem (2.19)’daki gibi birleştirilmiştir.

$$C_{+, \times}(A) = |A_{+, \diamond}^c - A_{\times, \square}^c| \quad (2.19)$$

Laganieri’nin köşe işleci geniş açılı köşeleri bulamayabildiği ve keskin açılı köşelerin piksellerini silebildiği için, Shih ve ark. [3], yöntemin eksikliklerini gidermek ve daha iyileştirmek üzere Laganieri’in yöntemini yeniden düzenlemişlerdir. Bu düzenlenmiş yöntemde iki dedektörün birleşim işlemini Denklem (2.19) yerine aşağıdaki gibi tanımlanmıştır:

$$C_{+, \times}(A) = |A - A_{+, \diamond}^c| \cup |A - A_{\times, \square}^c| \quad (2.20)$$

Bu işlemin üzerine bir de katılık (strictness) değeri eklenerek eski yöntemin eksik olduğu yönlerin giderilmesi amaçlanmıştır. Katılık değerinin uygulanması ile görüntülerde gürültüye olan duyarlılık ve nesne kenarlarında oluşan girinti ve çıkıntılar azaltılmıştır. Eğer katılık değeri 1 olarak alınırsa, Shih yöntemi Laganieri gibi çalışır. Belirli katılık değeri aralığında bu yöntem Laganieri göre daha iyi sonuçlar vermektedir [3].

3. DİJİTAL TOPOLOJİ ve BULANIK DİJİTAL TOPOLOJİ

Genel topoloji, rastgele seçilen herhangi bir noktanın komşuluğundaki, sınırsız sayıda noktalardan veya uzay elemanlarından oluşan kümeleri inceleyen bir bilim dalıdır. Genel topoloji süreklilik, kenar ve sınır gibi kavramlarla ilgilidir, bu yüzden sınırsız sayıda noktanın, komşuluğundaki noktalarla olan ilişkisini inceler [21].

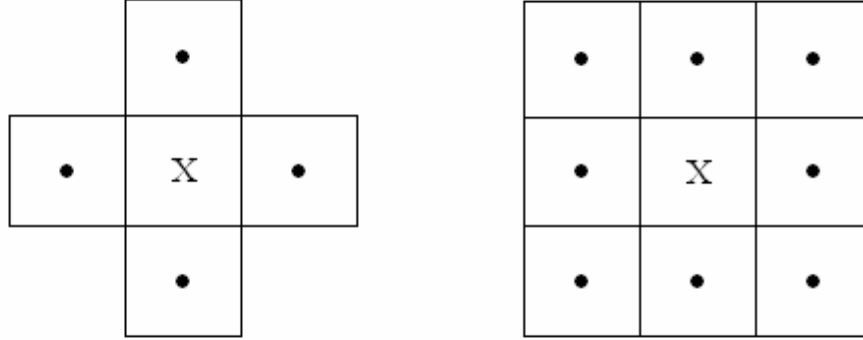
Doğal dünyadaki görüntüler her ne kadar kesiksiz uzayda tanımlanmış olsa da bir görüntünün bilgisayar ortamına aktarılması için sayısallaştırılması gerekir. Doğadaki görüntüler için genel topolojiyi kullanmak mümkün olmasına rağmen sayısal görüntüler sonlu sayıda elemandan oluştuğu için daha uygun bir topoloji kullanılması gerekir. Sayısal bir görüntü, piksellerin gri yoğunluk değerlerinin oluşturduğu bir kümedir. Bu küme üzerinde komşuluk, bağlılık, sınır gibi kavramları dijital topoloji kullanarak tanımlamak mümkündür. Dijital topoloji nesne sınırlarını, kenarları ve bölgeleri kesin bir doğrulukla tanımlayabilen ve yerlerini tespit eden uygulamaları içerir [16]. Genelde sayısal topolojiye ait tanımlamalar ikilik görüntüler üzerinde yapılmıştır. Gri seviye görüntülerin incelenebilmesi için, bulanık topoloji kavramı kullanılarak sayısal topoloji genelleştirilmiştir [17].

3.1 Bağlılık

Bağlılık kavramı dijital topolojinin temel kavramlarından biridir. Birbirine bağlı bileşenler görüntü içindeki temel bilgi birimleridir. Piksellerin birbirleri ile olan bağlılıkları görüntünün anlaşılabilmesi açısından önemlidir. Nesneyi oluşturan piksellerin birbirleriyle olan ilişkileri, bağlılık (connectivity) kavramının önemli bir uygulama alanı bulmasını sağlamaktadır [22].

Rosenfeld 1970'lerin başında görüntüyü, düğümleri piksellerden oluşan ve kolları komşu pikselleri birbirine bağlayan bir grafik olarak incelemeyi önermiş ve bu grafiği “komşuluk grafiği” olarak tanımlamıştır [17]. Böylece piksellerin oluşturdukları sınırlı sayıda eleman içeren kümeler için topolojik kavramları tanımlayabilme imkânı ortaya çıkmıştır. Bir pikselin yatay ve

düseydeki tüm komşuları, 4-komşuluk, buna ek olarak köşegenlerde bulunan bütün pikseller 8-komşuluk olarak adlandırılır Şekil 3.1’de bir pikselin 4-komşuluğu ve 8-komşuluğu gösterilmektedir.



Şekil 3.1 4'lü komşuluk (solda) ve 8'li komşuluk (sağda) grafikleri

İkilik görüntülerin topolojik analizleri sadece iki değerli olmasından dolayı kolaydır. Gri seviye görüntülerde ise görüntü 8 bitlik ise 0–255 arasındaki değerlerden oluştuğu için bağlılık tanımının uyarlanması gerekir. Bu görüntülerde bağlılığın incelenmesi için bulanık kümeler teorisinden yararlanılabileceği düşünülmüştür. 1979’da Rosenfeld [17] görüntünün piksel değerlerini, bulanık küme teorisine, ikilik formdan çok değerli forma modelleyerek dijital topolojiyi genişletmiştir. Eğer piksellerin yoğunluk değerleri $[0,1]$ aralığında ölçülürse, bir pikselin yüksek yoğunluklu parlak pikseller kümesi içindeki üyelik derecesinden (DOM-Degree of Membership) bahsedilebilir. Bulanık topoloji “bağlılık” kavramını, bağlantılılık derecesini (DOC- Degree of Connectedness) içine alacak şekilde düzenler. DOC ölçülen yoğunluk değerlerinden hesaplanabilir. Bulanık bağlılık tanımı, görüntüdeki piksellerin birbirleriyle ne ölçüde bağlı olduklarını göstermektedir.

3.2 Bulanık Küme Teorisi

Σ bir kesiksiz uzayda iki boyutlu pikseller kümesini gösteren bir dikdörtgen matris olsun. $A \subset \Sigma$, A bu kümenin bir alt kümesi olmak üzere ve $p \in A$, p de bu kümenin bir elemanı olsun. Bu ilişki Denklem (3.1) ile açıklanabilir:

$$\mu_A(p) = \begin{cases} 1, & \text{eğer } p \in A \\ 0, & \text{eğer } p \notin A \end{cases} \quad (3.1)$$

Buradan yola çıkarak bu fonksiyonun $[0,1]$ aralığında değer alabilen bir karakteristik fonksiyonunu tanımlarsak; $\mu_A(p)$, bir bulanık topoloji üyelik fonksiyonu olarak isimlendirilir. Bir p elemanı için kümenin elemanı olup olmadığını belirten 0 ve 1 değerlerini almak yerine, $\mu_A(p)$ artık p 'nin A içindeki (DOM) üyelik derecesini gösterir. p 'nin tamamıyla A kümesi içinde olduğu $\mu_A(p)=1$, A kümesi içinde olmadığını $\mu_A(p)=0$, ifadeleri gösterir. Eğer $0 < \mu_A(p) < 1$ ise p , A kümesi için kısmi bir üyeliğe sahiptir denir. Böylece Σ 'nin bir alt kümesi A içindeki her $p \in \Sigma$ için, $\mu_A(p)$ $[0,1]$ aralığında değer alıyorsa, A bir bulanık kümedir.

3.3 Bulanık Topolojinin Tanımı ve Özellikleri

Bulanık topolojinin sayısal görüntülere uygulanması Rosenfeld [17] tarafından sunulmuştur. Bütün bağıllık tanımları sonlu piksel ızgarası üzerinde hem 4-komşuluk hem de 8-komşulukta uygulanmıştır.

Σ bir görüntünün piksel yerleşimini koordinat düzleminde gösteren bir küme olsun. I görüntüdeki her pikselin yoğunluk değerlerini saklayan bir harita olmak üzere $I(p)$, p pikselinin yoğunluk değerini gösterir. $I(\Sigma)$ ise bütün görüntünün yoğunluk matrisini verir. I ve $I(\Sigma)$ birbirinin yerine kullanılabilirler. Bir görüntüdeki piksellerin üyelik derecesi şöyle hesaplanır:

$$\mu_1(p) = \frac{I(p)}{k}, \quad k = \max\{I(p) \mid p \in \Sigma\} \quad (3.2)$$

$\mu_1(p)$ bir pikselin normalize edilmiş yoğunluğudur ve görüntüdeki parlak pikseller içinde p 'nin üyelik derecesini vermektedir. Gri seviye bir görüntüde bütün pikseller için maksimum yoğunluk değeri (k) sabit olduğu için, $\mu_1(p)$ yerine $I(p)$ kullanılmasında bir sakınca yoktur.

$A \in \Sigma$ olmak üzere $\mathcal{V}: p = p_0, p_1, p_2, \dots, p_n = q$ dizisi A içinde p 'den q 'ya bir yol olsun. Bu yolun bağıllık kuvveti yol içindeki en zayıf noktanın değerine karşılık gelir.

$$S_A(\mathcal{Y}) \equiv \min \{ \mu_A(p_i) \} , \quad 0 \leq i \leq n , \quad \forall p_i \in \mathcal{Y} \quad (3.3)$$

Sayısal görüntülerde bir pikselden diğerine sonlu sayıda birçok yol bulunur. İki piksel arasındaki bağlantılılık derecesini hesaplamak için farklı tüm yolların bağlantılık kuvvetleri hesaplanır. İki piksel arasındaki yollardan bağlantılık kuvveti en yüksek olan bu iki pikselin bağlantılılık derecesini belirler.

$$C_A(p, q) \equiv \max \{ S_A(\mathcal{Y}) \} , \quad \mathcal{Y} \in A \quad (3.4)$$

Tanımdan da anlaşıldığı gibi iki pikseli birbirine bağlayan yollardan kuvveti en yüksek olan yolun bağlantılık değeri DOC'dir. Bir piksel kümesinin içinde tüm pikseller arasındaki bağlantılılığı belirlemek de bu yolla mümkündür.

$$C_A(A) \equiv \min \{ C_A(p, q) \} , \quad p, q \in A \quad (3.5)$$

Piksel kümesinin DOC kümesi içindeki en küçük değeri o kümenin bağlantılılık değerini verir. Bu da iki noktanın arasındaki bağlantılılığın tanımlamasıdır.

$$\mathcal{Y}^c : p = p_0, p_1, p_2, \dots, p_n = q \quad (3.6)$$

Eğer iki nokta arasında yalnızca bir yol varsa, bu yol üzerindeki tüm piksellerin bağlantılılık dereceleri iki uç pikselin üyelik değerlerinden küçük olanından büyüktür.

$$\mu_A(p_i) \geq \min \{ \mu_A(p), \mu_A(q) \} \quad (3.7)$$

Bu koşulu sağlayan yol için iki piksel birbirine bağlıdır. Benzer şekilde p ve q aşağıdaki koşulu sağlıyorsa bu pikseller içinde birbirine bağlıdır demek mümkündür.

$$C_A(p, q) = \min \{ \mu_A(p), \mu_A(q) \} \quad (3.8)$$

Eğer iki pikseli birbirine bağlayan yollardan en kuvvetli yol üzerindeki piksellerin üyelik dereceleri içinde minimum olanı bu iki uç pikselden birinin üyelik derecesine eşitse, bu iki piksel birbirine bağlıdır.

Şekil 3.2 iki pikseli birbirine bağlayan iki farklı yolu ve bu yol üzerindeki piksellerin yoğunluk değerlerini göstermektedir. p ve q arasındaki yollardan sol taraftaki yol, bağlantılık koşulunu sağlar, yol üzerinde p ve q 'nun üyelik değerinden daha küçük değerde bir üyelik değerine sahip bir piksel yoktur. p ve q 'nun birbirine bağlı olduğu göstermek için bu yolun bulunması yeterlidir. Sağ taraftaki

yolun bağıllık kuvveti p ve q piksellerinin minimum değerinden daha küçük olduğu için, bu yol p ve q piksellerini birbirine bağlamaz.

50	50	50	50	120	50	50	50	50
50	50	50	120	120	120	50	50	50
50	50	120	120	120	120	120	50	50
50	120	170	120	120	120	50	50	50
50	50	120	50	50	50	50	50	50
50	50	170	50	50	170	50	50	50
50	50	170	170	50	170	170	50	50
50	50	170	70	50	170	50	50	50
50	50	170	170	50	170	170	50	50
50	50	170	170	170	170	170	50	50
50	50	50	170	170	170	50	50	50

Şekil 3.2 p ve q piksellerini birbirine bağlayan iki farklı yoldaki yoğunluk değerleri

İki piksel arasındaki bağlantılılık değerini hesaplamak için bu pikseller arasındaki yolların tümünün DOC değerinin hesaplanması gerekir. Tüm yolların tek tek hesaplanması işlem yoğunluğunu arttıran bir yöntemdir. Şenel ve ark. [16] çalışmalarında bu yolların DOC değerini hesaplamadaki işlem yoğunluğunu azaltmayı sağlayan bir kavram kullanmıştır. Bu kavram bir piksel kümesi için piksellerin üyelik derecesini hesaplamayı kolaylaştıran bir yöntemdir.

3.4 α -Bağılantılılık Kavramı ve Özellikleri

Şenel [4] tarafından kullanılan α -bağılantılılık kavramı gri seviye görüntülerde DOC derecesini, ikili (binary) görüntülerde ise bağıllığı gösterir. Eğer p 'den q 'ya giden yolun bağılantılılık derecesi p 'nin üyelik derecesine eşitse p , q 'ya α -bağılantılıdır [4].

$$C_A(p, q) = \mu_I(p) \quad (3.9)$$

Diğer bir deyişle, p 'den q 'ya bağıllık kuvveti en yüksek olan \mathcal{Y} yolu üzerindeki tüm piksellerin üyelik dereceleri içinde, üyelik derecesi en küçük olan

p pikselidir. Bir görüntü üzerinde α -bağlantılılıkları belirleyebilmenin başka bir yolu da görüntüyü belirli bir t değerinde eşiklemektir. Gri seviye bir I görüntüsünün bütün piksellerini bir t değerinde eşiklersek ikili bir B_t görüntü matrisi elde edilir.

$$B_t = \begin{cases} 1, & \text{eğer } I(p) \geq t, \\ 0, & \text{eğer } I(p) < t \end{cases} \quad (3.10)$$

B_t görüntü matrisiyle ve A görüntü matrisinin kesişim kümesi $B_t \cap A$ içinde p ve q pikselleri birbirine bağlı ise yani, B_t içinde p 'den q 'ya giden ve tüm pikselleri 1 olan bir yol varsa, p , q 'ya α -bağlantılıdır denir. Birbirine herhangi bir yönde α -bağlantılı olmayan (p 'den q 'ya veya q 'dan p 'ye), iki piksel arasındaki yolun bağlantılık kuvveti bir r pikselinin üyelik derecesine eşitse, $S_A(\gamma) = \mu_I(r)$, r pikseli hem p hem de q 'ya α -bağlantılıdır. Ters şekilde, eğer r , p 'ye $\gamma(r, p)$ yolu üzerinden α -bağlantılı ve yine r , q 'ya $\gamma(r, q)$ yolu üzerinden α -bağlantılı ise, bu iki yolun birleşiminin bağlantılık kuvveti $S_A(\gamma(p, q)) = \mu_I(r)$, r 'nin üyelik derecesine eşittir. Açık ki α -bağlantılılık yansıma özelliğine sahiptir, $C_A(p, p) = \mu_I(p)$, p kendisine α -bağlantılıdır. Ayrıca geçişlilik özelliğine de sahiptir, p , q 'ya α -bağlantılı ise ve q da r 'ye α -bağlantılı ise p , r 'ye α -bağlantılıdır. Ancak simetrik değildir, p , q 'ya α -bağlantılı iken, $\mu_I(q) > \mu_I(p)$ olduğu için q , p 'ye α -bağlantılı değildir.

Bu yöntemle gerçekleştirilen DOC hesaplamalarında, p ve q arasındaki bütün yolları tek tek analiz edilmesine gerek kalmadığı için, işlem hacmini belirli ölçüde düşürmekte ve yöntemin hesaplama hızını arttırmaktadır. Yöntemin algoritması gerçekleştirilirken, eşik değeri olan t değerine ilk olarak $t = \min\{\mu_I(p), \mu_I(q)\}$, p ve q piksellerinden üyelik derecesi minimum olanı atamak algoritmanın verimini ve hızını arttıracaktır.

3.5 Bağlantı Haritası (The Connectivity Map)

Eğer p , $(2m+1) \times (2n+1)$ boyutundaki bir A penceresinin merkez noktası olursa, ($m > 0, n > 0$), bu pencerenin bağlantı haritası (DOC Map), p 'nin pencere

içindeki diğer piksellere nasıl bağlandığını gösterir. Şekil 3.3'te görülen bu tip haritalara bağlantı haritaları veya conn-map adı verilir.

140	141	141	140	139	140	75
148	145	145	148	126	141	60
131	114	104	103	85	95	55
102	98	96	100	90	100	45
96	97	101	102	88	80	44
90	95	100	95	85	60	42
85	88	94	94	99	80	40

(a)

100	100	100	100	100	100	75
100	100	100	100	100	100	60
100	100	100	100	85	95	55
100	98	96	100	90	95	45
96	97	100	100	88	80	44
90	95	100	95	85	60	42
85	88	94	94	99	80	40

(b)

Şekil 3.3. 7x7'lik pencere içindeki görüntünün Bağlantı Haritası (a) Gerçek görüntünün 7x7'lik bir kesiti, (b) 7x7'lik kesitin merkez pikseldeki değerine göre Bağlantı Haritası

Bağlantı haritası p pikselinin A penceresi içerisindeki diğer tüm q piksellerine olan bağlantılılık derecesini gösterir. $M(p) = I(p)$ harita içindeki piksellerin α -bağlantılılık ilişkisine göre yoğunluk değerlerini ilk aşamada tanımlar. p pikseli bu harita içindeki bazı piksellere α -bağlantılıdır, p 'nin α -bağlantılı olduğu piksellerin oluşturduğu alt küme A_x olsun, p 'ye α -bağlantılı olan piksellerin alt kümesi A_y ve p ile α -bağlantılılık ilişkisi bulunmayan piksel alt kümesi de A_z olsun.

X_i, Y_j, Z_k ($i, j, k > 0$) A penceresi içindeki piksel noktaları olmak üzere, eğer $X_i \in A_x$, ise conn-map üzerindeki değeri $M(X_i)=I(X_i)$ olarak aynı kalır. Eğer $Y_j \in A_y$, ise conn-map üzerindeki değeri $M(Y_j)=I(p)$ olur, $Z_k \in A_z$, ise conn-map üzerindeki değeri ise Z_k 'nın p pikseline olan bağlantılılık değerini alır. Tüm alt kümeler için özet halinde yazarsak;

$$X_i \in A_x \text{ olan tüm pikseller için, } M(X_i)=I(X_i)$$

$$Y_j \in A_y \text{ olan tüm pikseller için, } M(Y_j)=I(p)$$

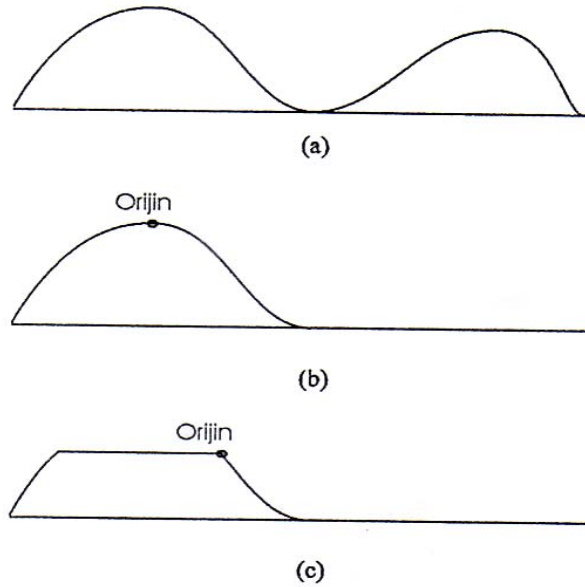
$$Z_k \in A_z \text{ olan tüm pikseller için, } M(Z_k)=C_A(Z_k, p).$$

olur.

3.6 Bağlantı Haritalarının Özellikleri

Bağlantı haritaları, gözlem penceresi merkezinde bulunan piksele bağlı olmayan piksellerin yok edilmesini sağlayabilir. Bir gözlem penceresi içinde yer alan piksellerin yoğunluk değerleri kullanılarak yeni bir pencere oluşturulmaktadır. Bu pencere içindeki değerler, merkez piksele olan bağlantılık derecelerini göstermektedir. Görüntünün içindeki nesnelerin nasıl algılandığına bakılarak iki farklı bağlantı haritası oluşturulabilir. Zira görüntü içinde birbirine bağlı parlak pikseller ve bunların oluşturduğu nesnelerin bulunması istenebileceği gibi, koyu piksellerin birleşerek oluşturdukları koyu nesnelerin de bulunması gerekebilir. Eğer görüntü olduğu gibi kullanılırsa, parlak pikseller için bağlantı haritası hesaplanabilir. Maksimum piksel değerine göre görüntünün tersi alınır, koyu pikseller için de bağlantı haritası elde edilebilir.

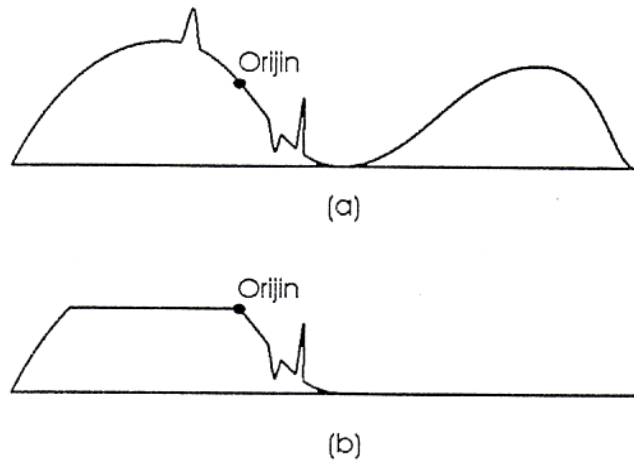
Bağlantı haritalarının nasıl çalıştığını görebilmek için, üzerinde birbirine yakın iki çizginin bulunduğu bir görüntü kullanılacaktır (Şekil 3.4.a). Bu görüntünün belli bir yerinden kesit alınarak, bir boyuta taşınması tercih edilmiştir.



Şekil 3.4 İki yakın çizgi nesnesinin oluşturduğu görüntü kesiti (a) orijinal kesit, (b) merkez, nesnenin tepe noktasında iken bağlantı derecesi haritası, (c) merkez yamaçta yer aldığı durumda bağlantı derecesi haritası

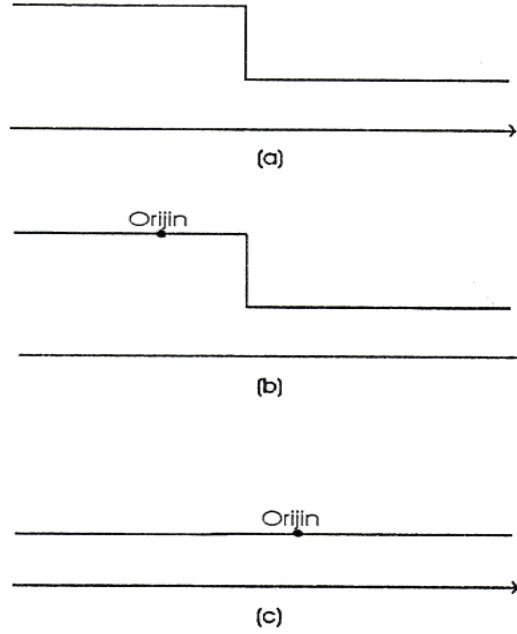
İki boyutlu uzayda bağlantıların çok farklı yollar üzerinden gerçekleşebilmesine rağmen, bağlantı haritasının nasıl çalıştığının gösterilebilmesi için kesit görüntüler yeterlidir [19]. Şekil 3.4.b'de gösterilen orijin referans alınır, ortaya çıkan bağlantı haritasında sağdaki çizgi görülmektedir. Görüntüde 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 3.4.c). Diğer bir deyişle, bağlantılı olmayan nesnelere bir miktar bozulmayla ortadan kaldırılmaktadır. Bağlantı haritasının bu özelliği sorun olarak görülse de, soldaki nesnenin bir kısmının ortadan kaldırılması sağdaki nesnenin kendi piksellerinin bağlılık özelliğini değiştirmeyeceği için ve bağlantı haritaları süzgeç penceresi çerçevesinde oluşturulduğu için aslında sorun teşkil etmemektedir [19].

Şekil 3.4.a'da gösterilen görüntüdeki üç piksele gürültü eklenirse (Şekil 3.5.a), Şekil 3.5.b'deki bağlantı haritası elde edilir. Orijin pikselinden yüksek olan ve gürültü içeren pikseller, soldaki nesnenin bir kısmıyla birlikte temizlenmektedir. Orijinin sağında yer alan gürültülerde deformasyon yaratılarak, sağdaki nesneyle birlikte yok edilmektedir. Kısacası, bağlantı haritasında gürültü bileşenleri eğer orijine bağlı değilse, bağlantılı olmayan diğer nesnelere birlikte temizlenmektedir. Bu bağlamda bağlantı haritası gürültü temizlemek için kullanılabilir [19].



Şekil 3.5 (a) Gürültü eklenmiş görüntünün kesiti, (b) Bağlantı haritası

Adım kenarlar için bağlantı haritası, kenarın yalıtılmasını sağlamaktadır. İdeal adım kenar, Şekil 3.6.a'da gösterilmektedir. Şekil 3.6.b, orijin soldaki yüksek bölüme konulursa elde edilen bağlantı haritasını göstermektedir. Şekilde görüleceği gibi, adım kenar bozulma olmadan bağlantı haritasına taşınmaktadır. Diğer yandan, orijin sağdaki düşük piksel değerli bölüme konulursa, bağlantı haritasında düz bir grafik elde edilmektedir (Şekil 3.6.c). Bu durum gözlem penceresindeki en düşük değerli pikselin, kendi piksel değerine eşdeğer derecede diğerlerine bağlanmasından ileri gelmektedir [16]. Çünkü en düşük değerli piksel her zaman diğerlerine bağlıdır. Bu örnekten de anlaşılacağı gibi, bağlantı haritası asimetrik şekilde, parlak nesnelere öne çıkarır şekilde çalışmaktadır.



Şekil 3.6 (a) Adım kenarın kesiti, (b) Orijin soldaki yüksek bölümde yer alırsa, elde edilen bağlantı derecesi haritası, (c) Orijin sağdaki bölümde yer alırsa elde edilen bağlantı derecesi haritası

İlk incelemede, bağlantı haritasında gözlemlenen asimetrik durumun sorunlar çıkarabileceği düşünülebilir. Yukarıda da belirtildiği gibi, bağlantı haritası görüntünün kendisi ve negatifinden olmak üzere iki şekilde elde edilmektedir. Her iki harita birbirini tamamlayan bilgilere sahiptir. Parlak ve koyu bağlantı haritalarından elde edilen bilgiler ve bağlantı haritalarındaki birbirini tamamlayan asimetriklik birleştirilebilir.

Bağlantı haritası oluşturmak için boyutları 3x3'ten daha büyük olan görüntü parçaları kullanılmalıdır. Çünkü 3x3 görüntü parçasında, 8'lik bağlantı kullanılırsa bütün elemanlar birbirine bağlı durumdadır. Aynı mantıkla hareket edilirse 5x5 parça da yeterli olmayabilir. Genellikle, bağlantı haritasının uygulanabilmesi için 5x5'lik gözlem pencerelerinden büyük parçalar seçilmelidir.

3.7 Birbirine Bağlı Bileşenlerin İşaretlenmesi (Connected Component Labeling-CCL)

Bağlantı haritalarının (conn-map) oluşturulması sırasında birbirine bağlı bileşenlerin işaretlenmesi (CCL) yöntemini kullanmak piksellerin birbirine olan bağlantısının varlığını belirlemek açısından kullanışlı olacaktır. A penceresi içinde ele alınan merkez p pikseline bağlantısı olan pikselleri CCL kullanarak tespit etmek daha kolaydır. p ile arasında bağlantıyı sağlayan bir yol bulunan ve p 'nin α -bağlantılı olduğu pikseller için DOC değerini bu haritayı kullanarak işaretlemek mümkündür.

Birbirine bağlı bileşenlerin işaretlenmesi işleminde ilk olarak görüntü taranır ve piksellerin birbirine bağlılıkları temel alınarak görüntü çeşitli bölgelere ayrılır. Bağlı bileşende bulunan her bir pikselin, piksel yoğunlukları birbirine eşittir ve birbirlerine aynı yoldan bağlıdırlar [18].

CCL ikili ve gri seviyeli görüntülerde çeşitli şekillerde kullanılabilir. Geliştirilen yöntemde ikili girdi görüntüsü 4-komşuluk esas alınarak uygulanmıştır. CCL algoritması görüntüyü satır bazında piksel değeri olan p noktasını bulana kadar tarar. p noktası bulunduktan sonra dördü komşuluğundaki noktalar sınanarak, tarama işlemi sırasında daha önce işlenip işlenmedikleri kontrol edilir. Bu bilgi temel alınarak, p noktasının işaretlenmesi aşağıdaki gibi yapılır [18].

- Eğer komşuluğundaki 4 noktada piksel değerleri 0 ise, p 'ye yeni bir etiket atanır.
- Eğer komşuluğundaki herhangi bir nokta 1 değerine sahipse, o noktanın etiketi p 'ye atanır.

- Eğer birden çok nokta 1 değerine sahipse, herhangi birinin etiketi p 'ye atanır ve diğer etiketlerin birbirine eşit olup olmadığı kontrol edilir.

Tarama işlemi sona erdikten sonra birbirine eşit olan etiketler tek bir etikete atanır. Son basamak olarak da görüntü tekrar taranarak, Şekil 3.7'deki gibi birbirine bağlı ama farklı etiketlere sahip noktaların etiketleri tek bir etiketle değiştirilir [18].

1	1	1	1	0	1	1
0	1	1	0	0	1	1
0	0	0	0	0	1	0
0	0	1	1	0	0	0
0	1	0	1	0	0	0
1	1	1	1	0	1	1
1	0	0	0	0	1	1

(a)

2	2	2	2	0	3	3
0	2	2	0	0	3	3
0	0	0	0	0	3	0
0	0	4	4	0	0	0
0	4	0	4	0	0	0
4	4	4	4	0	5	5
4	0	0	0	0	5	5

(b)

Şekil 3.7 CCL yönteminin uygulanması. (a) Gerçek görüntü kesiti (b) CCL yöntemiyle işaretlenmiş görüntü kesiti.

4. TOPOLOJİK ORTANCA SÜZGEÇLER

Topolojik ortanca süzgeçler 2002 yılında klasik ortanca süzgeçlerin performansını arttırmak amacıyla bulanık bağlantılık teorisi kullanılarak tanımlanmıştır[4, 16]. α -bağlantılılık [17] kavramını kullanarak, bir pikselin belirli bir süzgeç penceresi içinde kalan komşuluğundaki piksellerle olan bağlantılılık derecesi hesaplanarak görüntünün bağlantı haritaları oluşturulur. Bağlantı haritası, süzgeç penceresinin merkez noktasının komşuluğunda bulunan fakat merkez noktadan bir vadiyle ayrılmış tepe noktalarını, merkez noktayla tamamen bağlantısız bir hale getirir [16]. Bağlantı haritası üzerinden hesaplanan ortanca değer de sadece merkez noktayla bağlantısı olan tepe noktaları ya da düz alanlar üzerinden hesaplanır. Klasik ortanca süzgeçlerin aksine topolojik ortanca süzgeçler, merkez noktanın komşuluğundaki bağlantısı bulunmayan özellikler üstünde daha etkisizdir. Bu yüzden topolojik ortanca süzgeçler de gürültü yok edilirken görüntüdeki önemli özellikleri korunmuş olur [4,16].

Şenel [4], aşınma operatörü olarak adlandırabileceğimiz TL: aydınlık-bağlı-piksellerin topolojik ortancası ve genişleme operatörü olarak adlandırabileceğimiz TD: karanlık-bağlı-piksellerin topolojik ortancası, adında iki temel topolojik ortanca süzgeç tanımlamıştır. Bir görüntü kümesinin I , süzgeç penceresi A üzerinden, p bu süzgeç içinde bir piksel olmak üzere, aydınlık-bağlı piksellerin ortancası, TL işlemi aşağıdaki şekilde tanımlanabilir:

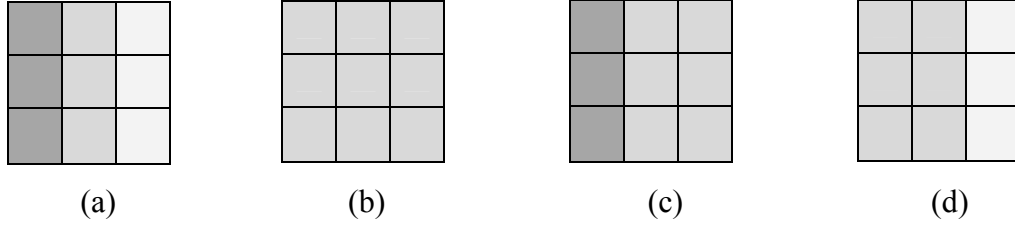
$$TL [I; A](p) = (I \ominus^T A) = \underset{q \in A+p}{median} \{C_{A+p}(p, q)\} \quad (4.1)$$

I görüntüsü içindeki karanlık bağlı piksellerin ortancası TD işlemi de şöyle tanımlanabilir:

$$TD [I; A](p) = (I \oplus^T A) = 1 - \underset{q \in A+p}{median} \{C_{A+p}(p, q)\} \quad (4.2)$$

Burada p ve q piksellerin görüntü üzerindeki yerleri, $C_{A+p}(p, q)$ 'de p ve q piksellerin bağlantılılık derecesini göstermektedir. TL doğrudan gerçek görüntüdeki pikseller üzerine uygulanır. TD uygulamak için I görüntüsünün negatifini kullanılmalıdır. Şekil 4.1'de görüldüğü gibi görüntünün bir A süzgeç penceresine TL uygulandığında merkez piksele α -bağlantılı olan aydınlık pikseller

merkez pikselin deęerini alacaktır. A süzgeç penceresinde TD uygulanırsa merkez piksele α -baęlantılı olan karanlık pikseller yine merkez pikselin üyelik derecesini alacaktır.



Şekil 4.1 3x3 görüntü kesiti (a) gerçek görüntü kesiti, (b) klasik ortanca yönteminin uygulanması, (c) TL uygulanması, (d) TD uygulanması

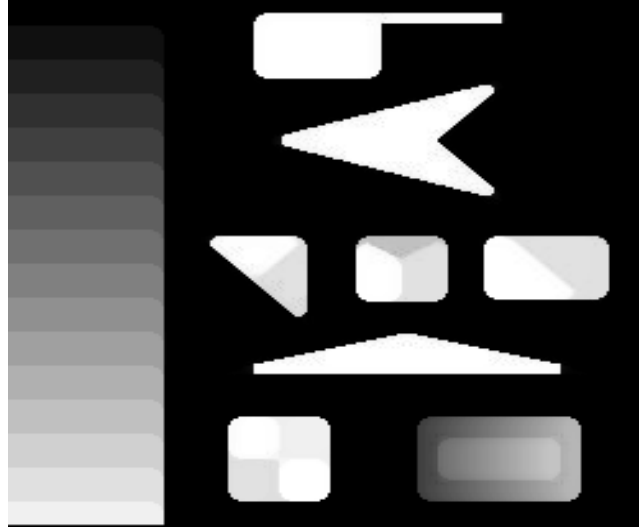
Bu yöntemin uygulaması sonucu görülen odur ki $TL [I ; A](p) \leq I(p)$ olur ve TL uygulanmış görüntü gerçek görüntüye göre klasik aşınma operatörü kadar olmasa da aşınma uygulanmış gibi olur. $TL[I ; A]$, ortalama olarak I görüntüsünden daha karanlık bir görüntü oluşturur. TL filtresi klasik ortanca filtreye göre gürültülü köşelerde daha keskin bir tepki verir [4].

TL aşınma gibi bir etki verirken, TD ise klasik genişleme operatörüne benzer şekilde genişleme yapar. TL ve TD operatörlerinin aşınma ve genişleme operatörlerine olan benzerliğinden dolayı, onlardan açma ve kapama operatörleri gibi çalışan, bileşik filtreler türetmek mümkündür [16].

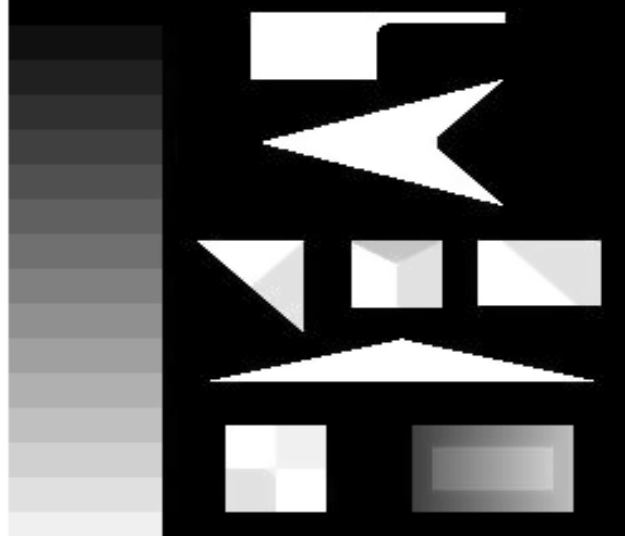
$$TDL [I ; A] = (I \ominus^T A) \oplus^T A = TD [TL [I ; A]; A] \quad (4.4)$$

$$TLD [I ; A] = (I \oplus^T A) \ominus^T A = TL [TD [I ; A]; A] \quad (4.5)$$

Şekil 4.2'de TL ve TD filtrelerinin ayrı ayrı uygulanmış görüntüleri yer almaktadır. TL filtresi konveks köşeleri aşındırırken, TD filtresi konkav köşelerde yumuşatma yapmıştır. Ayrıca köşeler aşınmış olmasına rağmen kenarlar korunmaktadır.



(a)

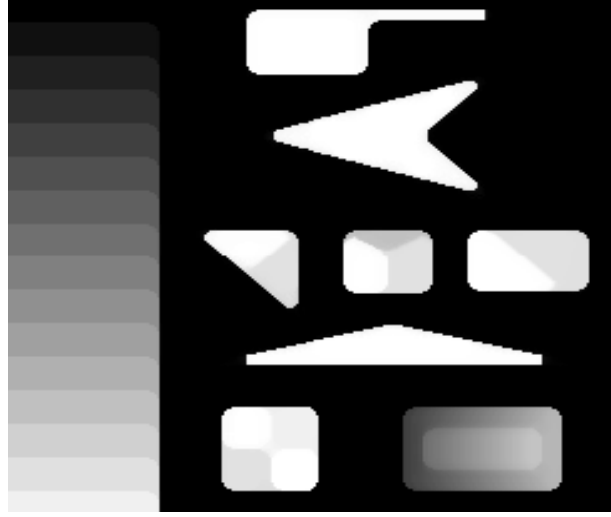


(b)

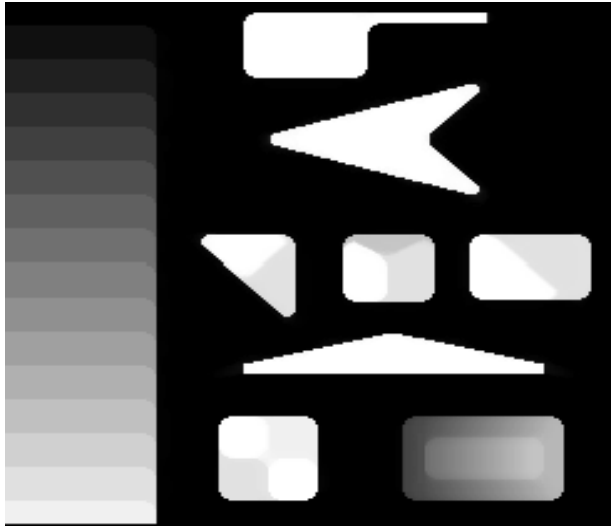
Şekil 4.2 TL ve TD uygulanmış görüntüler (a) TL (b) TD

TL ve TD filtrelerinde olduğu gibi TDL filtresi sanki açma operatörü gibi ve TLD filtresi ise sanki kapama operatörü gibi çalışır, ancak klasik açma kapama operatörleri ile aynı etkiyi göstermezler.

Şekil 4.3'te görüldüğü gibi TLD ve TDL filtresi yapay görüntülerde birbirine çok yakın sonuçlar vermektedir. Ancak görüntülerin yoğunluk değerleri incelendiğinde köşelerdeki ve gürültü bulunan alanlardaki değerleri farklılık göstermektedir.



(a)



(b)

Şekil 4.3 TLD ve TDL uygulanmış test görüntüleri (a) TDL , (b) TLD

4.1 Topolojik Ortanca Süzgeçlerin Etkileri

Topolojik ortanca süzgeçler (TOS), klasik ortanca süzgeçlerin (KOS) gürültü karşısında gösterdiği etkiyi iyileştirmek üzere geliştirilmiştir. KOS ile özdeş olan bu filtreler, izole impuls üzerinde ve izole gürültüsüz bir dik köşe üzerinde benzer etkiyi gösterirler. Ancak pencerenin yarısından fazla yer tutan, küçük şekiller üzerindeki etkileri farklıdır. Bu tür şekiller KOS tarafından ortadan kaldırılırken, sırasıyla karanlık veya aydınlık olma özelliğine göre TL ve TD filtreleri bu şekilleri korur [4].

Eğer orta bölgesi dar bir şerit şeklinde ve farklı tonda olan bir görüntüye geniş ölçekli bir pencere içinde KOS uygulanırsa, KOS bu dar şeridi ortadan kaldırırken, aynı boyutta bir pencere içinde parlaklığına göre TL ve TD'den uygun olanı uygulandığında, şeridin şeklini korur. Bu yüzden kullanılan topolojik filtrenin hangisi olduğu önemlidir. Karanlık şerit üzerinde TD, KOS gibi etki gösterirken, aydınlık şerit de TL, KOS gibi etkiye sahiptir. Buna bağlı olarak açık renkli bir görüntü üzerinde karanlık şekiller var ise bu görüntüye TL veya TDL, karanlık bir görüntü üzerinde açık renkli şekiller var ise TD veya TLD filtreleri uygulanmalıdır. Tam tersi şekilde kullanılırsa sonuçlar büyük süzgeç pencereleri içinde uygulanan KOS'un etkisinden bile kötü olabilir.

Eğer bir görüntü çift yönlü (hem karanlık, hem aydınlık) gürültü içeriyorsa TL ve TD'den biri kullanıldığında yalnızca bir yöndeki gürültünün etkisi azaltılmış olur. TDL ve TLD filtreleri oldukça simetrik ama bir miktar sapma gösterebilirler.

TLD ve TDL filtreler yoğun gürültü ortamında uygun tondaki görüntüye uygulandığında, özellikle kenarların korunması bakımından KOS'a göre daha etkilidirler. KOS kenarların keskinliği bozarken, topolojik filtreler bu etkiyi göstermez.

4.2 Topolojik Ortanca Yöntemiyle Köşe Bulma

Bu bölümde, gri seviye görüntüler üzerinde morfolojik aşınma ve genişleme operatörleri kullanılarak geliştirilmiş bir köşe bulma algoritmasından bahsedilmiştir [4]. Bu yöntem Şenel ve ark. [16] tarafından sunulan topolojik

ortanca süzgeç yöntemi kullanılarak morfolojik açma kapama işlemlerini kullanan köşe bulma yöntemidir.

Geliştirilen köşe bulma algoritması 3 aşamadan oluşmaktadır.

1. Adım :

$$D_1 = (I \oplus^T A) \ominus^T A \quad (4.6)$$

$$T_1 = |I - D_1|$$

2. Adım :

$$D_2 = (I \ominus^T A) \oplus^T A \quad (4.7)$$

$$T_2 = |I - D_2|$$

3. Adım :

$$\text{Köşeler} = T_1 \cup T_2 \quad (4.8)$$

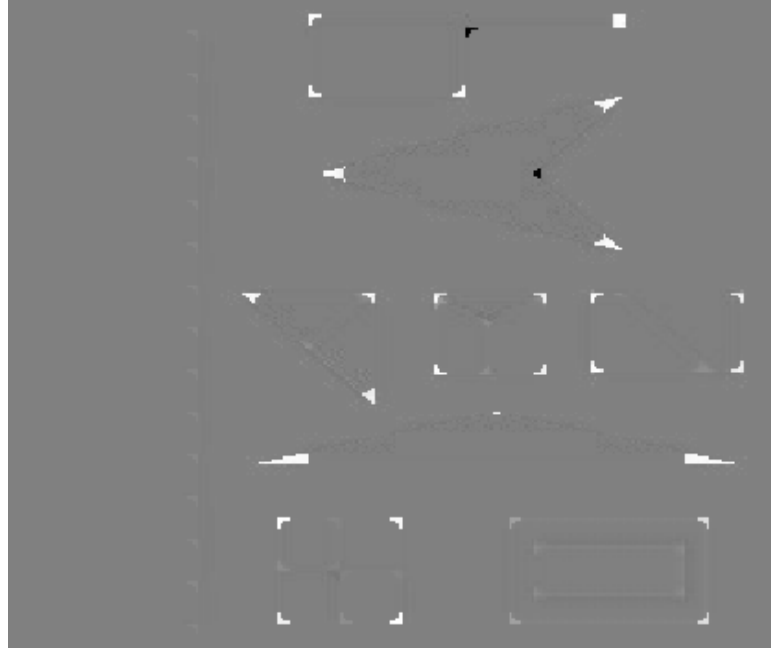
TL ya da TD'yi görüntüye uygulama işleminin basamakları birbirinin aynısıdır. Aradaki tek fark TD'in ters görüntü üzerinden uygulanmasıdır. İlk olarak maske penceresindeki pikseller (maske matrisi) küçükten büyüğe sıralanarak bir diziye (thrlist) kopyalanır. Daha öncede bahsedildiği gibi iki nokta arasındaki bir yolun büyüklüğü içerisindeki en küçük pikselin değeri kadardır. Bu yüzden başlangıç olarak bağlantı haritasının (maske penceresi boyutunda bir matris) tüm elemanları işlenecek olan penceredeki en küçük piksel değerine eşitlenir. Daha sonra pikseller arasındaki bağıllık derecelerini bulmak amacıyla maske matrisinin elemanları dizinin bir sonraki elemanı ile (maske matrisi için eşik değeri) karşılaştırılır. Bu değerden büyük olanlara "1", küçük olanlara da "0" değeri verilerek yeni bir matris elde edilir (maskthr). Elde edilen bu matris bağlı bileşenlerin işaretlenmesi işlemine tabi tutulur. Birbirine bağlı bölgeler aynı sayıyla etiketlendikten sonra, maske matrisinin orta noktasındaki elemanla aynı bölge içinde bulunan noktalar yani merkez noktaya bağlı olan noktalarının piksel değerleri eşik değeriyle değiştirilir. Bu işlem thrlist dizisindeki tüm elemanlar için uygulanarak bağlantı haritası ortaya çıkartılır. Elde edilen bağlantı haritasından hesaplanan ortanca değer yeni bir görüntüde merkez noktaya karşılık gelen pikselin yerine yazılır. Bu işlem tüm pikseller için tekrar uygulanarak yeni bir görüntü oluşturulur [4].

Maske penceresi olarak 5x5, 7x7, 9x9, 11x11 gibi dairesel bir maske kullanılabilir. Algoritmanın birinci basamağı görüntüdeki konkav, ikinci basamak konveks köşeleri bulmaktadır. 3. basamakta da bu köşeler birleştirilmektedir.

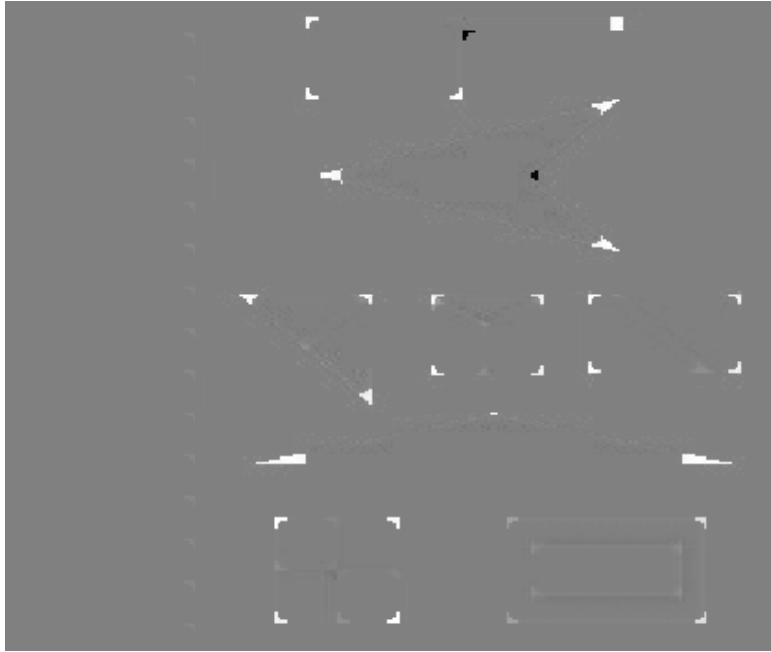
Köşelerin yerlerinin tam olarak işaretlenmesi için maksimum olmayanların bastırılması yöntemi (non-maxima suppression) kullanılmıştır. Bu yöntem sayesinde görüntülerde bir piksel kümesi halinde görülen köşeler tek bir nokta olarak işaretlenmiş olur.

Şenel ve Cihan [17] tarafından geliştirilen algoritma Shih ve ark.'nın [3] Laganiere'in algoritmasını temel alarak geliştirdiği algoritmaya benzemektedir. Morfolojik aşınma ve genişlemeyi aynı yöntemle kullanan Shih, Laganiere'den daha iyi sonuçlar elde etmiştir. Shih algoritmasının en önemli avantajı işlem zamanının çok kısa olmasıdır [16]. Fakat algoritma sadece belirli tür görüntülere (line-drawing images) uygulayabilmektedir. Yeni geliştirilen algoritma tüm gri seviye görüntüler üzerinde uygulanabilmektedir. İki algoritma da aşınma ve genişleme olarak tanımlanan iki kavramı kullanmasına rağmen, bu yöntem topolojik işlemlerde piksellerin birbirleriyle olan bağlantıları kullanıldığından, morfolojik işlemlere göre daha karmaşık ve zaman alıcıdır. İşlem zamanını azaltmak için algoritmanın 2. adımı uygulanmayabilir. Böylece az sayıda köşe göz ardı edilmiş olurken işlem zamanı yarı yarıya azaltılmaktadır [16].

Şekil 4.4'te Topolojik Ortanca Yöntemi uygulanılarak köşeleri tespit edilen test görüntüleri yer almaktadır. Topolojik Ortanca Filtreler uygulanarak elde edilen iki farklı görüntünün orijinal görüntüden farkının alınmasıyla oluşan I-TDL ve I-TLD görüntüleri köşeleri belirgin bir şekilde göstermektedir.



(a)



(b)

Şekil 4.4 Topolojik Ortanca Yöntemi uygulanarak köşeleri tespit edilen test görüntüleri I-TLD ve I-TDL uygulanmış test görüntüleri (a) I-TLD, (b) I-TDL

5. TOPOLOJİK ORTANCA YÖNTEMİYLE KÖŞE BULMA ALGORİTMASININ İYİLEŞTİRİLMESİ

Bu çalışmada Şenel ve Cihan [4] tarafından geliştirilen topolojik köşe bulma algoritmasının iyileştirilmesi ve geliştirilen yöntemin eski haliyle ve seçilen bazı köşe bulma algoritmalarıyla karşılaştırılması yapılmıştır.

Geliştirilen yöntem topolojik köşe bulma algoritmasında hesaplanan orijinal görüntüden elde edilen, köşeleri belirlenmiş T_1 ve T_2 görüntüleri üzerinde eş koordinatlı piksellerin maksimum ve minimum değerlerini ayrı ayrı karşılaştırmalı olarak hesaplar ve bulunan değerleri iki ayrı görüntü olarak oluşturur.

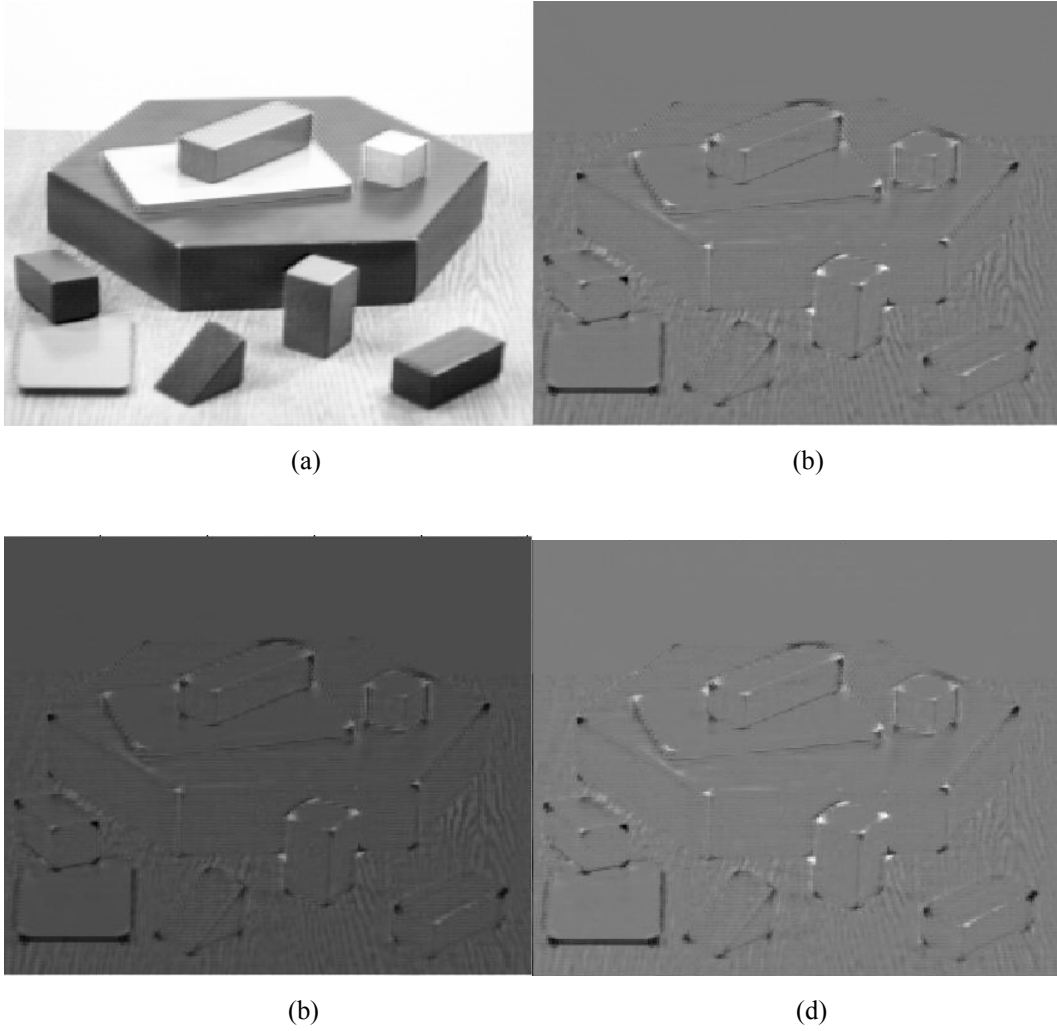
$$\text{MAX}(i,j) = \max \{ T_{1(i,j)} \cup T_{2(i,j)} \} \quad (5.1)$$

$$\text{MIN}(i,j) = \min \{ T_{1(i,j)} \cup T_{2(i,j)} \} \quad (5.2)$$

Daha ayrıntılı açıklayacak olursak orijinal görüntüye TDL filtresi uygulanıp görüntüden elde edilen fark ile, aynı şekilde, orijinal görüntüye TLD filtresi uygulanıp orijinal görüntüden çıkarılarak elde edilen iki farklı görüntünün aynı koordinatlı piksellerinden mutlak yoğunluk değeri büyük olanını, MAX olarak tanımladığımız görüntüye, aynı koordinatlı iki pikselden mutlak yoğunluk değeri küçük olanı da MIN olarak tanımladığımız görüntüye aktarılır. Böylece iki farklı görüntü elde edilmiş olur. MAX görüntüsü, aynı konumda, yoğunluk değerleri yüksek olan piksellerden, MIN görüntüsü de yoğunluk değerleri düşük olan piksellerden oluşur. MAX ve MIN görüntüleri karşılaştırıldığında, MAX görüntüsünün MIN görüntüsüne göre daha iyi sonuçlar verdiği gözlemlenmiştir. Çünkü MAX görüntüsü aydınlık değerlerin en yüksekini, karanlık değerlerin en düşüğünü içerir. Böylece aydınlık köşeler ve karanlık köşeler daha belirgin olacaktır. Ayrıca iki fark görüntüsünün toplanması ile elde edilen görüntü $T_1 \cup T_2$ gerçek yoğunluk değerlerini göstermemektedir. Bunun sebebi filtrelenmiş görüntüler orijinal görüntüden çıkarıldığında negatif değerler alabilmektedir ayrıca benzer iki görüntünün toplamı görüntünün sayısal sınırlarını aşmaktadır. Aynı koordinatlı bir piksele denk gelen iki değerden biri negatif biri pozitif değer taşıyorsa ve eğer ki bu piksel bir köşe noktası ise, iki değerlerin toplanması sonucu sifıra yaklaştıracığı için ve pikselin değeri belirli bir eşik değerinin altında

kalacağı için köşe tespit edilemeyecektir. Geliştirilen yöntem bu eksikliği ortadan kaldırır.

Şekil 5.1’de “Blok” görüntüsüne geliştirilen algoritmanın uygulanması ile elde edilen MAX, MIN ve $T_1 \cup T_2$ görüntüleri bulunmaktadır. Görüntülerde köşeler belirgin şekilde kendini göstermektedir. MAX, görüntüsünün hem konkav hem de konveks köşeleri tespit ettiği ayrıca $T_1 \cup T_2$ ile karşılaştırıldığında yüksek çözünürlüklü ve gerçek görüntülerde ayrıntılarda bulunan köşeleri daha iyi tespit ettiği gözlemlenmiştir.

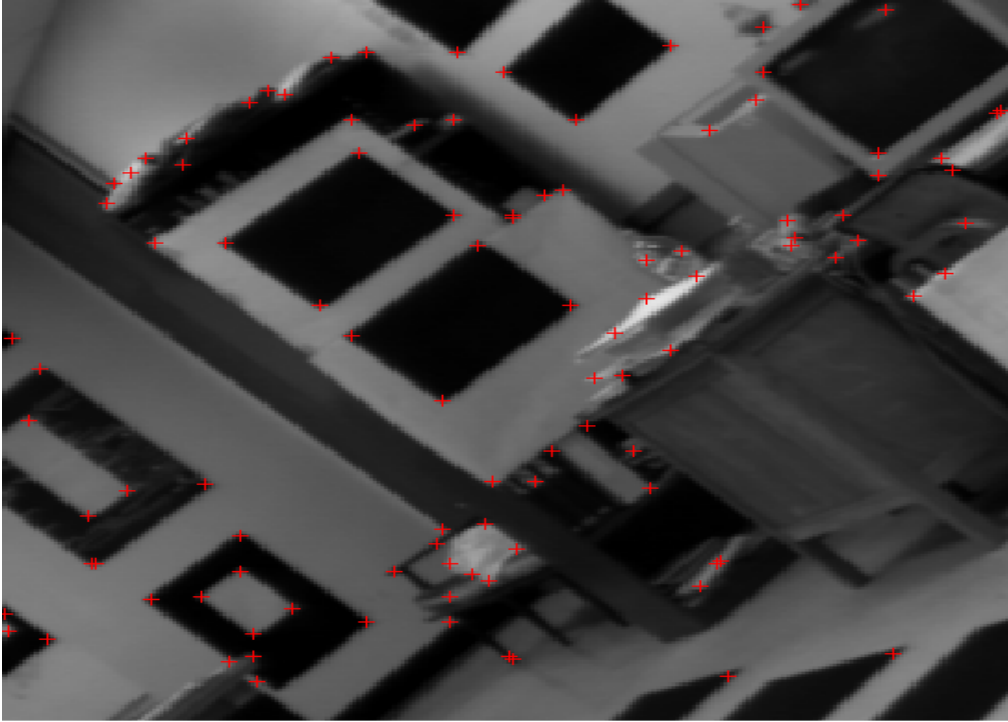


Şekil 5.1 Topolojik ortanca yöntemiyle köşe bulma algoritmasının sonuçları (a) orijinal görüntü, (b) $T_1 \cup T_2$ görüntüsü, (c) MAX görüntüsü, (d) MIN görüntüsü

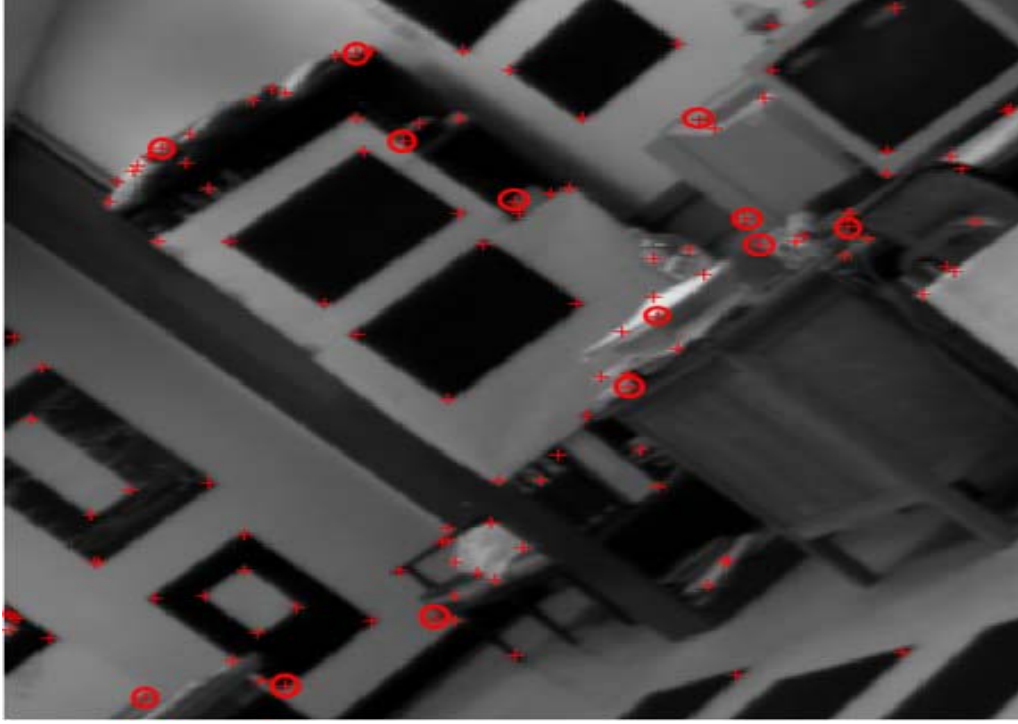
5.1 İyileştirilmiş Algoritmanın Karşılaştırma Sonuçları

Geliştirilen yöntemin başarımını diğer köşe bulma yöntemleri ile karşılaştırmadan önce eski yöntemden farkını gerçek görüntüler kullanarak açıklamak mümkündür. Şekil 5.1’de negatif görüntüleri görülen yöntemin, Şekil 5.2 ve Şekil 5.3’de karmaşık ve ayrıntısı fazla olan “Lab” görüntüsünün bir kesitinde farklılığı gösterilmiştir. Şekil 5.2’deki görüntü filtreler uygulandıktan sonra oluşturulan $T_1 \cup T_2$ işleminin sonucudur. Şekil 5.3 ise MAX işlemi uygulanmış görüntüdür. İki yöntemin köşeleri bulmada gösterdiği farklılıklar daire içine alınarak ortaya konmuştur. MAX görüntüsü eski yöntemin sonuçlarına göre 13 köşede farklılık göstermiş ve bu köşeleri doğru olarak tespit etmiştir.

T_1 ve T_2 görüntülerinin toplanması, iki görüntünün birinde pozitif diğerinde negatif değer alan aynı koordinatlı piksellerin görüntüdeki etkisini azaltır. Dolayısıyla köşenin yoğunluk değerinin eşik değeri altında kalmasına neden olur. MAX görüntüsü mutlak değer olarak büyük değeri seçtiği için karanlık veya aydınlık değeri korur ve köşelerin korunmasını sağlar.



Şekil 5.2 “Lab” görüntüsünün bir kesiti üzerinde önceki algoritmanın sonuçları



Şekil 5.3 “Lab” görüntüsünün bir kesiti üzerinde iyileştirilen algoritmanın sonuçları

5.2 İyileştirilmiş Algoritmanın Diğer Köşe Bulma Yöntemleri ile Karşılaştırma Sonuçları

Bir köşe bulma algoritmasının yapay görüntüler için diğer köşe bulma algoritmalarıyla karşılaştırılması nispeten kolaydır. Yapay görüntüler belirli geometrik şekillerin bir araya gelmesiyle oluşturulduğu için köşeleri gözlemsel belirlemek de mümkündür. Kesin doğru olarak kabul edilen gözlemsel olarak belirlenmiş köşeler ile kullanılan yöntemlerin tespit ettiği köşeleri gösteren sonuçların karşılaştırılması köşeler sayılarak yapılabilir. Doğal görüntülerde uygulanan yöntemler arasında karşılaştırma yapmak biraz daha zor olacaktır. Çünkü “köşe” olarak tanımlanan kavram ile görüntüler üzerinde yöntemlerin köşe olarak buldukları noktalar aynı olmayabilir. Bu yüzden doğal görüntüler üzerinde karşılaştırma yapılırken, daha önceden kesin doğru olarak kabul edilerek belirlenen köşeler ile yöntemlerden elde edilen sonuçlar sayısal olarak karşılaştırılır. Ayrıca yöntemin doğru tespit ettiği köşelerin yanı sıra, köşe olmayan yerleri köşe olarak belirlemesi ya da tespit edemediği köşelerin sayısı da yöntemin başarısını belirlemek için kullanılır. Bunların yanı sıra köşe noktalarının

yerlerini doğru tespit edip etmemesi veya ne kadar hatayla yaklaşık olarak doğru tespit etmesinin de incelenmesi yöntemlerin başarısının karşılaştırılması açısından yararlı olacaktır.

Bu çalışmada iki farklı tipteki görüntü üzerinde geliştirilen yöntem uygulanmış, aynı görüntüye belirlenen diğer köşe bulma algoritmaları da uygulanarak sonuçlar karşılaştırılmıştır. Yöntemlerin farklı tipteki görüntüler üzerinde karşılaştırılması amacıyla yapay ve gerçek görüntülerden faydalanılmıştır. Köşe bulma yöntemlerinde test amaçlı olarak genellikle tercih edilen, keskin kenarlı geometrik şekiller içeren ve birbirine yakın nesnelere içeren farklı görüntüler kullanılmıştır. Kullanılan görüntüler, gri seviye görüntüler ya da doğal görüntüler olarak tarif sınıflandırılabilir..

İyileştirilen algoritmanın başarısını sınamak için literatürde karşılaştırma prosedürlerinde bulunan beş algoritma Harris [8], Kitchen-Rosenfeld [1], Laganier [2], SUSAN [11], CSS [10] ve Bulanık Topolojiye Dayalı Köşe Bulma algoritması [4], değişik kontrast ve açıda köşeler içeren bir yapay görüntü ve bir de gerçek görüntü üzerine uygulanmıştır. Algoritmaların performansını ölçmek için tespit ettikleri doğru köşe sayısı ve tespit edemedikleri köşe (false negative-FN) sayıları ile köşe olarak tespit ettikleri köşe olmayan noktaların (false positive-FP) sayılarının toplamından oluşan Toplam Hata değerleri karşılaştırılmıştır.

Tüm köşe bulma algoritmaları için en iyi sonucu veren eşik değerleri seçilmiş ve tüm algoritmalarda aynı yöntemle gerçek köşe noktalarının yerleri tespit edilmiştir.

Şekil 5.4 ve Şekil 5.5 iki farklı görüntü için geliştirilen algoritmanın sonuçlarını ve diğer köşe bulma yöntemlerinin sonuçlarını göstermektedir. Çizelge 5.1 ve Çizelge 5.2’de ise algoritmaların bu iki görüntü için sayısal sonuçları listelenmektedir.

Çizelge 5.1: Şekil 5.4 için köşe bulma sonuçları

Algoritma	Doğru Köşeler	FP	FN	Toplam Hata
Kitchen	47	14	14	28
Harris	61	26	0	26
CSS	55	1	6	7
Laganiere	59	0	2	2
SUSAN	61	0	0	0
İyileştirilmiş Algoritma	61	0	0	0

Çizelge 5.2: Şekil 5.5 için köşe bulma sonuçları

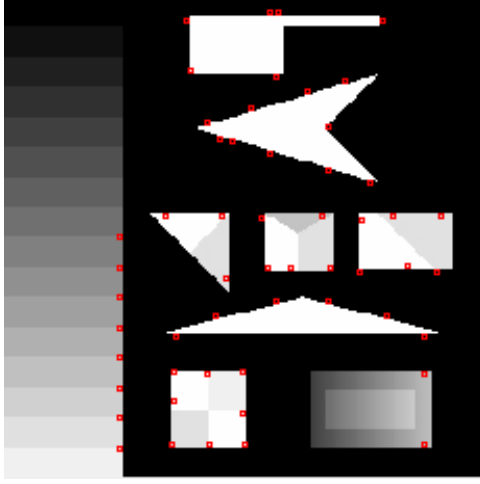
Algoritma	Doğru Köşeler	FP	FN	Toplam Hata
Kitchen	49	35	12	47
Harris	51	2	10	12
CSS	54	2	7	9
Laganiere	45	7	16	23
SUSAN	50	18	11	29
İyileştirilmiş Algoritma	55	5	6	11

Çizelge 5.1’de görüldüğü gibi yeni yöntem tüm köşeleri FN üretmeden doğru olarak tespit ederek bu görüntünün tasarımcısı olan ve görüntüye ismini veren SUSAN yöntemi gibi en iyi sonucu ürettiği ve onun başarısına ulaştığı görülmektedir. İkinci görüntüde ise algoritma diğer yöntemler içinde en fazla köşe bulan yöntem olmuştur ve CSS yöntemi dışında diğerleri arasında en az hata oranına sahiptir. Ayrıca benzer şekilde morfolojik işlemlere göre köşe bulan Laganiere algoritmasından çok daha başarılı bir sonuç verdiği görülmektedir. SUSAN yöntemi ilk görüntüde yeni algoritma ile birlikte en iyi başarıyı elde etmesine rağmen ikinci görüntüler karşılaştırıldığında yeni yöntemin başarısı daha açık bir şekilde görülmektedir. İlk görüntüde Harris yöntemi tüm köşeleri bulmasına rağmen çok fazla sayıda FN ürettiği için, Kitchen-Rosenfeld yöntemi ise daha az sayıda köşe tespit edebildiği için başarısı en düşük olan yöntemler

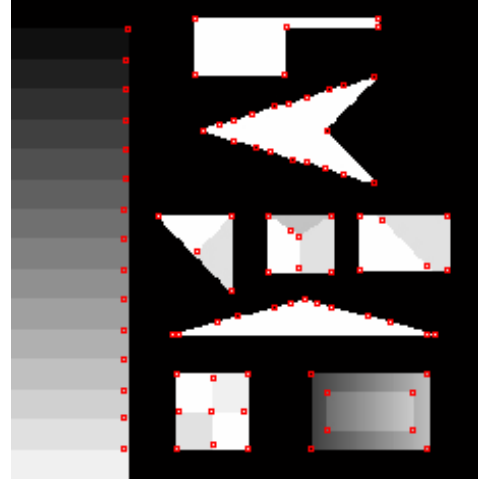
olduđu grlmektedir. İkinci grntde ise Kitchen-Rosenfeld yntemi ve SUSAN yntemleri yksek hata oranları nedeniyle en dřk bařarıma sahiptirler.

Topolojik ortanca yntemi kullanılarak kře tespiti yapan yntemin tek eksik yanı geniş aılı křelerde filtrelerin daha az ařındırma yapması sebebiyle bu křelerin tespitinde zorlanmasıdır. Maske penceresi iinde eđer geniş aılı blge geniş yer kaplamakta ve daha karanlık bir blge ise ortanca deđer křeyi oluřturan pikseller zerinde ok byk deđerim gstermemektedir. Bu nedenle bu křelerde fazla ařınma olmaz ve křelerin negatif grntde grlmesi ve dolayısıyla tespiti zorlařır.

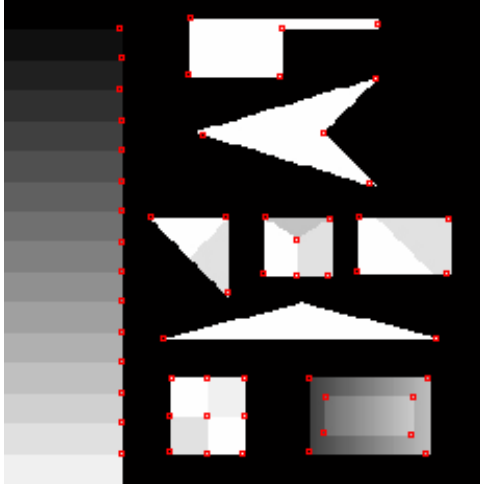
Sunulan algoritma, test edilmemiř olmasına rađmen, ortanca szge kullanımından dolayı, algoritmanın grltl grntler zerinde de iyi sonular vereceđi dřnmektedir. nk Topolojik Ortanca Filtreler grlt ieren grntlerde grltnn etkisini azaltarak grntleri yumuřatmaktadırlar. Bylece grlt etkisi zaten azaldıđı iin tekrar bir yumuřatma iřlemine gerek yoktur.



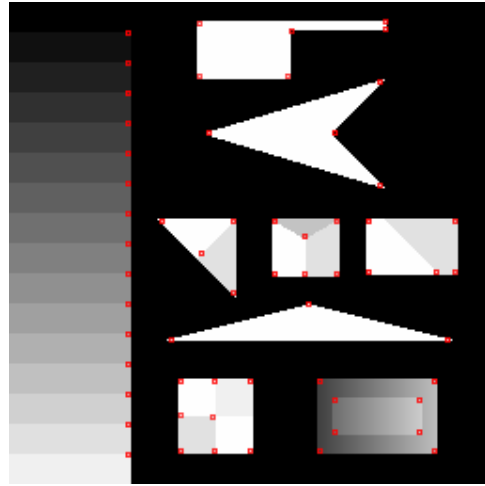
(a)



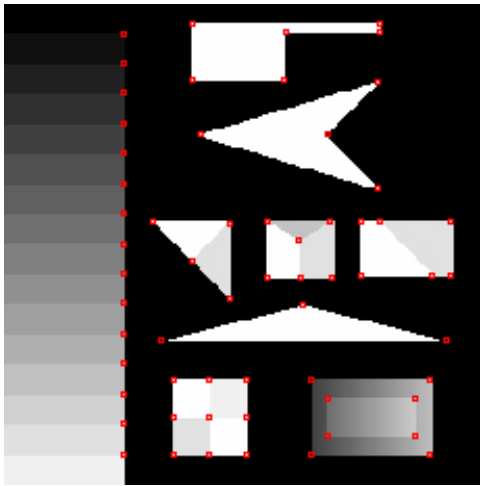
(b)



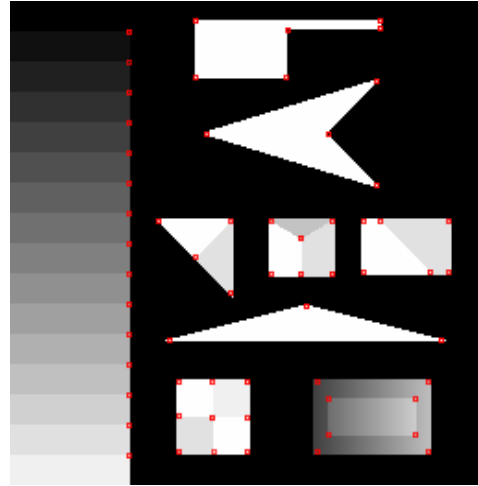
(c)



(d)

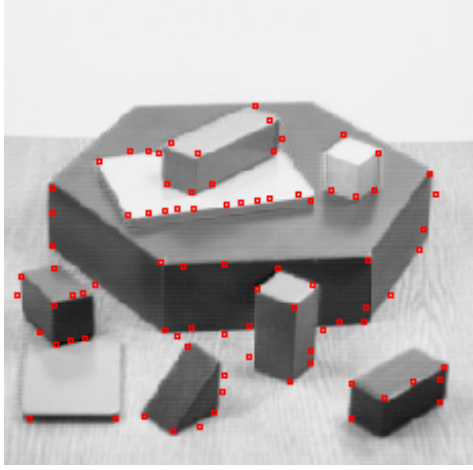


(e)

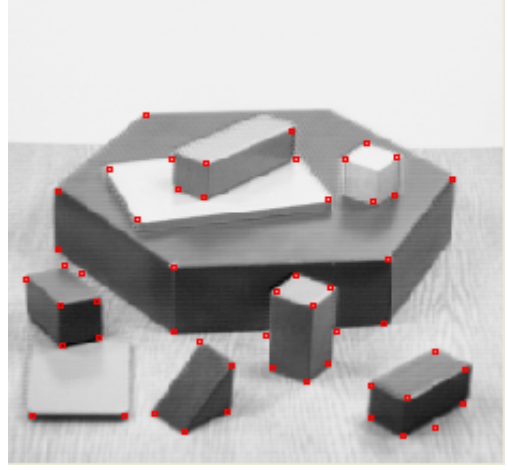


(f)

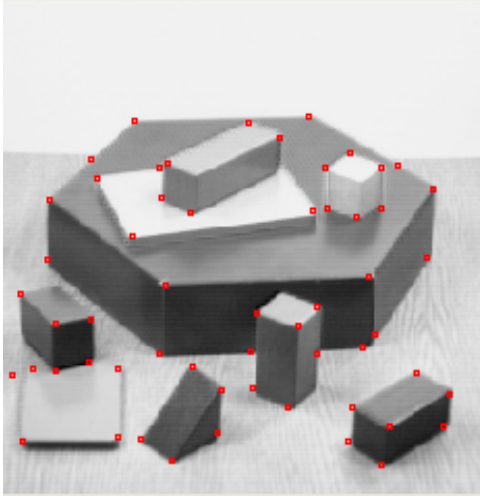
Şekil 5.4 Yapay Test Görüntüsünün Sonuçları: (a) Kitchen - Rosenfeld, (b) Harris, (c) CSS, (d) Laganiere, (e) SUSAN, (f) İyileştirilmiş algoritma



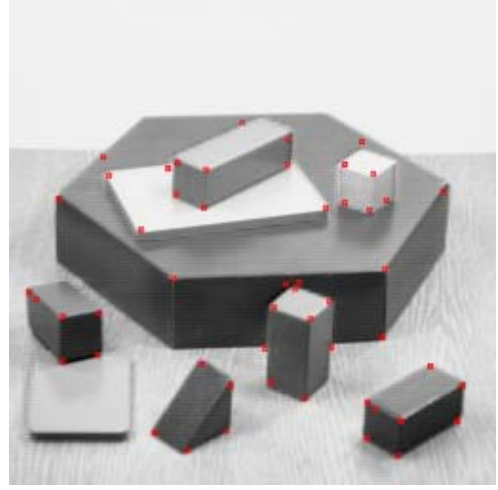
(a)



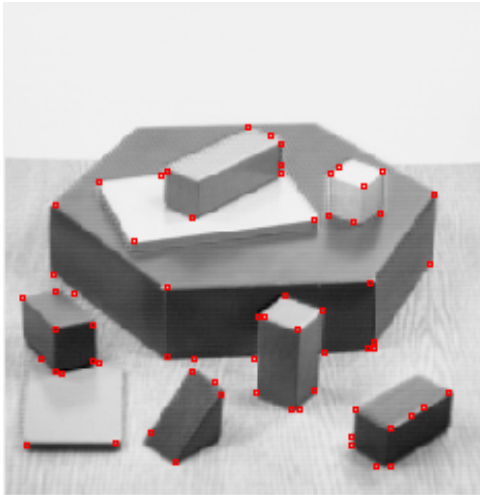
(b)



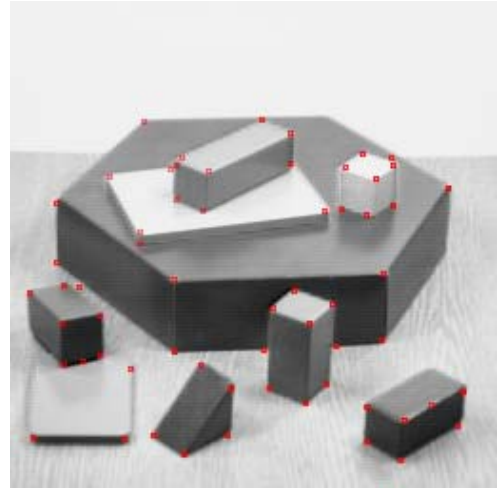
(c)



(d)



(e)



(f)

Şekil 5.5 Blok Görüntüsünün Sonuçları: (a) Kitchen-Rosenfeld , (b) Harris , (c) CSS , (d) Laganiere , (e) SUSAN , (f) İyileştirilmiş algoritma

6. SONUÇ

Bu çalışmada gri seviye görüntüler üzerinde köşe bulan bir topolojik köşe bulma algoritmasının iyileştirilmesi yapılmıştır. Geliştirilen algoritma yapay ve gerçek görüntüler üzerinde uygulanmış, doğru köşeleri başarı ile tespit ederken, düşük hata oranı ürettiği saptanmıştır. Algoritma üzerinde yapılan değişikliklerle yeni görüntü sonuçları eski görüntü sonuçlarına nazaran ayrıntılı görüntülerde küçük nesnelerin de köşelerini doğru olarak belirleyebilmektedir. Algoritma topolojik işlemler içerdiğinden ve harcanan zaman diğer algoritmalara göre fazla olduğundan ileride algoritmayı hızlandıracak çalışmaların yapılması ve yöntemin gerçek zamanlı uygulamalarda kullanılması planlanmaktadır.

KAYNAKLAR

- [1] Kitchen L. ve Rosenfeld A., *Gray level corner detection*, Pattern Recognition Letters, pp. 95-102, 1982.
- [2] Laganieri R., *A morphological operator for corner Detection*, Pattern Recognition **31** (11), s. 1643–1652, 1998.
- [3] Shih F.Y., Chuang C. ve Gaddipati V., *A modified regulated morphological corner detector*, Pattern Recognition Letters, s. 931-937, Nisan 2005.
- [4] Senel H.G., Cihan İ.K., *Gri Seviye Topolojik Köşe Tespiti*, 2007.
- [5] Lin R. S., Chu C. H. ve Hsueh Y. C., *A modified morphological corner detector*, Pattern Recognition Letters, **19** (3), s. 279–286, 1998.
- [6] Haralick R.M. ve Shapiro L.G., *Computer and Robot Vision*, Addison-Wesley, 1992 ve 1993.
- [7] He X.C. ve Yung N.H.C., *Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support*, Proceedings of the 17th International Conf. on Pattern Recognition, **2**, s. 791-794, 2004.
- [8] Harris C. G. ve Stephens M.J., *Combined Corner and Edge Detector*, In Proceedings of the Fourth Alvey Vision Conference, Manchester, s. 147-151, 1988.
- [9] Deriche R. ve Giraudon G., *A Computational Approach for Corner and Vertex Detection*, International Journal of Computer Vision, **10** (2), s. 101-124, 1993
- [10] Mokhtarian F. ve Suomela R., *Robust Image Corner Detection Through Curvature Scale Space*, IEEE Trans. on PAMI, **20** (12), s. 1376-1381, 1998.
- [11] Smith S.M. ve Brady J.M., *SUSAN - a new approach to low level image processing*, Int. Journal of Computer Vision, **23** (1), s. 45-78, May 1997.
- [12] Tomasi C. ve Kanade T., *Detection and Tracking of Point Features*, Carnegie Mellon University, Tech. Report CMU-CS-91-132, April 1991.
- [13] Zhang X. ve Zhao D., *A morphological algorithm for detecting dominant points on digital curves*, SPIE Proceedings, Nonlinear Image Processing 2424, s. 372–383, 1995.

- [14] Sobania A., ve Evans J.P.O., *Morphological corner detector using paired triangular structuring elements*, Pattern Recognition, **38** (7), s. 1087-1098, 2005.
- [15] Sojka E., *A new algorithm for detecting corners in digital images*, Proc. SCCG, ACM SIGGRAPH, NY, s. 55-62, 2002.
- [16] Senel H.G., Peters R.A. II, ve Dawant B., *Topological Median Filters*, IEEE Trans. Image Processing, **11** (2), s. 89-104, 2002.
- [17] Rosenfeld A., *Fuzzy Digital Topology*, Inform. Control, **40**, s. 76-87, Ocak 1979.
- [18] Aybar E., *Topolojik Kenar İşleçleri*, Anadolu Üniversitesi, Doktora Tezi 2003.
- [19] Hazer M., *Bulanık Topolojiye Dayanan Kenar Bulma Algoritması*, Anadolu Üniversitesi, Yüksek Lisans Tezi 2007.
- [20] Parks D. ve Gravel J.P., “*Corner Detectors*”,
<http://www.cim.mcgill.ca/~dparks/CornerDetector/index.htm>.
- [21] Cihan İ.K., *Topolojik ortanca yöntemi ile mamogramlarda kanser teşhisi*, Anadolu Üniversitesi Yüksek Lisans Tezi 2002.
- [22] Udupa, J. K., Samarasekera, S., *Fuzzy Connectedness and Object Definition: Theory, Algorithms and Applications in Image Segmentation, Graphical Models and Image Processing*, **58** (3), 246-261, 1996.
- [23] Alkaabi S., Deravi F., *Candidate Pruning For Fast Corner Detection*, Electronic Letters **40** (1), 2004
- [24] Beaudet P.R., *Rotationally invariant image operators*, International Joint Conference on Pattern Recognition, pp. 579-583, 1973.
- [25] Moravec H. P., *Towards Automatic Visual Obstacle Avoidance. Proc. 5th International Joint Conference on Artificial Intelligence*, pp. 584, 1977.
- [26] Canny, J. F. *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **8** (6), 1986.
- [27] Mokhtarian F. ve Mohanna F., *Enhancing the curvature scale space corner detector*, Proc. Scandinavian Conf. On Image Analysis, pp. 145-152, Bergen, Norway, 2001
- [28] Bovik, A., *Handbook of Image and Video Processing*, CRC Pres. 2003.

- [29] Mehrotra R., Nichani S. ve Ranganathan N., *Corner detection*. Pattern Recognition, 23:1223--1233, 1990.