

T.C.
OKAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FAZ ALAN TOPOLOJİK KISITLAMALARI

Necati SAĞIRLI

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN

Yrd. Doç. Dr. Rüyam ACAR

İSTANBUL, Temmuz 2018

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI



FAZ ALAN TOPOLOJİK KISITLAMALARI
YÜKSEK LİSANS TEZİ

NECATİ SAĞIRLI

tarafından

YÜKSEK LİSANS

derecesi şartını sağlamak için hazırlanmıştır.

Haziran 2018

Program: Bilgisayar Mühendisliği

FAZ ALAN TOPOLOJİK KISITLAMALARI

YÜKSEK LİSANS TEZİ

NECATİ SAĞIRLI

tarafından

OKAN ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümüne

Yüksek Lisans

derecesi şartını sağlamak için sunulmuştur.

Onaylayan:

Chair
Name
Danışman
Yrd. Doç. Dr. Rüyam ACAR

Member's
Name
Üye
Doç. Dr. Erchan APTOULA

Member's
Name
Üye
Yrd. Doç. Dr. Pınar YILDIRIM

Haziran 2018

Program: Bilgisayar Mühendisliği

ÖZET

Bu tezde yapılan çalışmanın amacı, ara yüzey modellemelerinde faz alan denklemlerini kullanırken topolojik evrimin sınırlandırılması için kısıtlama alanları oluşturmaktır. Faz alanlarının topolojik incelemelerinde uygulanan klasik yaklaşımlar, kritik alanların etkili bir şekilde belirlenmesinde yeterli olmamıştır. Bu sebeple dijital topolojideki basit olmayan nokta (kritik nokta) fikrinden yola çıkarak kritik alan hesaplama yöntemleri oluşturulmuştur. Bu yöntemler faz-alan özelliklerine özgü, bölge tabanlı yöntemler olup, lokal pencerelerde bağlantılı bileşen etiketleme algoritmaları ve filtreler kullanarak kritik alan hesaplamaları yapılmıştır. Elde edilen kritik alanlar uzaklık filtreleri de kullanılarak derecelendirilmiştir. Sonuç olarak faz alan topolojik evrimin sınırlandırması için bölge tabanlı ve derecelendirilmiş kritik alanlar oluşturulmuştur.

Anahtar Kelimeler: Dijital topoloji, kritik alanlar, faz-alan yüzey modelleme

ABSTRACT

The aim of this work is to create constraint areas for limiting topological evolution in phase field surface equations. The classical approaches applied in topological examinations of phase fields have not been effective in identifying critical fields. For this reason, we developed methods to determine topologically critical areas inspired by the simple point idea used in digital topology. These are region-based methods specific to phase-field properties, and critical area calculations are performed using connected component labeling algorithms and filters in local windows. Critical areas obtained are also rated using distance filters. As a result, region-based and graded critical areas are defined for limiting the phase field topological evolution.

Key Words: Digital topology, critical areas, phase-field interface modelling

TEŞEKKÜR

Bu tezin hazırlanmasında bilgileri ile bana her konuda ışık tutan, gün, saat aldirmaksızın büyük bir sabır ve hoşgörü ile değerli desteklerini esirgemeyen, tez danışmanım Sayın Yrd. Doç. Dr. Rüyam ACAR'a verdiği tüm emekler için en içten dileklerle teşekkürlerimi sunarım.

Bu çalışma boyunca uygulama-kodlama hususunda bana destek olan, beraber beyin fırtınası yaptığımız bilgisayar mühendisi arkadaşlarım Selahattin ve Talha'ya teşekkür ederim.

Bu aşamaya kadar gelmemi sağlayan, başta eğitimim olmak üzere her konuda destekçim olan çok değerli aileme, yoğun çalışmalarım sırasında sabır ve anlayış gösteren sevgili eşim Sultan'a teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ŞEKİL LİSTESİ.....	vi
I. GİRİŞ	1
II. TOPOLOJİ KURAMI.....	3
2.1 Morse Teorisi	4
2.2 Dijital Topoloji.....	5
III. FAZ ALAN MODELİ	7
IV. BAĞLANTILI BİLEŞENLER VE ETİKETLEME YÖNTEMLERİ	10
4.1 Özyinelemeli algoritma.....	10
4.2 Two Pass algoritması	13
V. TOPOLOJİK KISITLAMALAR İÇİN KRİTİK ALAN HESAPLAMALARI ..	18
5.1 Kritik Nokta	18
5.2 Birleşme ve Ayrılma	19
5.3 Bağlantılı bileşen sayılarına göre kritik alan hesaplaması	21
5.3.1 Bağlantılı bileşen sayılarına göre kritik alan uygulaması	22

5.4 Mask ile kritik alan hesaplamaları	33
5.5 Uzaklık filtresine dayalı kritik alan hesaplamaları	39
VI. SONUÇ VE GELECEK ÇALIŞMALAR	55
VII. KAYNAKLAR	57
VIII. ÖZGEÇMİŞ	59



ŞEKİL LİSTESİ

Şekil 2.1 Topolojik deęişim.....	3
Şekil 2.2 Morse fonksiyonu minima, saddle, maxsima örnek kritik alanlar (5).....	4
Şekil 2.3 8’li komşuya dayalı dijital topolojide basit ve basit olmayan noktalar	6
Şekil 3.1 Farklı iki bağlantılı bileşenin birleşme durumu (11).....	7
Şekil 3.2 Bir bağlantılı bileşenin incelerek ayrılma durumu (11).....	8
Şekil 3.3 Faz-Alan gelişimi (11).....	8
Şekil 4.1 Özyinelemeli algoritma adımları	12
Şekil 4.2 Algoritmalar için örnek görüntü	13
Şekil 4.3 Two Pass algoritması adımları.....	15
Şekil 4.4 İlk geçiş etiketlemesi	16
Şekil 4.5 Şekil 4.1’in two pass uygulanmış hali	16
Şekil 5.1 Kritik alan ve kritik nokta.....	19
Şekil 5.2 Birleşme ve ayrılma için kritik alan.....	20
Şekil 5.3 Birleşmeden dolayı kaynaklanan yerel topoloji deęişimi kısıt alanı	21
Şekil 5.4 Birleşme durumu için örnek görüntü	23
Şekil 5.5 Birleşme durumu kritik bölge.....	23
Şekil 5.6 Herhangi bir (i, j) noktası için 11x11’lik lokal bölge matris	24
Şekil 5.7 Şekil 5.6’nın two pass uygulanmış şekli.....	24
Şekil 5.8 İşlem adımı	25
Şekil 5.9 Birleşme durumu için ele alınan örnek görüntünün 11x11’lik lokal bölge matrislerine göre kısıtlanmış şekli	25

Şekil 5.10 Ayrılma durumu için örnek görüntü	26
Şekil 5.11 Ayrılma durumu kritik bölge	27
Şekil 5.12 Herhangi bir (i, j) noktası için 11x11'lik lokal bölge matris	27
Şekil 5.13 Şekil 5.12'nin two pass uygulanmış şekli.....	28
Şekil 5.14 Ayrılma işlem adımı	28
Şekil 5.15 Ayrılma durumu için ele alınan örnek görüntünün 11x11'lik lokal bölge matrislerine göre kısıtlanmış şekli	29
Şekil 5.16 Hedeflenen ve hesaplanan kritik bölge.....	29
Şekil 5.17 Şekil 5.9'dan alınan bir parça	30
Şekil 5.18 Şekil 5.15'ten alınan bir parça	30
Şekil 5.19 Kritik alanlardan alınan noktalar için lokal matrisler	31
Şekil 5.20 Birleşme ve ayrılma için kritik alanlar	32
Şekil 5.21 Tanımlanan masklar.....	33
Şekil 5.22 Dört yönlü mask	34
Şekil 5.23 Birleşme örnek görüntü	35
Şekil 5.24 Herhangi bir (i, j) noktası için 11x11'lik lokal bölge matris ve two pass uygulanmış şekli	35
Şekil 5.25 Dört yönlü mask ve lokal bölge matrisi.....	36
Şekil 5.26 Birleşme ve ayrılma durumları için bağlantılı bileşen sayılarına göre (sol) ve mask yöntemine göre (sağ) kritik alanlar.....	37
Şekil 5.27 Bağlantılı bileşen sayılarına göre (sol) ve mask yöntemine göre (sağ) kritik alanlar kıyaslaması.....	39
Şekil 5.28 Uzaklık matrisi.....	44
Şekil 5.29 Bölge matrisi.....	44

Şekil 5.30 11x11'lik örnek görüntü parçası.....	45
Şekil 5.31 Şekil 5.30'un two pass uygulanmış şekli.....	46
Şekil 5.32 11x11'lik uzaklık ve bölge matrisi	46
Şekil 5.33 Ele alınan lokal matris için uzaklık ve bölge matrisi.....	47
Şekil 5.34 Birleşme durumu için derecelendirilmiş kritik noktalara örnek	49
Şekil 5.35 Birleşme durumu için uzaklık filtresine dayalı farklı pencere boyutlarına göre kritik alanlar	50
Şekil 5.36 Ayrılma durumu için derecelendirilmiş kritik noktalara örnek	51
Şekil 5.38 Karmaşık yüzeyler için uzaklık filtresine dayalı farklı pencere boyutlarına göre kritik alanlar	54

I. GİRİŞ

Topolojik özelliklerin yüzeyler üzerinde ayrık modellenmesinin görüntü hesaplama üzerindeki etkisi büyüktür. Görüntü işleme alanındaki birçok çeşitli uygulama topolojik özelliklerin tanımlanmasına ihtiyaç duyar. Birçok çalışma, farklı uygulamalara gerek duyarak konuyu ele almaktadır. Bu çalışmamızda, faz alanı ara yüzey modellemelerinde topolojik yapının bozulmaması adına kritik alan hesaplamaları için uyguladığımız yöntemler anlatılmaktadır.

Faz alan denklemleri ara yüzey modellemelerinde uygulanan iyi bir yöntemdir. Fakat faz alan denklemlerini kullanırken topolojik evrimin sınırlandırılması için farklı uygulamalara ihtiyaç duyulmaktadır. Öncelikle topolojik yapının bozulmasında etkili olacak bölgelerin belirlenmesi gerekir. Bu durum kritik alan fikrini ortaya atar ve faz alan denklemi çalışırken belirlenen kırık alanlar dışındaki yüzeylerde denklemin çalışması sağlanır. Faz alanlarının topolojik özellikleri, bildiğimiz kadarıyla derinlemesine incelenmemiştir ve klasik yaklaşımlar kritik alanların etkili bir şekilde belirlenmesinde yeterli değildir.

Kritik alan hesaplamalarında morse teorisi ve dijital topoloji, araştırdığımız iki konu olmuştur. Morse teorisi ara yüzeylerdeki noktaların türevlerinin 0 olduğu her noktayı kritik nokta olarak kabul eder. Bu teori bizim düşüncemize ters düşmektedir. Aslında yapmak istediğimiz sadece bileşenler arasında kalan bölgelerde kritik alan hesabı yapmaktır. Dijital topolojide ise basit nokta fikri kullanılır. Seçilen herhangi bir nokta

kaldırıldığında veya eklenildiğinde bağlantılı bileşen sayısında değişme oluyor ise o nokta basit olmayan nokta veya kritik nokta olarak da adlandırılır. Faz-alan modellerinde kritik bölgeye karar verilirken, faz-alan özelliklerine özgü, bölge tabanlı bir yaklaşımın daha uygun olacağını öneriyoruz. Bu düşünceler çerçevesinde, faz- alan modellerinde topolojik açıdan kritik bölgenin belirlenmesinde uygulanan yöntemlere değinilmiştir. Bu yöntemler, dijital topolojideki basit nokta fikrinden esinlenerek lokal bölgeler üzerinde çalışılmış yöntemlerdir. Bu sayede lokal pencereler alınıp bu pencereler üzerindeki kritik noktalar üzerinden faz-alan kısıtlamalarına ulaşılır. Kritik yüzeylerin alan büyüklükleri, ele alınan lokal pencerelere göre değişkenlik gösterir. Bu uygulamayı daha da ileriye taşıyarak dereceli kritik alan hesabı yapılmaktadır. Burada filtreye dayalı bir hesap uygulanmaktadır. Belirlenen kritik noktalara uygulanan filtreler üzerinden kritiklik derecesi de atanarak kritik alan oluşturulur. Faz alan yüzeyleri geçiş bölgeleri olduğundan kritik alanlar da aynı zamanda birer geçiş bölgesidir. Kritik alanların derecelendirilmesi faz alan yüzeylerindeki geçişlerin derecelere göre oluşmasına olanak sağlar.

II. TOPOLOJİ KURAMI

Topoloji, şekil yüzeylerinin özelliklerini inceleyen geometri dalıdır. Şekil yüzeylerinin sürekli deformasyonlarında değişmeyen özellikleri üzerinde durur. Deformasyon sırasında şekil yüzeylerinde delik oluşturmak veya delikleri kapatmak, kesmek veya birleştirmek gibi işlemler topolojik yapının bozulmasına neden olur (1). Görüntüyü, delik yapısını koruyarak, çekilip büzülterek, topolojik yapısını bozmadan farklı bir şekle dönüştürebiliriz.



Şekil 2.1 Topolojik değişim

Genel topoloji bağlanma, kompaktlık ve süreklilik gibi mekanın yapısal özelliklerinin incelenmesini içerir. Özellikle bu özelliklerin sürekli deformasyonlar altında korunması üzerinde durulmaktadır. Bu özellikler set teorisi ve geometri ile incelenir (2).

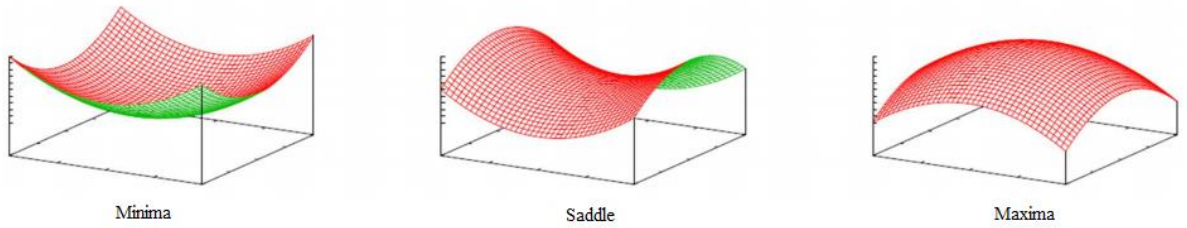
Cebirsel topoloji, homoloji ve homotopi gibi cebirsel araçlarla topolojik uzayların sınıflandırılmasını inceler. Homotopi, sürekli deformasyon altında değişmezliklerine dayanarak topolojik özellikleri sınıflandırır. X ve Y 'den biri diğerine sürekli olarak birbirine dönüştürülebiliyorsa homotopiktir (3).

Homoloji grubu ise, topolojik uzaydaki deliklerin sayısıyla ilgilidir. Euler özellikleri bir nesnedeki kenar, köşe ve yüz sayısını topolojik özellikleriyle ilişkilendirir. Morse teorisi kritik noktaları inceler; Bu teori, yüzeydeki ayırt edilebilir bir fonksiyonun kritik

noktalarını inceleyerek bir yüzey topolojisi ile ilgilenir. Dijital topoloji, görüntü analizlerinde görüntülerin topolojik yapısının incelenmesine olanak sağlar (2).

2.1 Morse Teorisi

Morse teorisi türevlenebilir manifoldların topolojisi üzerinde çalışmalar sonucu ortaya çıkmıştır. Morse teorisi, türevlenebilir manifoldu yine türevlenebilir bir fonksiyon ile inceler. Manifoldların topolojik tüm özelliklerini başka hiçbir şeye bakmadan anlayabilir ve böylelikle kritik noktaları da inceleyebilir. M türevlenebilir bir manifold olsun. M 'den reel sayılara tanımlanan bir f fonksiyonunun tüm türevlerinin 0 olduğu nokta f 'nin kritik noktasıdır. Bu genel morse fonksiyonudur. Fakat morse teorisi kritik noktaları dejenere olmayan fonksiyonlarla ilgilenir (4). Morse teorisi, bir yüzeyin topolojisini o yüzeydeki bir fonksiyonun kritik noktalarına bakarak inceler. Kritik noktalar bu fonksiyonun minima, maxima ve eyerleri olarak tanımlanır (5).



Şekil 2.2 Morse fonksiyonu minima, saddle, maxsima örnek kritik alanlar (5)

Bazen geometrik modellemelerde verilerin özelliklerini analiz etmek için fonksiyonların topolojisiyle ilgilenilir. Topolojik analizde de güçlü bir araç olan Reeb grafiği, seviye kümelerinin çıkarılmasını hızlandırmak için çok kullanışlı bir veri yapısıdır. Reeb grafikleri, seviye kümelerinin bağlı bileşenlerini noktalara aktararak elde edilir (6). Bununla birlikte, Reeb grafikleri tüm topolojik bilgileri

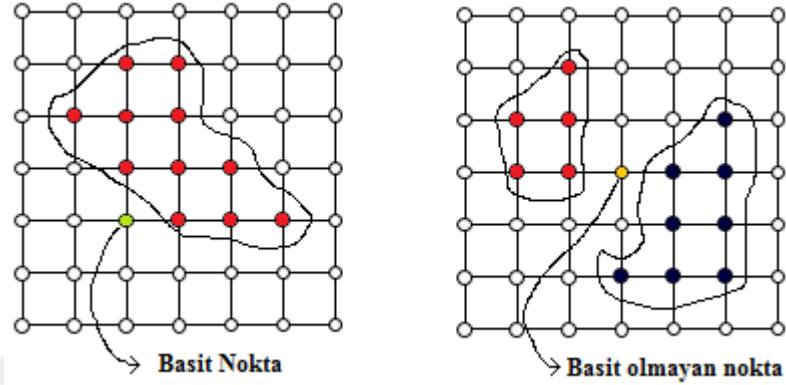
sağlayamayabilir; Örneğin, grafik yapısının, özelliklerin tamamen geometrik bağlantısını temsil etmediğinden, bazı boşluklar atlanabilir. Morse Smale kompleksi, verilerin gradyan akışının analizine dayanarak daha eksiksiz bir topolojik yapı oluşturur. Burada, seviye ayarlanan konturlar yerine, kritik noktalar arasındaki akış davranışı kullanılır. Bu akış davranışını temsil etmek için, kritik noktaları birleştiren entegre hatlar inşa edilir. Bu süreç, görüntüyü ayrılmaz çizgilerle ayrılmış düzgün bölgelere ayırır ve daha sağlam ve yapısal olarak kararlı bir topolojik tanım sağlar (2).

2.2 Dijital Topoloji

Dijital topoloji, görüntü analizlerinde topolojik yapıların incelenmesi düşüncesinden yola çıkarak Rosenfeld (7), tarafından ortaya atılmıştır. Görüntülerin analizi yapılırken bileşenleri bağlayarak, ayırarak çeşitli özellikleri incelenir. Bu sebeple bileşenler arasındaki sınırları çizmek, ilişkileri belirlemek amacı ile önemli bir matematik temeli sağlar (7). Dijital topoloji, görüntü evriminin topolojik özelliklerinin uygulanması olarak da tanımlanabilir.

Dijital topolojinin temel kavramı, bağlantı yaklaşımdan ortaya çıkmıştır. Buna göre, bir noktadaki bitişik ilişkiler gözlemlenmeye çalışılmıştır. Her ne kadar farklı komşular tanımlanmış olsa da, en çok bilinen ve uygulanan bağlantı tanımı Rosenfeld tarafından tanımlanan bir noktanın 4 ve 8 komşularına dayanmaktadır. Bu komşulara dayalı olarak elde edilen bağlı bileşenler, dijital topolojide homotopi gruplarını oluşturur (2). Dijital topolojide bir başka önemli kavram basit noktalardır. Aslında, bağlantı noktaları ile birlikte basit nokta kavramı dijital topolojinin temelini oluşturur. Bir noktanın silinmesi

görüntüdeki bileşenlerin sayısını deęiřtirmiyorsa o nokta basit nokta olarak tanımlanır (Şekil 2.3).



Şekil 2.3 8'li komşuya dayalı dijital topolojide basit ve basit olmayan noktalar

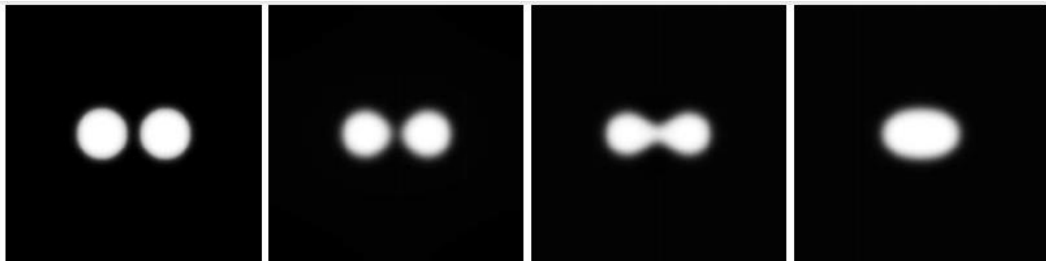
Görüntünün küresel topolojik özellikleri, bağlantılı bileşenlerin sayısı ile ilgilidir. Temel olarak, bileşenlerin sayısı deęişmez ise, o alanda topolojik yapı korunur (8). Basit olmayan noktalar da kritik noktalar olarak kabul edilir. Dijital topolojide, sayısallařtırmadaki belirsizlikler veya belirli uygulamanın gereklilikleri nedeniyle çeřitli kavramlar tanımlanmaktadır. Her seferinde tek bir nokta kaldırmak yerine, bir grup noktanın eşzamanlı olarak çıkarılmasının etkisi göz önünde bulundurulur (9). Buna göre kritik bölgeler veya minimal kümeler (minimum büyüklükteki basit setler) tanımlanmıştır (10).

III. FAZ ALAN MODELİ

Faz alan bir yüzey modelidir (11). İki fazın değerlerini içerir. Faz-alan modelinde keskin geçişli ara yüzler ince geçiş bölgeleri ile ifade edilir. Yüzey kuvvetleri bu geçiş bölgesine düzgün bir şekilde dağılır. Cahn-Hilliard faz-alan modeli, normal olarak bir ikili alaşım sisteminin faz ayrılmasını simüle etmek için kullanılır. Cahn-Hilliard denklemi diffüz ara yüz tipinin faz alan modellerini açıklar. Faz değerlerinin gelişimi aşağıdaki denklemle belirlenir.

$$\frac{\partial C}{\partial t} = M \nabla^2 \mu = M \nabla^2 (f'_\varphi(C) - \xi^2 \nabla^2 C) \quad (3.1)$$

Cahn Hilliard denklemi bir difüzyon denklemidir. Yüzey enerjisi minimizasyonunu sağlar. Bu enerjilerin minimizasyonunu ve denklemin gelişimini şu şekilde belirler. Yüzeydeki eğriliğin minimizasyonu yüzeyin düzleşmesini ve topolojik değişimleri sağlar. Topolojik değişimlerde yüzeylerin birbirine yakınlığına göre birleşmesi ve ayrılması durumları gerçekleşir.

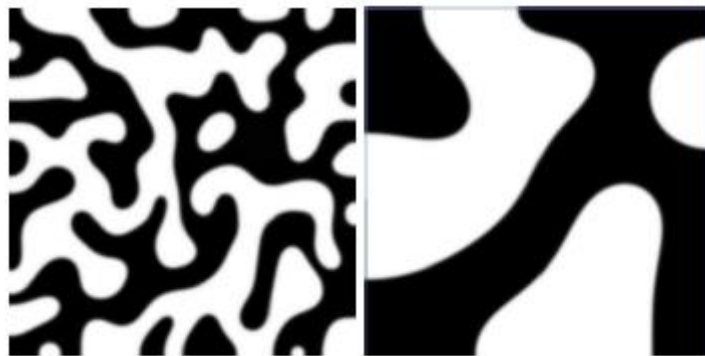


Şekil 3.1 Farklı iki bağlantılı bileşenin birleşme durumu (11)



Şekil 3.2 Bir bağlantılı bileşenin incelerek ayrılma durumu (11)

Kalınlık parametresi iki yüzey arasındaki geçiş uzaklığını verir. Yüzeyin kalınlığına göre denklemdaki difüzyon ve enerji terimleri karşılıklı çalışarak yüzeyin diffüz olmasını ve bir yandan da eğriliklerin azalmasını sağlar. Aynı zamanda yüzeyin dağılması, yüzey etrafında farklı katmanlar oluşturur ve iki yüzey birbirine yakın olduğu zaman iki yüzeyin dağılmasından oluşan katmanlar üst üste gelir. Bu durumda enerji devreye girer. Sadece uygun bir enerji fonksiyonu tanımlayarak farklı malzeme özelliklerini veya karmaşık ara yüz davranışları modellenebilir. Burada katmanların iki fazdan birisine geçiş yapması sağlanır. Yüzeyin kalınlığına göre bu gelişimler daha hızlı veya yavaş olabilir.



Şekil 3.3 Faz-Alan gelişimi (11)

Denklemdaki “ M ” sabiti ara yüz hareketliliğini temsil eder. Bu hareketlilik sabit hareketlilik olarak düşünülür. Bu durumda hareketliliğin artması daha çok zaman

harcamak demektir. \mathcal{E} (epsilon) parametresi ise ara yüz kalınlığını temsil eder ve ara yüz faaliyetlerinin etkileşim aralığını tanımlar. Yüzey kalınken bütün bu gelişimler daha hızlı olmaktadır. Yüzey inceldikçe gelişimler de yavaşlar. Fakat daha küçük ölçeklerde ve daha keskin şekilde modelleme ortaya çıkar (11).



IV. BAĞLANTILI BİLEŞENLER VE ETİKETLEME YÖNTEMLERİ

Birbirine bağlı olan piksellerin oluşturduğu kümeye **bağlantılı bileşen (connected-component)** denir. Bağlantılı bileşenler, görüntü işleme tekniklerine dayalı uygulamalarda önem arz etmektedir. Bir görüntü üzerindeki alan özelliklerinin tespit edilip analiz edilmesindeki rolü çok büyüktür (12). Bağlantılı bileşen etiketleme, dijital görüntülerdeki bağlı bölgeleri tespit etmek ve üzerinde çalışılması için kullanılır. Biz de çalışmalarımızda bağlantılı bileşenler üzerinden ilerleyeceğiz. Bağlantılı bileşenlerin topolojik değişimlerinde birbirine bağlanmaması ve kopmaması, üzerinde duracağımız konu olacaktır. Bu nedenle bağlantılı bileşenleri ayrı ayrı konumlandırmamız gerekecektir. Yukarıda da belirttiğimiz üzere farklı etiket numaraları atayarak bu işlemi sağlayacağız. Bağlantılı bileşenlerin belirlenmesi üzerine birçok farklı algoritma bulunmaktadır. Biz çalışmalarımızda bağlantılı bileşenleri etiketlemek amacıyla **özyinelemeli** ve **two pass (çift geçiş)** algoritmalarını inceledik.

4.1 Özyinelemeli algoritma

Özyinelemeli algoritma bağlantılı bileşen etiketleme yöntemlerinde de kullanılır. Her bir pikselin 4 veya 8'li komşuluklarını kontrol ederek etiketleme yapmayı hedefler (13). Siyah beyaz bir görüntü ele aldığımızı düşünelim ve 8'li komşuluklarını göz önünde bulundurarak bağlantılı bileşenleri etiketlemek için çalışalım. Öncelikle hangi grup üzerinde çalışmak istiyorsak (siyah ya da beyaz) o piksellere, etiket numaralarından da ayırt etmek amacıyla -1 ataması yapılır. Görüntü içerisinde -1'e eşit olan ilk noktaya

ulaştığında bu noktaya bir etiket numarası atar ve 8’li komşuluklarını kontrol etmeye başlar. Her 8’li komşulukta bulduğu -1 numarası için aynı etiketi atar ve bu etiket numarasını komşuluklarında -1 numarası bulamayana kadar kendi içerisinde dönerek tekrarlar. İşlem sonuna geldiğinde ilk seçilmiş olan bağlantılı bileşen etiketlenmiş olacaktır. Sonrasında görüntü içerisinde ikinci ilk -1 numarasını bulur ve farklı bir etiket numarası atar. Bu işlemlerin sonucunda bağlantılı bileşenleri etiketleyerek birbirinden net bir şekilde ayırmış olacaktır.

Bir etiket numarası tanımla(label)

Tüm görüntü üzerinde dön

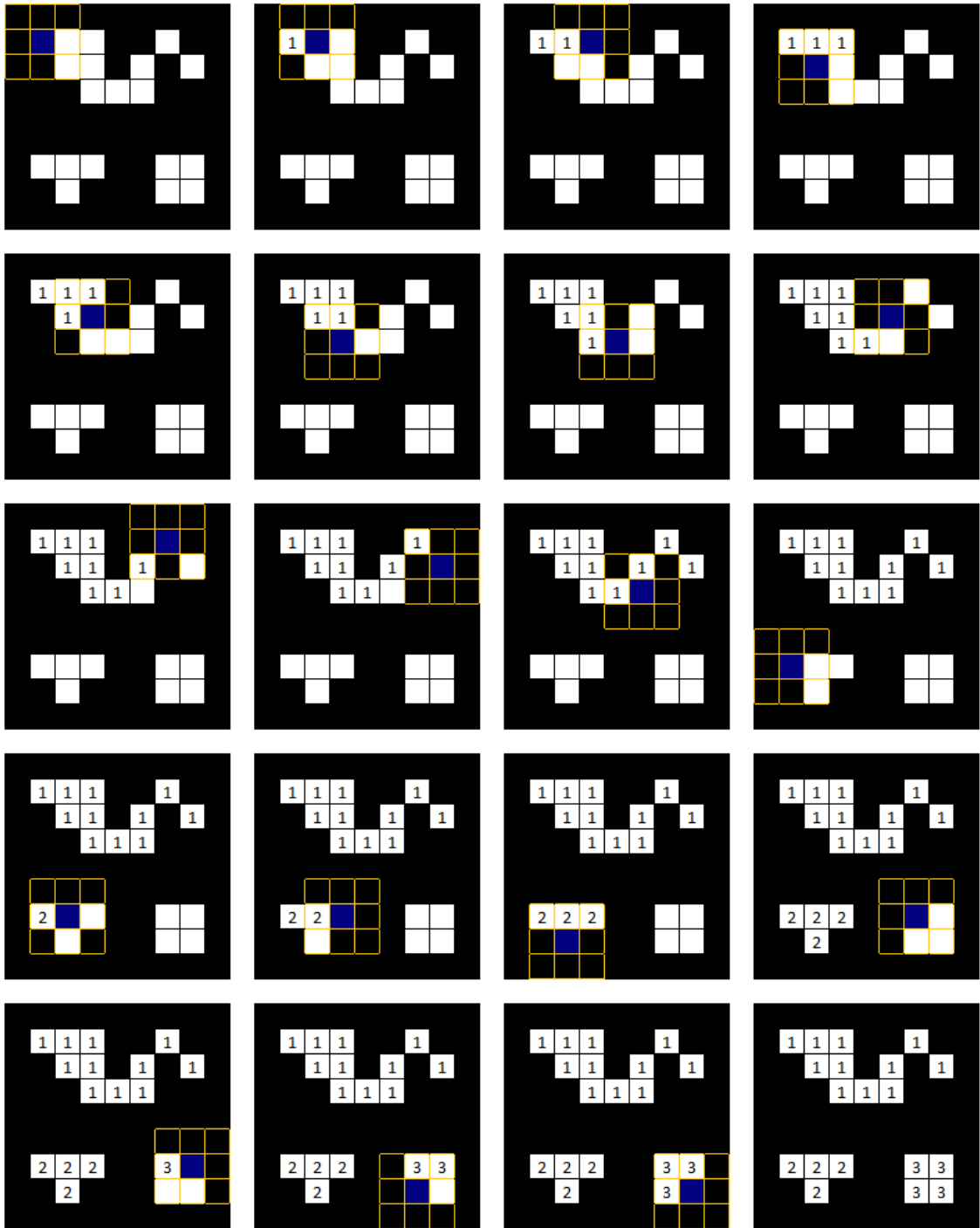
```
{
    Eğer piksel -1 ‘ e eşit ise
    {
        label++
        Ara fonksiyonunu çalıştır
    }
}
```

Ara Fonksiyonu

Label numarası ata

Tüm 8’li komşuluklarda dön

```
{
    Eğer komşu piksel -1 e eşit ise
    {
        Ara fonksiyonunu çalıştır
    }
}
```



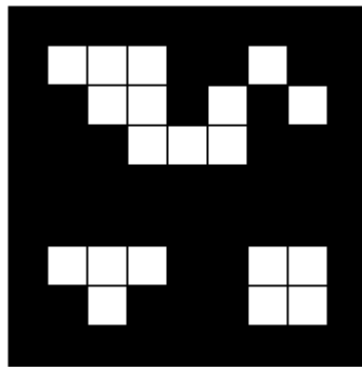
Şekil 4.1 Özyinelemeli algoritma adımları

Şekil 4.1’de özyinelemeli algoritma adımları gösterilmiştir. özyinelemeli algoritma ile bağlantılı bileşenleri etiketlerken bir bileşeni yakaladığı an o bileşendeki tüm etiket

atamasını yaptıktan sonra ikinci bileşene geçmektedir. Bu işlemi sürekli yaparak da tek bir geçişte görüntüdeki tüm bağlantılı bileşenleri etiketlemektedir.

4.2 Two Pass algoritması

Two pass algoritması isminden de anlaşılacağı üzere iki geçişli bir algoritmadır ve görüntü üzerinde iki geçiş yaparak etiketleme yapar. İlk geçişte her pikselin komşuluklarını kontrol ederek geçici etiketlerin atamasını yapar. İkinci geçişte ise denklik sınıflarını kontrol ederek her bir sınıfın en küçük etiketi ile önceki etiketi değiştirir (14). Two pass algoritması her bir pikselin çevresindeki komşuluk ilişkisi olarak ele alınan 4 ya da 8 pikseli inceleyerek etiketleme yapmayı hedefler. Bu durum, bağlantılı bileşenleri nasıl ayırmak istediğimize göre değişkenlik gösterebilir. Siyah ve beyaz renkten oluşan bir görüntü ele alalım ve 8’li komşuluklarını inceleyecek şekilde algoritmamızı tanımlayalım. Birinci ve ikinci geçiş olarak ayrı ayrı ele alalım ve sonrasında özyinelemeli algoritma adımlarını incelediğimiz görüntü (Şekil 4.2) üzerinde two pass algoritması adımlarını da inceleyip aradaki farkı tartışalım.



Şekil 4.2 Algoritmalar için örnek görüntü

Birinci geiş

Tüm beyaz piksellerde dön

{

Seilen beyaz pikselin 8 komşuluğunda dön (4 veya 8 komşuluk)

{

Eğer 8 komşuluğundaki beyaz piksellere atanmış herhangi bir etiket yok ise
benzersiz bir etiket ata

Eğer 8 komşuluğunda beyaz piksellere atanmış bir etiket var ise bu etiketlerin
en küçüğünü seç ve onu ata

Komşu etiketler arasındaki denklięi sakla

}

}

İkinci geiş

Tüm etiketlenen pikseller üzerinde dön

{

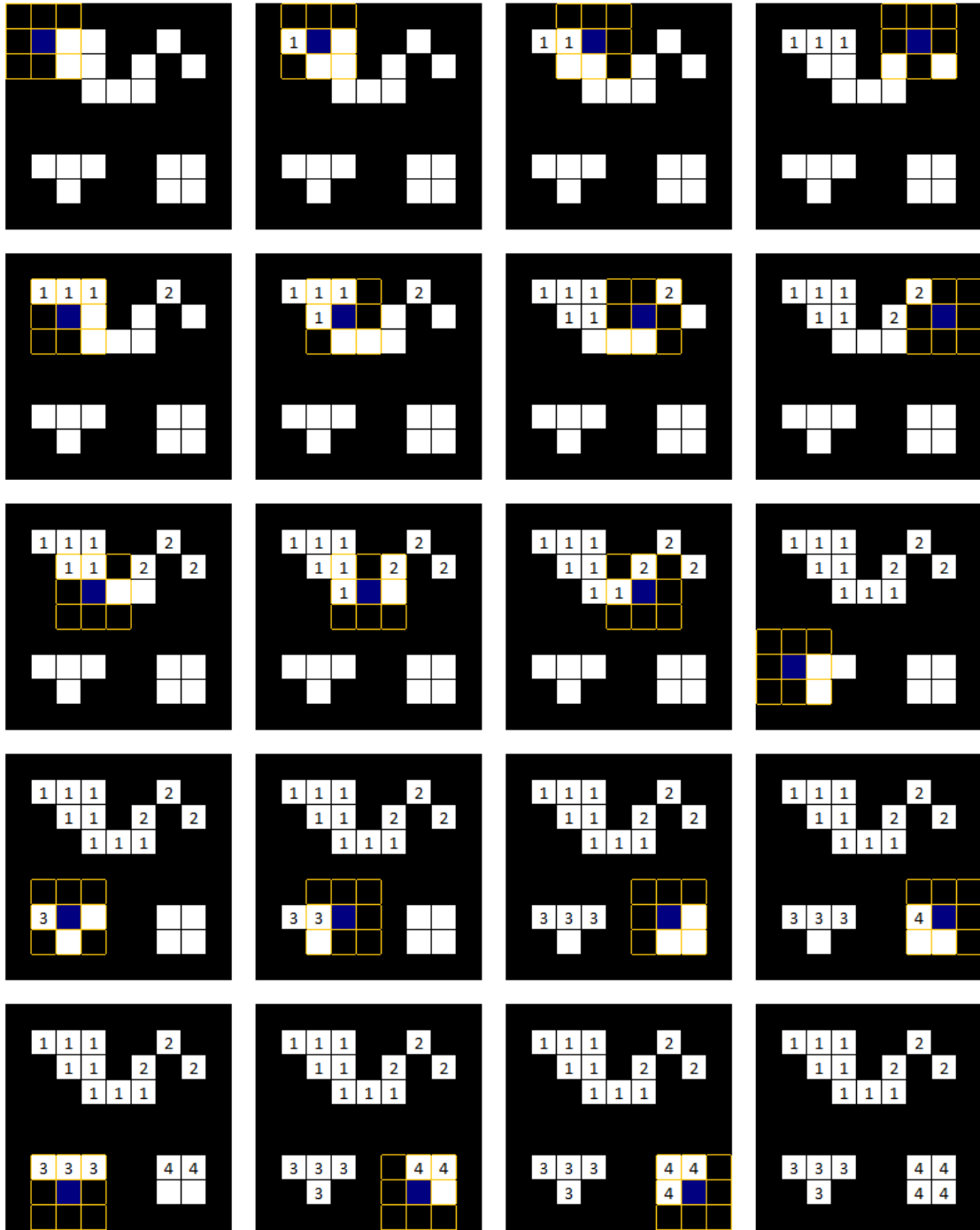
Seilen pikselin 8’li komşuluğunda dön

{

8’li komşuluğundaki en küçük etiketi bul ve denklięe göre o etiketi ata

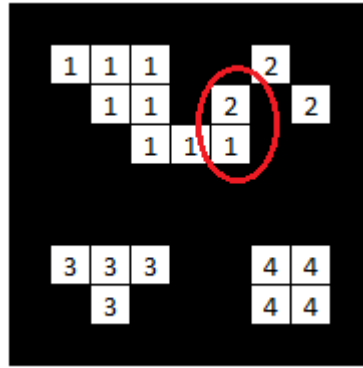
}

}



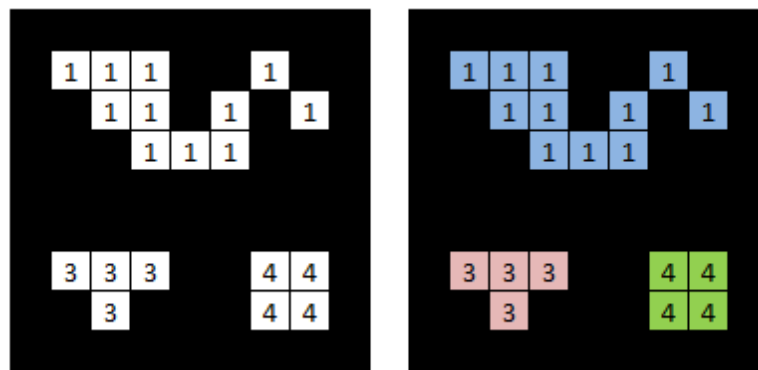
Şekil 4.3 Two Pass algoritması adımları

Şekil 4.3 te şekil 4.2'in two pass uygulanırken ki adımları gösterilmiştir. Burada verilen örnekler ilk geçişi göstermektedir. İlk geçişin etiketlenmiş son hali aşağıdaki gibidir.



Şekil 4.4 İlk geçiş etiketlemesi

Dikkat edilirse sağ üst köşedeki piksellerde 1 ve 2 rakamları komşuluk olarak yan yana gelmiştir. Bu durumlarda olan etiketlemeler için ikinci geçiş devreye girmektedir. İkinci geçiş her bir piksel etiketini, eşdeğer komşu elemanı etiketiyle değiştirir (14). Bu durumda ikinci bir geçiş daha yapılacak olunursa, 2 rakamı ile etiketlenmiş olan pikseller 1 rakamına dönüşecektir ve üstteki birbirine komşu pikseller aynı numara ile etiketlenmiş olacaktır. Bu işlemler sonunda bölgeler birbirinden tam anlamıyla ayrılmış olacaktır.



Şekil 4.5 Şekil 4.1'in two pass uygulanmış hali

Two pass algoritması yukarıda belirttiğimiz şekilde çalışmaktadır. Çalışmalarımızda two pass algoritması yardımıyla görüntüleri gruplara ayırıp işlemlerimize devam

edeceğiz. Bu şekilde bağlantılı bileşenleri net bir şekilde algılayıp, geliştirdiğimiz yöntemler ile kısıtlama yüzeylerini bulmayı hedeflemekteyiz.

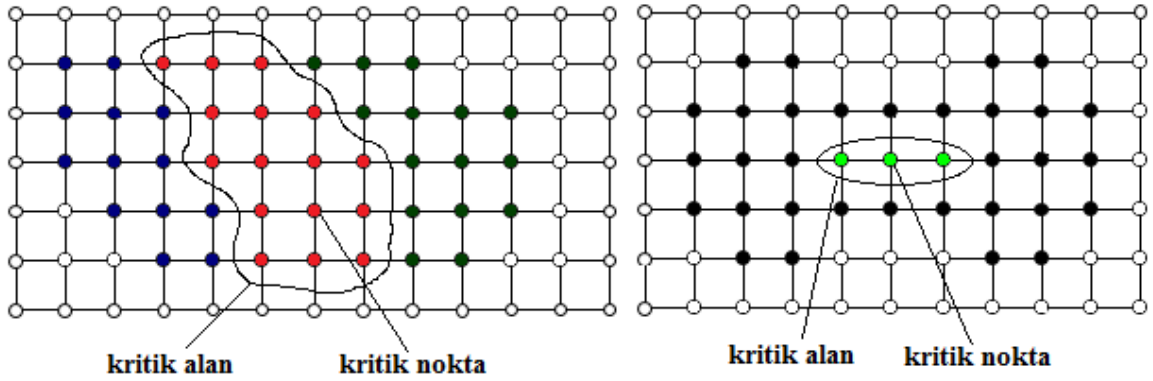


V. TOPOLOJİK KISITLAMALAR İÇİN KRİTİK ALAN HESAPLAMALARI

Görüntü modellemede topolojik yapının bozulmaması adına bağlantılı bileşenler için kısıtlama alanları oluşturmamız gerekmektedir. Görüntü üzerinde lokal parçalar alıp işlem adımlarını oluşturarak hesaplamalar yaptık. Lokal parçalarda en küçük eleman olan pikseller üzerinden ilerleyerek bu işlemleri gerçekleştirdik. Her bir lokal parçanın içerisinde tek bir kritik nokta belirleyip, tüm görüntü üzerinde hesaplanan kritik noktaların bileşiminden de kritik alanlara ulaşmayı hedefledik.

5.1 Kritik Nokta

Kısıtlama alanlarını oluşturan her bir nokta kritik nokta olarak adlandırılır. Kritik noktalar, birbirine belirlenen uzaklıktan yakın olan bağlantılı bileşenler arasında kalan noktaların her biridir. Diğer taraftan, yalnızca bir bağlantılı bileşenin belirlenen uzaklıktan fazla incelme durumunda arada kalan noktaların her biri kritik nokta olacaktır. Birbirine bağlı olan kritik noktalar kritik alanları oluşturmaktadır. Bu alanlar kısıt alanları olup resmin modellenmesinde bu alanlara dokunulmaması sağlanır. Şekil 5.1’de (sol) görüldüğü üzere yeşil ve mavi yüzeyler farklı iki bağlantılı bileşeni oluşturmaktadır. Arada kalan alan ise kritik noktaların oluşturmuş olduğu kritik alandır. Diğer taraftan, beyaz zemin üzerine işlenmiş siyah bölgede incelme durumu için kritik noktaları ve bu noktalardan da kritik alan oluştuğunu görmekteyiz.



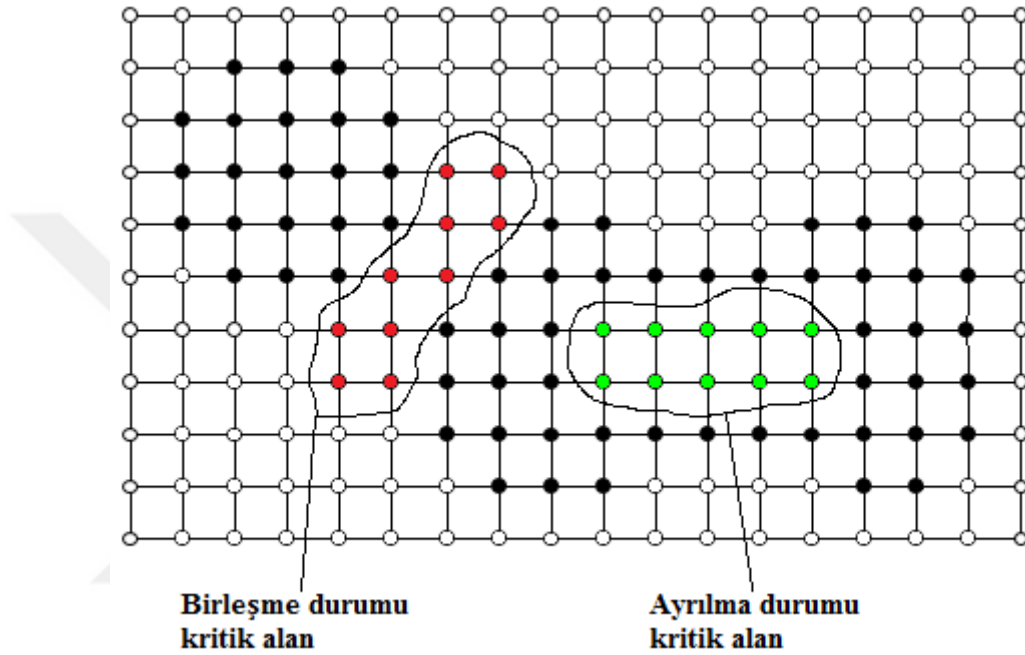
Şekil 5.1 Kritik alan ve kritik nokta

Aslında burada iki farklı durumun karşımıza çıktığını görürüz. Çalışmalarımızda bu durumları birleşme ve ayrılma durumları olarak nitelendirmekteyiz ve kritik alan hesaplamalarını, iki durumu göz önünde bulundurarak yapmayı hedeflemekteyiz.

5.2 Birleşme ve Ayrılma

Kritik noktaları belirlerken iki farklı durum karşımıza çıkar. Bu durumlardan birincisi birleşme durumu, ikincisi ise ayrılma durumudur. Birleşme durumlarında görüntü içerisindeki bağlantılı bileşen sayılarında azalma olur. Diğer taraftan ayrılma durumlarında ise bağlantılı bileşen sayılarında artma olur. Her iki durumda bağlantılı bileşen sayılarında değişiklik olduğundan görüntünün topolojik yapısının bozulduğunu görürüz. Bu nedenle kritik noktaları belirleyip kritik alanları oluşturduktan sonra kritik alanlarda herhangi bir işlem yapılmamasını hedefleriz. Görüntü modellemesi yapılırken birden fazla bağlantılı bileşen arasında geçişin olmasını engellemek için bulunan kırık noktalar birleşme durumunu simgelemektedir. Burada yapılmak istenen, modelleme yapılırken bağlantılı bileşenlerin genişlemesi durumunda diğer bağlantılı bileşenlerle belirlenen alandan fazla yaklaşmasını engellemektir. Tek bağlantılı bileşen üzerinde

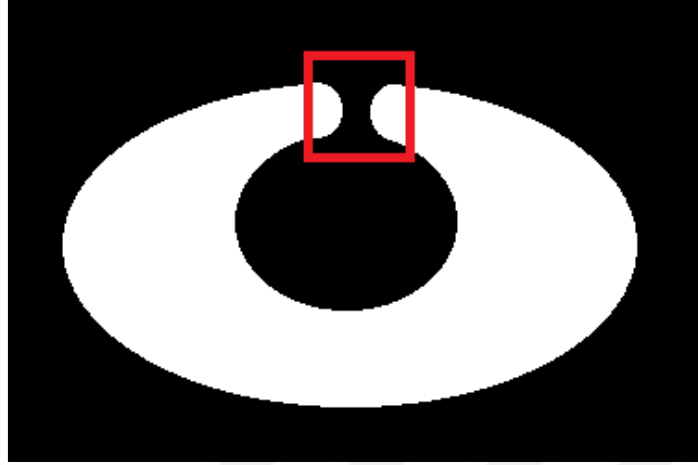
incelip kopma durumunu engellemek için bulunan kritik noktalar ayrılma durumunu simgeleyen kritik noktalar olacaktır. Burada yapılmak istenen ise, modelleme yapılırken, incelleme durumlarındaki bağlantılı bileşenlerin, maksimum belirlenen alana kadar incelmelerini sağlamaktır.



Şekil 5.2 Birleşme ve ayrılma için kritik alan

Şekil 5.2’de görüldüğü üzere iki farklı bağlantılı bileşenin arasında kalan alan birleşme durumu, tek bağlantılı bileşenin kopması düşünülerek konumlandırılan kritik alan ayrılma durumu olarak nitelendirilmiştir. Bu alanlar, ele alınan lokal parça boyutlarına göre değişiklik göstererek genişleyebilir ya da daralabilir.

Faz alan gelişiminde yerel topolojinin korunması da dikkate alınmalıdır. Bağlantılı bileşen sayısı aynı kalsa bile yerel topoloji değişebilir. Şekil 5.3’te, lokal topolojideki bir değişikliğin (birleşmeden dolayı) görüntüdeki bileşenlerin sayısını değiştirmeyeceği kritik bir bölgeyi göstermektedir.



Şekil 5.3 Birleşmeden dolayı kaynaklanan yerel topoloji değişimi kısıt alanı

Bu nedenle, hem bileşen sayısını hem de diğer işlemleri yerel olarak gerçekleştiriyoruz. Lokal parçalar olarak kritik alan hesaplama yöntemleri tezimizi destekleyici bir nitelik olmuştur. Bu, hesaplama süresini arttırır ancak faz alanı gelişimine gömüldüğünde algoritma geliştirilebilir.

5.3 Bağlantılı bileşen sayılarına göre kritik alan hesaplaması

Bu aşamada görüntü üzerinde kritik alan hesaplaması yapılırken belirttiğimiz üzere lokal matrisler kullanarak işleme devam edilir. Lokal matrisler içerisinde birden fazla bağlantılı bileşen olan alanları kısıtlamak öncelikli düşüncemizdir. Lokal bölgelerde bu uygulamayı tüm görüntü için yaptığımızda ise bütün yüzeylerdeki kısıtlamalara ulaşılır. Bağlantılı bileşen sayılarına göre kritik alanlar bulunurken görüntünün üzerindeki tüm pikselleri merkez kabul eden $n \times n$ 'lik lokal matrisler alınır. Matrislerin büyüklüğü yüzey kalınlığına göre belirlenir ya da ne kadarlık bir alana kısıtlama yapılmak istendiğine bağlı olarak ayarlanabilir. Lokal matris pikseller üzerinde gezinir. Her bir lokal bölge

matrisi için bağlantılı bileşenler (connected-component) kontrol edilir ve farklı bileşenlere etiket numaraları atanır. Bağlantılı bileşenlere etiket numarası ataması “two pass” algoritması yardımıyla yapılmaktadır. Etiket numaraları bir dizi içine yerleştirilir. Dizinin eleman sayısına göre birden fazla bağlantılı bileşen olup olmama durumu incelenir. Birden fazla bileşen olması durumunda seçilen nokta, kritik nokta olarak kabul edilir ve diğer noktalardan ayıracak kritiklik etiketi atanır. Fonksiyonun sözde kodu aşağıdaki gibidir.

Kritik_Nokta_Bul fonksiyonu

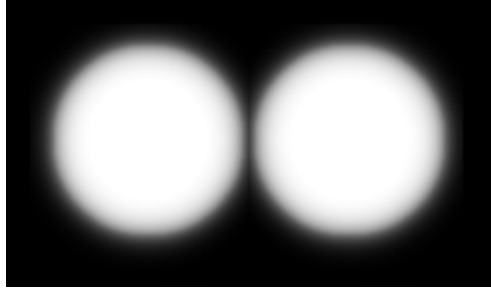
```

for (each row in image)
  for (each column in image)
    Two_Pass (local_matrix) // run two pass algorithm on each local window
    if connected_component_count>1
      Assign critical point

```

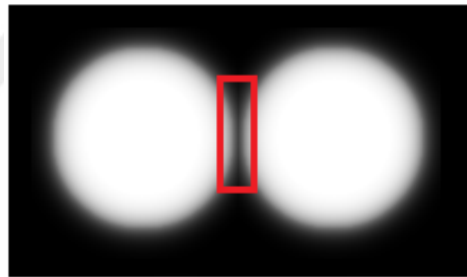
5.3.1 Bağlantılı bileşen sayılarına göre kritik alan uygulaması

Bağlantılı bileşen sayılarına göre kritik alan hesaplaması yapılırken lokal bölge matrislerinden yararlanacağımızdan bahsettik. Bu durumu bir örnekle daha anlaşılır hale getirelim. Öncelikle birleşme durumu üzerinde duralım ve aşağıda da gösterildiği üzere örnek bir resim üzerinden ilerleyelim.



Şekil 5.4 Birleşme durumu için örnek görüntü

Şekil 5.4'te görüldüğü üzere görüntü üzerinde iki farklı bağlantılı bileşen bulunmaktadır. Aslında bu görüntüde kısıtlama alanının hangi bölgede olacağı açıkça görülmektedir. İşlemler sonucunda hedeflediğimiz kritik bölge aşağıdaki şekilde gösterilmiştir.



Şekil 5.5 Birleşme durumu kritik bölge

İşlem adımlarını tanımlayarak ele alınan lokal bölgeler ile nasıl bir kritik alan oluşacağını inceleyelim. Gri seviyeli görüntü, öncelikle siyah beyaz konuma getirip siyahlık ve beyazlık durumlarına göre piksellere 0 – 1 ataması yapılır. Gri seviyeli resimlerde belirli bir kanal değeri seçilip o değer altı ve üstü şeklinde 0-1 atamaları yapılmaktadır. Örneğin RGB si 128 ve altındakiler 0, yani beyaz, 128 üstündekiler 1, yani siyah olarak kabul edilip işlemlere devam edilir. RGB eşik değeri, ele aldığımız görüntüye göre değişkenlik gösterebilmektedir. Ele alınan görüntüde, iki farklı

bağlantılı bileşen arasından seçtiğimiz herhangi bir (i, j) noktasını merkez kabul eden 11×11 'lik lokal bölge matrisi aşağıdaki gibi olsun.

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

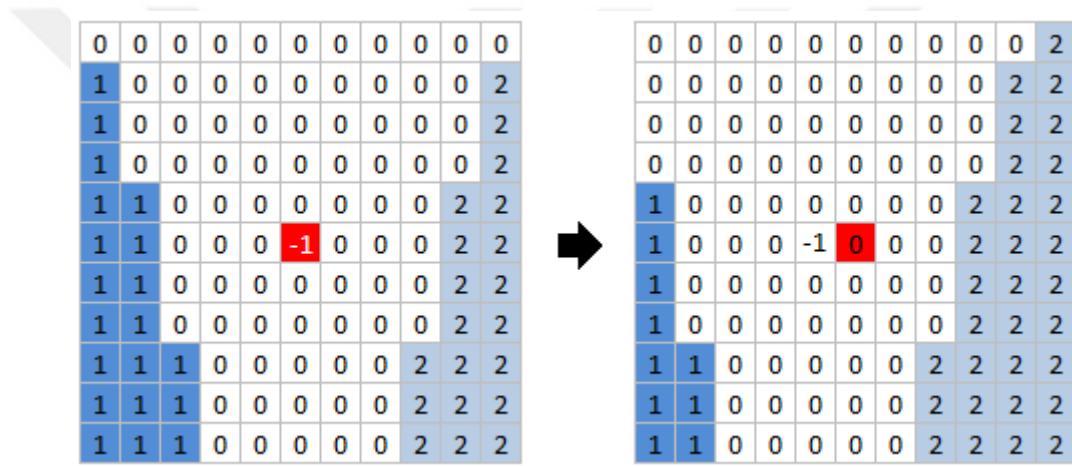
Şekil 5.6 Herhangi bir (i, j) noktası için 11×11 'lik lokal bölge matris

Burada seçtiğimiz nokta kırmızı ile belirtilen noktadır. Şekil 5.6 ise ele aldığımız noktayı merkez kabul eden 11×11 'lik lokal bölge matrisidir. Seçilen noktanın kritik olup olmama durumunu, yine seçilen lokal matriste birden fazla bağlantılı bileşen olup olmama durumunu inceleyerek anlarız. Bu sebeple lokal bölge matrisine two pass algoritması uygulanır ve birden fazla bağlantılı bileşen olma durumu incelenir.

0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	0	0	0	2
1	1	0	0	0	0	0	0	0	2	2
1	1	0	0	0	0	0	0	0	2	2
1	1	0	0	0	0	0	0	0	2	2
1	1	0	0	0	0	0	0	0	2	2
1	1	1	0	0	0	0	0	2	2	2
1	1	1	0	0	0	0	0	2	2	2
1	1	1	0	0	0	0	0	2	2	2

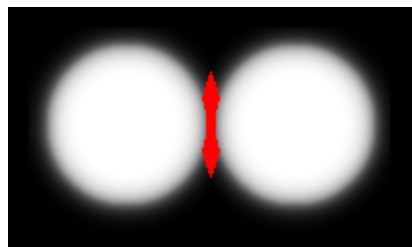
Şekil 5.7 Şekil 5.6'nın two pass uygulanmış şekli

Şekil 5.6'daki lokal bölge matrisine two pass uygulandığında Şekil 5.7'deki gibi bir matris elde edilir. Farklı bağlantılı bileşen numaraları bir diziye alınır (dizi = {1,2}). Dizinin eleman sayısı 1'den büyük olma şartını sağlıyor ise o nokta kritik nokta olacaktır. Seçilen lokal bölge matrisinde birden fazla bileşen olduğu açıktır ve seçilen i, j noktası kritik nokta kabul edilir. Bu noktayı diğer noktalardan ayıracak bir kritiklik değeri (-1) ile adlandırılır. Daha sonra bir sonraki noktaya geçilir ve aynı işlemler tekrarlanır.



Şekil 5.8 İşlem adımı

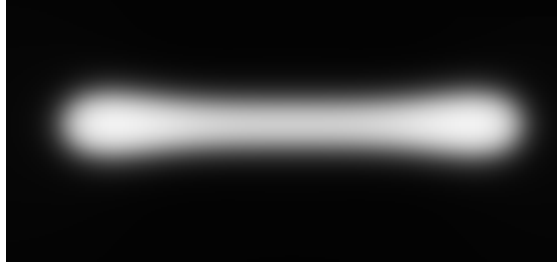
Lokal bölge matrisi tüm piksellere uygulanarak işlem adımları gerçekleştirildiğinde tüm kritik noktalar bulunmaktadır. İlgili işlemlerin sonucunda oluşan görüntü aşağıdaki gibi olmaktadır.



Şekil 5.9 Birleşme durumu için ele alınan örnek görüntünün 11x11'lik lokal bölge matrislerine göre kısıtlanmış şekli

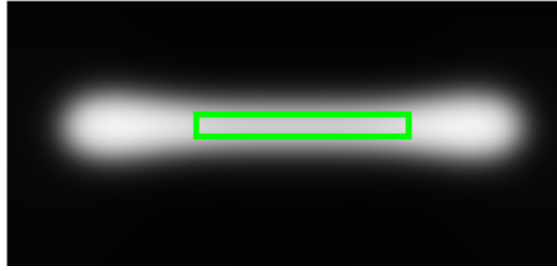
Burada, lokal matrisler yardımı ile bağlantılı bileşen sayılarına göre bulduğumuz tüm kritik noktalar kırmızı bölgeyi oluşturmuştur. Kırmızı bölge, bağlantılı bileşen sayıları ile kısıtladığımız bölge olarak kabul edilmektedir. Resim içerisindeki farklı bağlantılı bileşenlerin değişim durumlarında kısıtladığımız bölgeye kadar değişim gösterebilir ve kısıtladığımız bölgenin ilerisine gidemez.

Ayrılma durumları için ise two pass algoritması uygulanırken, ele alınan bağlantılı bileşen üzerinde işlem yapılır. Birleşme durumunun tersi düşünülür ve uygulama beyaz alanlarda yapılır. Beyaz alan üzerinde lokal bölge matrisleri alınarak two pass algoritması uygulanır. Bu sefer siyah alanlar farklı bileşen olarak algılanır. Birden fazla bağlantılı bileşen olma durumunda seçilen noktayı kritik nokta olarak belirler ve bir kritiklik etiketi atar. Ayrılma ve birleşme durumlarında kritik yüzeyleri ayırt etmek için farklı etiket numaraları atanır.



Şekil 5.10 Ayrılma durumu için örnek görüntü

Şekil 5.10'da görüldüğü gibi incelmış ve kopmak üzere olan bir görüntü mevcuttur. Görüntü üzerinde işlemler sonunda bulmayı hedeflediğimiz kritik bölge aşağıdaki gibidir.



Şekil 5.11 Ayrılma durumu kritik bölge

Görüntüdeki bağlantılı bileşen üzerinden seçilen herhangi bir pikseli merkez kabul eden 11x11'lik lokal matris aşağıdaki gibi olsun.

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

Şekil 5.12 Herhangi bir (i, j) noktası için 11x11'lik lokal bölge matris

Burada seçilen nokta yeşil noktadır. Bu noktanın kritiklik durumu belirlenirken bu sefer 0 noktalarındaki bileşenlere bakılır. Yani farklı bağlantılı bileşenleri ararken 0 noktalarına bakılır. Ele alınan lokal bölge matrisinde farklı bağlantılı bileşen olma durumunu inceleyelim;

3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2

Şekil 5.13 Şekil 5.12'nin two pass uygulanmış şekli

Burada birden fazla bileşen olduğunu görmekteyiz. Bu durumda ele aldığımız nokta kritik noktadır. Birleşme durumunda olduğu gibi ele alınan noktaya da kritiklik etiketi atanır ve bir sonraki noktaya geçilir. İşlemler tüm bileşen üzerindeki noktalara uygulanarak ayrılma durumu için tüm kritik noktalar bulunur.

3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2

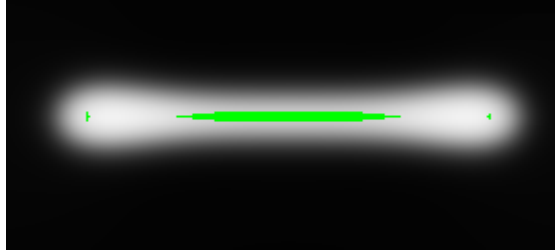


3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	-2	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2

Şekil 5.14 Ayrılma işlem adımı

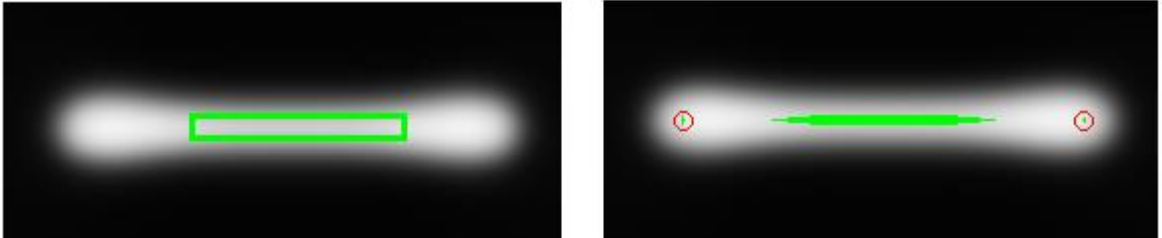
Burada bulunan kritik noktaların birleşiminden ise ayrılma durumu için kritik alanlar oluşturulur. Yüzeyler, inceleme durumlarında bu alandan ötesine geçemez. Bu şekilde yine görüntünün topolojik yapısı korunmuş olur. Birleşme ve ayrılma durumlarındaki

kritik yüzeyleri ayırt etme amacıyla kritik yüzeylere farklı etiket numaraları atanır. İşlemler tüm yüzeylere uygulandığında oluşan görüntü aşağıdaki gibi olmaktadır.



Şekil 5.15 Ayrılma durumu için ele alınan örnek görüntünün 11x11'lik lokal bölge matrislerine göre kısıtlanmış şekli

Ele alınan görüntülerde bağlantılı bileşen sayılarına göre kritik alan hesaplamaları uyguladığında birbirine uzak ve karşılıklı olmayan bölgeler için de kritik nokta belirlendiğini görürüz.

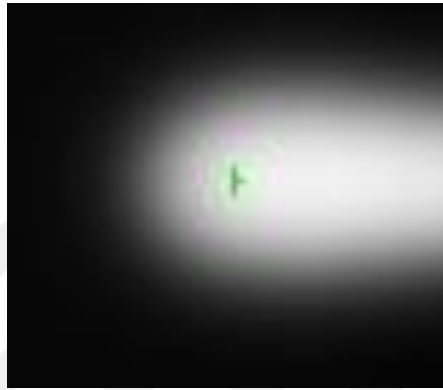


Şekil 5.16 Hedeflenen ve hesaplanan kritik bölge

Şekil 5.16'da örnek görüntü üzerinden hedeflediğimiz kritik bölge ve hesapladığımız kritik alanları gösteren görüntüler yer almaktadır. Sağ taraftaki görüntüde görüldüğü üzere hedeflediğimiz bölgenin dışında da kritik alan hesaplaması yapılmıştır. Bunun nedeni ise lokal pencerelerde karşılıklı olmayan birden fazla bağlantılı bileşen bulmasıdır. Birleşme durumları için ele aldığımız örneklerde de bu durum gözlemlenmektedir. Birleşme ve ayrılma için ele aldığımız örnek görüntülerin çıktılarından alınan birer parça (Şekil 5.9 ve Şekil 5.15) aşağıdaki gibidir.

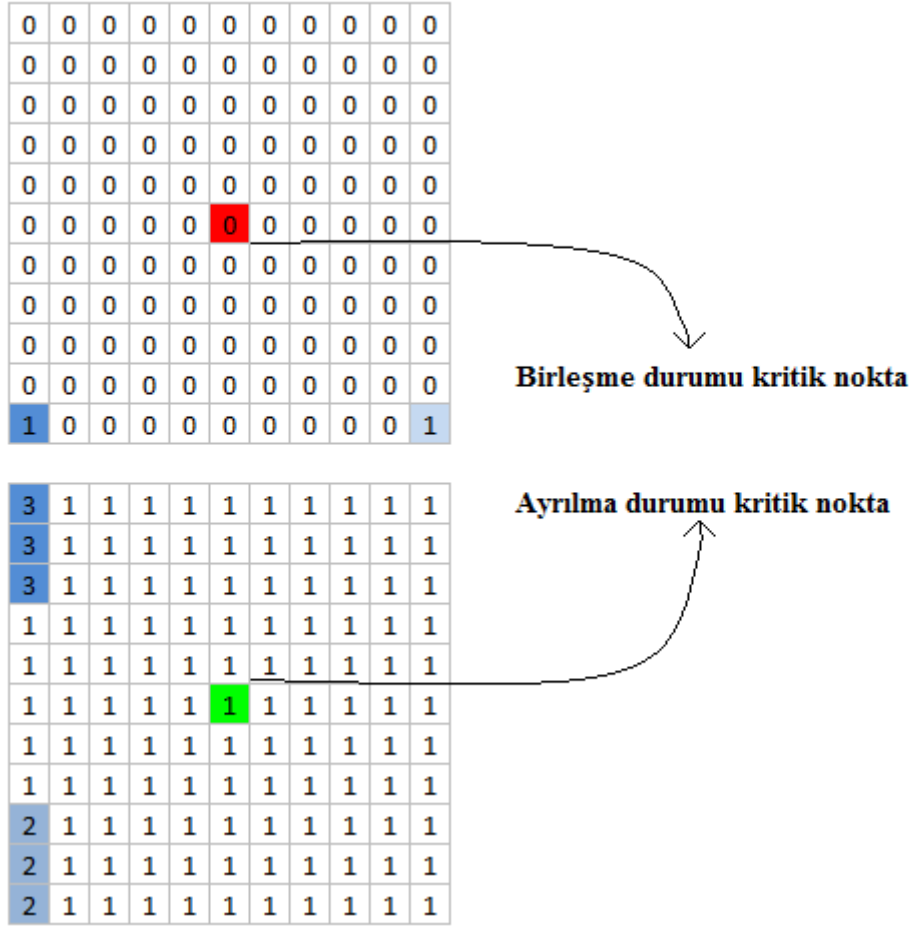


Şekil 5.17 Şekil 5.9'dan alınan bir parça



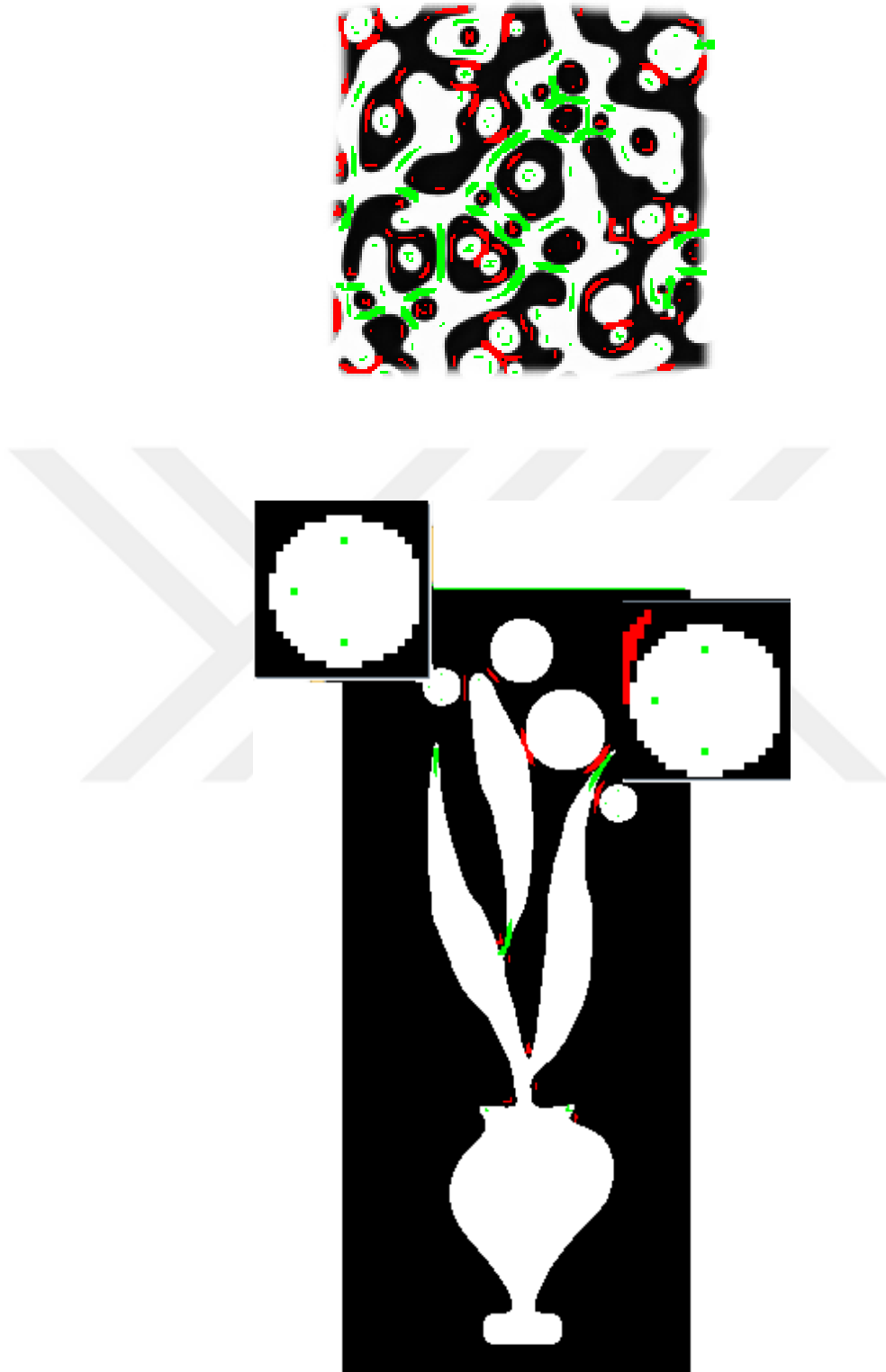
Şekil 5.18 Şekil 5.15'ten alınan bir parça

Şekil 5.17 ve Şekil 5.18'de görüldüğü üzere karşılıklı olmayan noktalar için de kritik noktalar ve kritik alanlar belirlenmiş durumda. Yüzeylerin bu noktalara kadar genişleyip veya incelmesi, yapmak istediğimiz düşünceye ters bir durum değildir. Ele alınan kritik noktaların lokal bölge matrislerinde bu durumu daha detaylı görebiliyoruz.



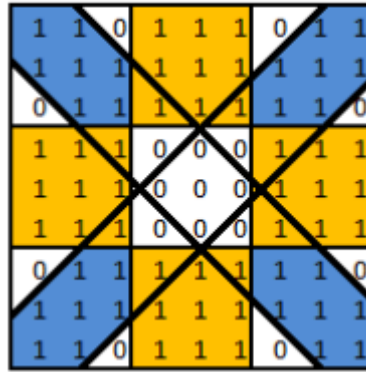
Şekil 5.19 Kritik alanlardan alınan noktalar için lokal matrisler

Şekil 5.19'da da görüldüğü üzere ele alınan lokal bölge matrisinde birbirine uzak ve karşılıklı olmayan bağlantılı bileşenler yer almaktadır. Biz bu durumu önermiyoruz. Farklı ve daha karmaşık yüzeylerde bu durum ayrıca istemediğimiz sonuçlar doğurmaktadır.



Şekil 5.20 Birleşme ve ayrılma için kritik alanlar

karşılıklı bölgelerde olma koşulunu sağlamasını şart koşar. Buna bağlı olarak maskları sadece dört yönde kullanmanın yeterli olduğunu gördük. Bu yaklaşım, her bir pikseldeki eğrilik ve uzaklık hesaplamaları olmaksızın kritik bölgeleri belirler. Yine ara yüzey kalınlığına göre pencere boyutu ayarlanabilir.



Şekil 5.22 Dört yönlü mask

Mask ile kritik alan hesaplamaları aşağıdaki şekilde kurgulanmıştır. Öncelikle tüm pikseller üzerinde her bir piksel için $n \times n$ lik lokal parçalar olarak işleme devam etmektedir. Two pass uygulanan bu lokal parçalar ile yine $n \times n$ lik masklar ile kesiştirilerek birden fazla karşılıklı farklı bağlantılı bileşen olma durumu saptanarak seçilen noktaya kritiklik değeri atanmaktadır. Fonksiyonun sözde kodu aşağıdaki gibidir.

Kritik_Nokta_Bul_fonksiyonu

```
for (each row in image)
```

```
  for (each column in image)
```

```
    Two_Pass (local_matrix) // run two pass algorithm on each local window
```

```
      for (each Mask in local matrix)
```

```
        if (Mask value==1)
```

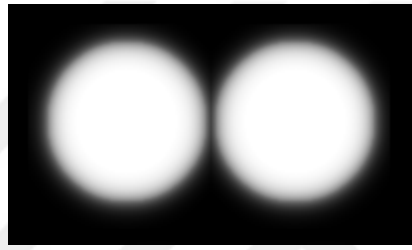
```
          if (new label found)
```

increment label count

if label_count>1

Assign critical point

Mask ile kritik alan hesaplamasını bir örnekle daha anlaşılır hale getirelim. Yine öncelikle birleşme durumu üzerinde duralım ve lokal bölge uygulamasında ele aldığımız örnek görüntü üzerinden ilerleyelim. Sonrasında ise mask yöntemi ile farklarını tartışalım.



Şekil 5.23 Birleşme örnek görüntü

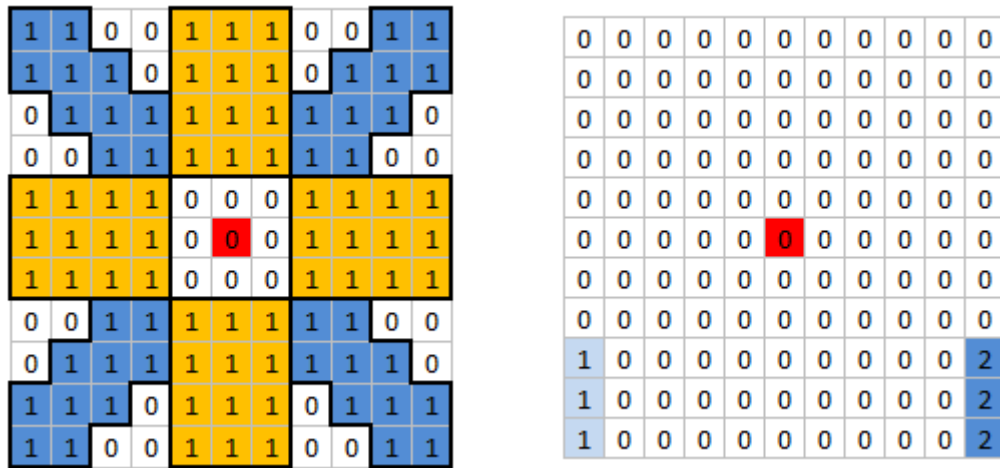
Görüntü üzerinde 0-1 etiketlemesi yaptıktan sonra yine görüntü üzerinden alınan herhangi bir (i, j) noktasını merkez kabul eden 11x11'lik lokal bölge matrisi aşağıdaki gibi olsun. Burada ele alınan nokta kırmızı ile belirtilen noktadır ve lokal pencereye two pass algoritması uygulanmış şekli aşağıdaki gibidir.

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

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	0	0	0	2

Şekil 5.24 Herhangi bir (i, j) noktası için 11x11'lik lokal bölge matris ve two pass uygulanmış şekli

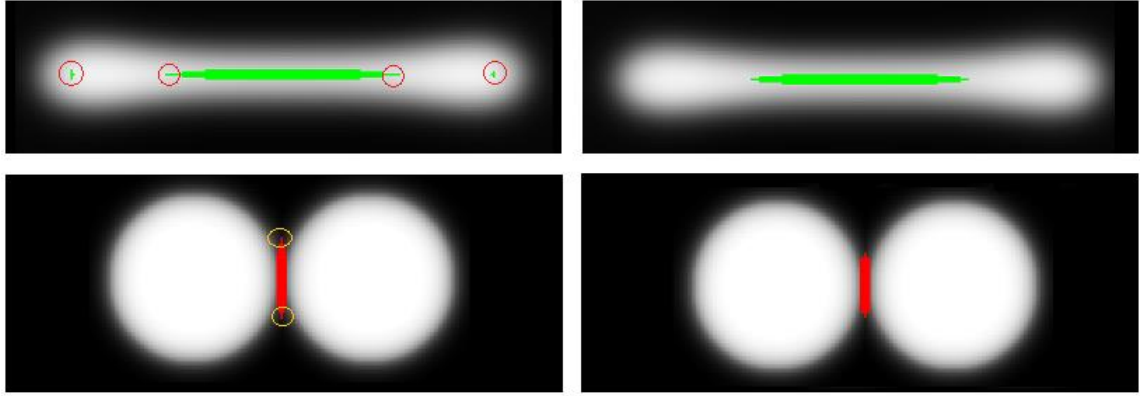
Two pass algoritmasını uyguladıktan sonra birden fazla bağlantılı bileşen olduğunu görüyoruz (dizi = {1,2}). Şu ana kadar ki işleyişimiz lokal bölge ile kritik alan bulma metodundaki adımların aynısıdır ve seçilen nokta bağlantılı bileşen sayısına göre kritik alan bulma yöntemine göre bir kritik noktadır. Şimdi bir de mask yönetimini devreye sokalım ve ele alınan noktayı tekrar inceleyelim. Alan belirlemelerini 11x11'lik bir alanda incelediğimizden ele alınacak mask'ı da 11x11'lik olacak şekilde tanımlamamız gerekmektedir.



Şekil 5.25 Dört yönlü mask ve lokal bölge matrisi

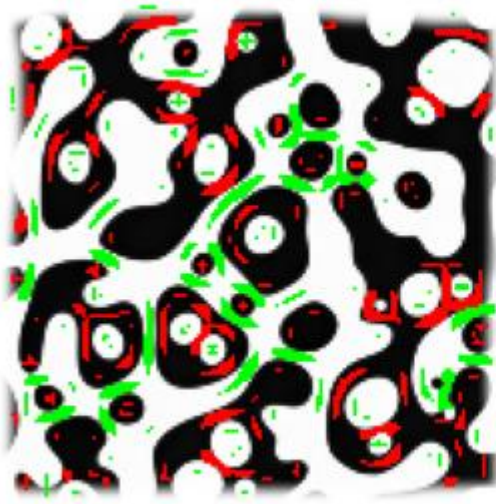
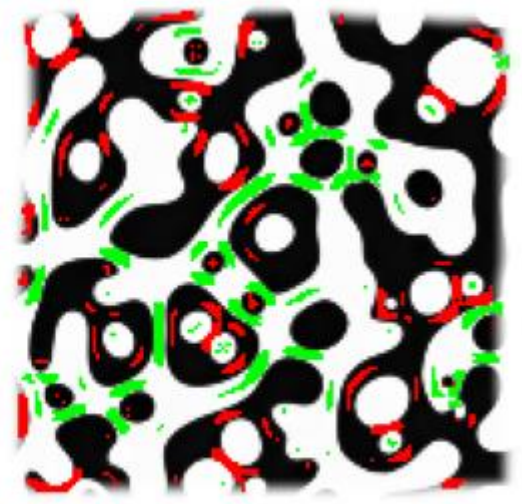
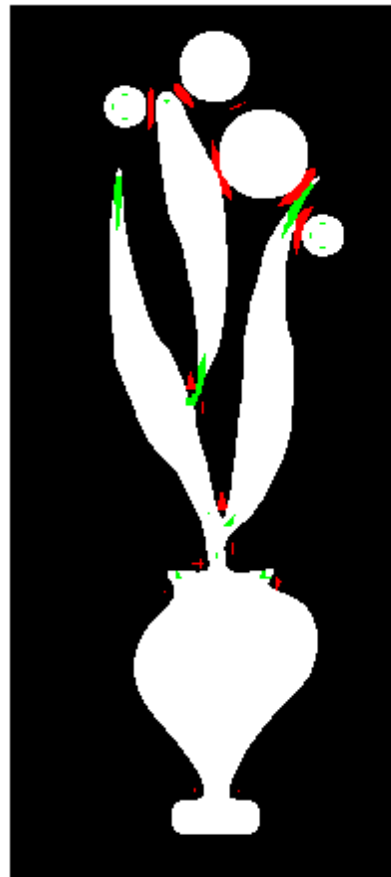
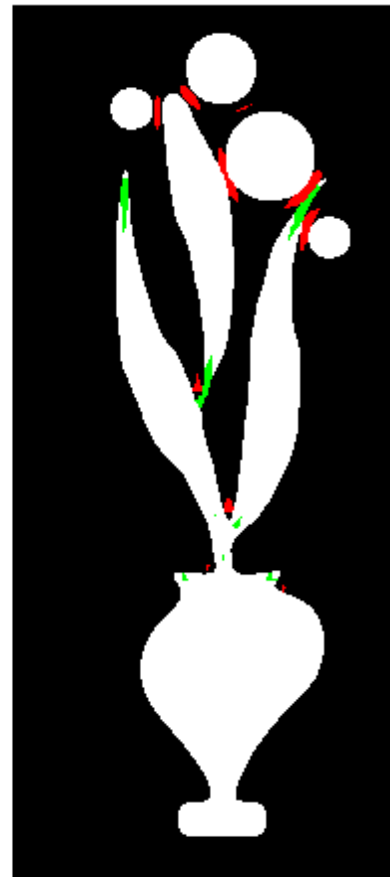
Görüldüğü üzere merkez bölgenin her iki tarafında da aynı anda bulunan farklı bileşenler mevcut değildir. Bu nedenle ele alınan nokta aslında bağlantılı bileşen sayılarına göre yaptığımız hesapta kritik nokta iken mask yöntemine göre kritik nokta değildir. Bu uygulamayı tüm yüzey için uyguladığımızda mask yöntemi ile kritik alanları bulmuş olacağız. Ayrılma durumu için de aynı işlem adımları yapılmaktadır. Tek farkı beyaz alanlar üzerindeki noktalar seçilip two pass algoritması uygulanırken siyah yüzeylere etiket atamaları yapılır ve işleme aynen devam edilir. Sonuç olarak bağlantılı bileşen sayılarına göre kritik alan bulma yöntemi ile hesaplanan her kritik

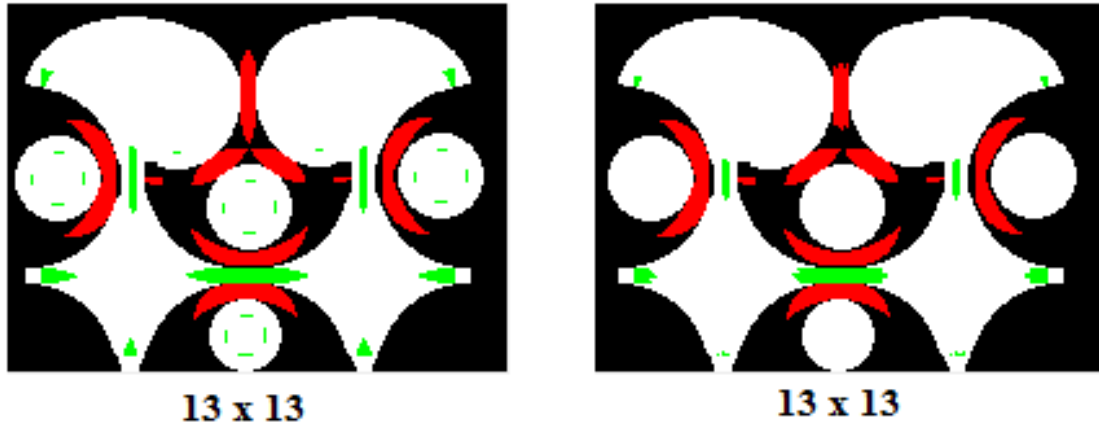
nokta, aslında mask yöntemine göre kritik nokta olmayabilir. Mask yönteminde karşılıklı olma şartını uygulayarak istisnaları yok etmiş oluruz. Şimdi ise ele alınan görüntüde hem bağlantılı bileşen sayılarına göre hem de mask yöntemi ile kritik alan hesabı yaparak farklılıkları inceleyelim.



Şekil 5.26 Birleşme ve ayrılma durumları için bağlantılı bileşen sayılarına göre (sol) ve mask yöntemine göre (sağ) kritik alanlar

Her iki yöntem için kritik alanları hesapladığımızda yukarıdaki görüntüleri elde etmekteyiz. Şekil 5.26'da sol tarafta bağlantılı bileşen sayılarına göre birleşme ve ayrılma durumunu belirten 11x11'lik lokal bölgelere göre hesaplanan kritik alanlar sağ tarafta ise mask yöntemine göre birleşme ve ayrılma durumunu belirten 11x11'lik lokal parçalara göre hesaplanan kritik alanlar mevcuttur. Mask yöntemi ile karşılıklı olmayan noktaların kritik nokta olmasını engellemiş olduk. Bu noktaları elimine ederek istediğimiz şekilde kritik alanları oluşturmuş durumdayız. Daha karmaşık yüzeylerde iki yöntem arasındaki farkı daha net görebiliriz.

 7×7  7×7  9×9  9×9



Şekil 5.27 Bağlantılı bileşen sayılarına göre (sol) ve mask yöntemine göre (sağ) kritik alanlar kıyaslaması

5.5 Uzaklık filtresine dayalı kritik alan hesaplamaları

Lokal yüzeylerde kritik alan hesaplamalarında mask yöntemi ile istediğimiz sonuçları elde edebildik. Diğer taraftan kritik alanların derecelendirilmesi ve modelleme yapılırken kritiklik derecesi de baz alınarak kısıtlama yapılması bir başka bakış açımız olmuştur. Bu nedenle, uzaklık fonksiyonlarını kullanarak, bu şekilde kritiklik derecesi de belirleyerek kritik alan hesaplamasını hedefledik. Bu sayede kritik alan hesaplanırken aynı zamanda kritiklik derecesi de atanarak kritik alan hesabı yapılmaktadır. Bu şekilde bulmuş olduğumuz kritik alanlarda kritiklik derecesine göre kısıtlama yaparak hedefimizi bir üst seviyeye taşımış olmaktadır.

Uzaklık filtresine dayalı kritik alan hesaplama yöntemi genel olarak şu şekilde işlemektedir. Resmin her bir pikseli için $n \times n$ 'lik lokal pencereler alınır. Lokal pencereler içerisinde two pass algoritması uygulanır ve birden fazla bağlantılı bileşen olması durumu incelenir. Birden fazla bağlantılı bileşen bulunması durumunda, farklı bölgeler için karşılıklı en yakın uzaklıklar belirlenip bu uzaklıkların en küçük olanı göz

önünde bulundurulur ve kritiklik etiketi atanır. Kritiklik etiketi atanırken uzaklık değerleri kullanılarak kritiklik derecesi belirlenir. İki bağlantılı bileşen arasında kalan orta alan, en kritik alan olarak değerlendirilir ve bağlantılı bileşenlere yaklaştıkça kritiklik seviyesi azalır. Uygulamanın sözde kodu aşağıdaki gibidir.

Kritik_Nokta_Bul fonksiyonu

```

for(each row i in image)
  for(each column j in image)
    lokalMatrisBirlesme = Lokal_Pencere_Al (birleşme, i,j)
    lokalMatrisAyrılma = Lokal_Pencere_Al (ayrılma, i,j)
    if (image value==0)
      etiketMatrisBirlesme = twoPass(lokalMatrisBirlesme)
      değer = KritiklikDerecesiBul(etiketMatrisBirlesme)
      if( değer ) // karşılıklı anlamlı toplam uzaklık var ise
        kritiklikMatrisi(i,j)= -değer // negatif değerler birleşme durumunu
    if (image value==1)
      etiketMatrisAyrılma = twoPass(lokalMatrisAyrılma)
      değer=KritiklikDerecesiBul(etiketMatrisAyrılma)
      if(değer) // karşılıklı anlamlı toplam uzaklık var ise
        kritiklikMatrisi(i,j)= değer // pozitif değerler ayrılma durumunu

```

kritiklik_Derecesi_Bul(etiketMatrisi) fonksiyonu

// 8 farklı uzaklık bölgesi için uzaklık değerlerini ve bileşen etiketlerini dizilere sakla

for(each uzaklık bölgesi k)

 for(each row i in etiketMatrisi)

 for(each column j in etiketMatrisi)

 if(etiketMatrisi[i][j]!=0) // eğer bileşen var ise

 UzaklıkEtiketDizisi[k][i].uzaklık=UzaklıkMatrisi[i][j]

 UzaklıkEtiketDizisi[k][i].etiket=EtiketMatrisi[i][j]

//4 farklı karşılıklı uzaklıklar için etiketleri aynı olanları elimine et ve karşılıklı

uzaklık dizilerine yaz

for(each uzaklık bölgesi k) // 8 farklı bölge için. k'yi 2 artır

 for(each uzaklıkEtiketDizisi i)

 if(UzaklıkEtiketDizisi[k][i].etiket== UzaklıkEtiketDizisi[k+1][i].etiket)

 karsilikliUzaklikDizisi[k][i]= UzaklıkEtiketDizisi[k][i].uzaklik

 karsilikliUzaklikDizisi[k+1][i]= UzaklıkEtiketDizisi[k+1][i].uzaklik

//4 farklı karşılıklı uzaklıkların toplamlarını diziye yaz

for(each karsilikliBolge k) // 8 farklı bölge için. k'yi 2 artır

 t= karsilikliUzaklikDizisi[k].uzunluk

 z= karsilikliUzaklikDizisi[k+1].uzunluk

 modDegerDizi={ 100,141,223 }

```

for(each karsilikliUzaklikDizisi[k] t)
    for(each karsilikliUzaklikDizisi[k+1] z )
        for( each ModDegerDizi p) // 3 deger için mod karsilastirmalarini yap
            //Karşılıklı uzunlukları bul ve topla
            if((mod(karsilikliUzaklikDizisi[k][t],
karsilikliUzaklikDizisi[k+1][z])==0
                || mod(karsilikliUzaklikDizisi[k+1][z],
karsilikliUzaklikDizisi[k][t])==0 )
                || (mod(karsilikliUzaklikDizisi[k][t], ModDegerDizi[p])==0
&& mod(karsilikliUzaklikDizisi[k+1][z], ModDegerDizi[p])==0) )
                Toplam= karsilikliUzaklikDizisi[k][t]+karsilikliUzaklikDizisi[k+1][z]
                Break
            EnKucukKarsilikliBolgeDizi[] = Min(Toplam)

//4 farklı karşılıklı uzaklıkların minimum toplamlarını oluşturan 4 farklı karşılıklı
uzaklıkların farkını bul
enKisazaklik = Min(EnKucukKarsilikliBolgeDizi[])
for(each EnKucukKarsilikliBolgeDizi w ) // 4 farklı minumum toplamda dön
    if(EnKucukKarsilikliBolgeDizi[w]== enKisazaklik )
        for(each karsilikliBolge k) // 8 farklı bölge için. k'yi 2 artır
            t= karsilikliUzaklikDizisi[k].uzunluk
            z= karsilikliUzaklikDizisi[k+1].uzunluk
            modDegerDizi={ 100,141,223 }
            maxUzunluk=100000

```

```

for(each karsilikliUzaklikDizisi[k] t)
    if(karsilikliUzaklikDizisi[k][t]< maxUzunluk)
        for(each karsilikliUzaklikDizisi[k+1] z )
            for( each ModDegerDizi p) // 3 deger için mod karsilastirmalarini yap
                // Enkucuktoplamı oluşturan karşılıklı uzaklıkları bul
                if(((mod(karsilikliUzaklikDizisi[k][t],
karsilikliUzaklikDizisi[k+1][z])==0
                || mod(karsilikliUzaklikDizisi[k+1][z],
karsilikliUzaklikDizisi[k][t])==0 )
                || (mod(karsilikliUzaklikDizisi[k][t], ModDegerDizi[p])==0
&& mod(karsilikliUzaklikDizisi[k+1][z], ModDegerDizi[p])==0))
&& karsilikliUzaklikDizisi[k][t]+ karsilikliUzaklikDizisi[k+1][z]==
EnKucukKarsilikliBolgeDizi[w] )
                    KritiklikEtiketi=
                    Mutlak(karsilikliUzaklikDizisi[k][t]- karsilikliUzaklikDizisi[k+1][z])
                    maxUzunluk= EnKucukKarsilikliBolgeDizi[w]
                    Break

```

Öncelikle uzaklık hesabı için aşağıdaki gibi uzaklık filtresi oluşturulur. Uzaklık filtresi ilk etapta 15x15 şeklinde tanımlanmıştır. Lokal parçalar alınarak nxn lik filtreler oluşturabilir. Lokal parçalar diğer metotlarda da olduğu gibi uygulamak istenilen alanlara göre değişkenlik gösterebilir.

987	921	860	806	761	728	708	700	708	728	761	806	860	921	987
921	846	781	720	669	632	608	600	608	632	669	720	781	846	921
860	781	705	640	583	538	509	500	509	538	583	640	705	781	860
806	720	640	564	501	446	412	400	412	446	501	564	640	720	806
761	669	583	501	423	360	316	300	316	360	423	501	583	670	761
728	632	538	446	360	282	223	200	223	282	360	446	538	632	728
708	608	509	412	316	223	141	100	141	223	316	412	509	608	708
700	600	500	400	300	200	100	0	100	200	300	400	500	600	700
708	608	509	412	316	223	141	100	141	223	316	412	509	608	708
728	632	538	446	360	282	223	200	223	282	360	446	538	632	728
761	670	583	501	423	360	316	300	316	360	423	501	583	670	761
806	720	640	564	501	446	412	400	412	446	501	564	640	720	806
860	781	705	640	583	538	509	500	509	538	583	640	705	781	860
921	846	781	720	669	632	608	600	608	632	669	720	781	846	921
987	921	860	806	761	728	708	700	708	728	761	806	860	921	987

Şekil 5.28 Uzaklık matrisi

Yukarıdaki filtreyi lokal parçada uzaklık bulmak için kullanırız. Uzaklıklar 8 ayrı bölgede aynı değerleri içerdiğinden farklı bölgelerdeki uzaklık kıyaslaması için aşağıdaki matrisi kullanırız.

8	8	8	8	8	8	8	3	3	3	3	3	3	3	6
1	8	8	8	8	8	8	3	3	3	3	3	3	6	6
1	1	8	8	8	8	8	3	3	3	3	3	6	6	6
1	1	1	8	8	8	8	3	3	3	3	6	6	6	6
1	1	1	1	8	8	8	3	3	3	6	6	6	6	6
1	1	1	1	1	8	8	3	3	6	6	6	6	6	6
1	1	1	1	1	1	8	3	6	6	6	6	6	6	6
1	1	1	1	1	1	1	0	2	2	2	2	2	2	2
5	5	5	5	5	5	5	4	7	2	2	2	2	2	2
5	5	5	5	5	5	4	4	7	7	2	2	2	2	2
5	5	5	5	5	4	4	4	7	7	7	2	2	2	2
5	5	5	5	4	4	4	4	7	7	7	7	2	2	2
5	5	5	4	4	4	4	4	7	7	7	7	7	2	2
5	5	4	4	4	4	4	4	7	7	7	7	7	7	2
5	4	4	4	4	4	4	4	7	7	7	7	7	7	7

Şekil 5.29 Bölge matrisi

Kritik noktaları belirlerken ki amaç, seçilen bir noktayı merkez kabul eden lokal matriste, en az iki bölgenin içerdiği karşılıklı iki nokta bulunabilmesidir. Bu takdirde

seçilen nokta kritik nokta olacaktır. Kritiklik seviyesi belirlenmesi için seçilen bu noktalar merkeze uzaklığı en kısa ve karşılıklı olması koşulunu sağlamalıdır. Bunu yaparken birden fazla karşılıklı iki nokta bulunabileceğinden bir algoritma geliştirdik. Aşağıda bu algoritmanın açıklamasını yapılmaktadır.

Herhangi bir görüntü içerisinde seçilen her hangi bir piksel için 11x11'lik bir lokal parça aldığımızı düşünelim.

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

Şekil 5.30 11x11'lik örnek görüntü parçası

Burada seçilen nokta kırmızı nokta olup merkez noktadır. Diğer yöntemlerde olduğu gibi ele alınan lokal parçaya two pass algoritması uygulayarak etiketleme yapılır ve topolojik bileşenlere ayrılır. Elde edilen sonuç aşağıdaki gibi olacaktır.

0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	3	3	3
2	2	0	0	0	0	0	0	3	3	3
2	2	0	0	0	0	0	0	3	3	3
2	2	0	0	0	0	0	0	3	3	3
2	2	0	0	0	0	0	0	3	3	3
2	2	0	0	0	0	0	0	3	3	3
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	4	0	0	0
0	0	0	4	4	4	0	0	0	0	0

Şekil 5.31 Şekil 5.30'un two pass uygulanmış şekli

Bu parçada birden fazla karşılıklı farklı bölge olduğunu görebiliyoruz(Dizi={1,2,3,4}).

Bu nedenle ele alınan nokta aslında kritik bir noktadır. Şimdi bu noktanın nasıl kritik olduğunu ve derecelendirmesini daha açık şekilde anlatalım.

Ele alınan noktanın kritiklik durumunu ve derecesini belirlemek için yukarıda tanımladığımız uzaklık ve bölge matrislerinden yararlanacağız. Alınan lokal bölge matrisi 11x11'lik olduğundan uzaklık ve bölge matrisi de 11x11'lik olmalıdır.

8	8	8	8	8	3	3	3	3	3	6
1	8	8	8	8	3	3	3	3	6	6
1	1	8	8	8	3	3	3	6	6	6
1	1	1	8	8	3	3	6	6	6	6
1	1	1	1	8	3	6	6	6	6	6
1	1	1	1	1	0	2	2	2	2	2
5	5	5	5	5	4	7	2	2	2	2
5	5	5	5	4	4	7	7	2	2	2
5	5	5	4	4	4	7	7	7	2	2
5	5	4	4	4	4	7	7	7	7	2
5	4	4	4	4	4	7	7	7	7	7

705	640	583	538	509	500	509	538	583	640	705
640	564	501	446	412	400	412	446	501	564	640
583	501	423	360	316	300	316	360	423	501	583
538	446	360	282	223	200	223	282	360	446	538
509	412	316	223	141	100	141	223	316	412	509
500	400	300	200	100	0	100	200	300	400	500
509	412	316	223	141	100	141	223	316	412	509
538	446	360	282	223	200	223	282	360	446	538
583	501	423	360	316	300	316	360	423	501	583
640	564	501	446	412	400	412	446	501	564	640
705	640	583	538	509	500	509	538	583	640	705

Şekil 5.32 11x11'lik uzaklık ve bölge matrisi

Bölüm ve uzaklık matrislerinden de 11x11'lik parçalar alarak ana parça ile keşitiriyoruz. Bu durumda asıl lokal matris, bölüm matrisi ve uzaklık matrisi aşağıdaki gibi olmaktadır.

0	0	0	3	3	3	3	3	0	0	0
0	0	0	0	0	3	3	3	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	2	2	2
1	1	0	0	0	0	0	0	2	2	2
1	1	0	0	0	0	0	0	2	2	2
1	1	0	0	0	0	0	0	2	2	2
1	1	0	0	0	0	0	0	2	2	2
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	4	0	0
0	0	0	4	4	4	0	0	0	0	0

0	0	0	8	8	3	3	3	0	0	0
0	0	0	0	0	3	3	3	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	6	6	6
1	1	0	0	0	0	0	0	6	6	6
1	1	0	0	0	0	0	0	2	2	2
5	5	0	0	0	0	0	0	2	2	2
5	5	0	0	0	0	0	0	2	2	2
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	7	7	0	0	0
0	0	0	4	4	4	0	0	0	0	0

0	0	0	538	509	500	509	538	0	0	0
0	0	0	0	0	400	412	446	0	0	0
0	0	0	0	0	0	0	0	0	0	0
538	446	0	0	0	0	0	0	360	446	538
509	412	0	0	0	0	0	0	316	412	509
500	400	0	0	0	0	0	0	300	400	500
509	412	0	0	0	0	0	0	316	412	509
538	446	0	0	0	0	0	0	360	446	538
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	412	446	0	0
0	0	0	538	509	500	0	0	0	0	0

Şekil 5.33 Ele alınan lokal matris için uzaklık ve bölge matrisi

Bölge matrisine göre karşılıklı olan bölgeler 1-2, 3-4, 5-6, 7-8 şeklindedir. Seçilen noktanın tam karşısında olma koşulunu bulmak için aramalarımızı bu şarta göre yapmamız gerekmektedir ve aynı zamanda uzaklıkların birbiri arasında birbirinin katları olma koşulu aranmaktadır. Bunun sebebi ise aralarında 180 derece olacak şekilde karşılıklı olmasını anlamaktır. Şimdi tüm bölgelerdeki uzaklıkları dizi şeklinde

oluşturalım ve tam karşılarında kurallarımıza uyan bir nokta olup olmadığını araştıralım. Burada yapacağımız karşılıklı noktaların uzaklıklarını toplamak ve en küçük toplamı veren karşılıklı noktaların mutlak farklarını alarak kritik noktanın kritiklik derecesini belirlemek olacaktır.

1.Bölge dizi = [400,412,446,500,509,538]

2.Bölge dizi = [300,316,360,400,412,446, 500,509,538]

* 1-2 kıyas dizi = [700] en küçük toplam

3.Bölge dizi = [400,412,446,500,509,538]

4.Bölge dizi = [500,509,538]

* 3-4 kıyas dizi = [900] en küçük toplam

5.Bölge dizi = [412,446,509,538]

6.Bölge dizi = [316,360, 412,446,509,538]

* 5-6 kıyas dizi = [824] en küçük toplam

7.Bölge dizi = [412,446]

8.Bölge dizi = [509,538]

* 7-8 kıyas dizi = [] karşılıklı nokta olmadığı için bunu dikkate almıyoruz []

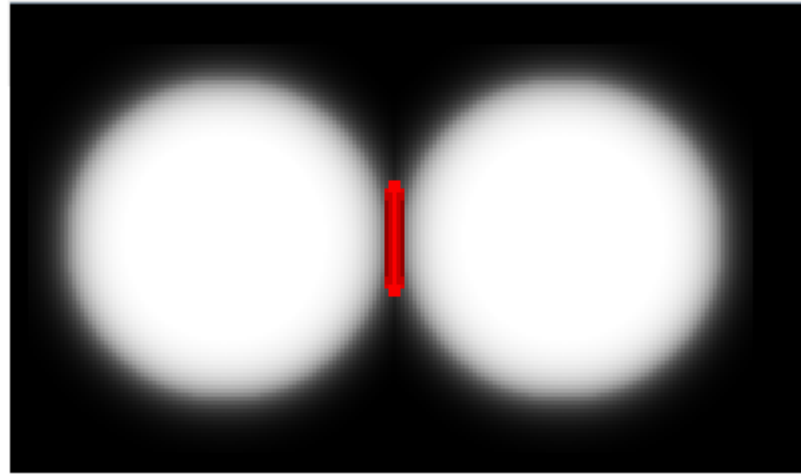
Sonuç olarak karşılıklı noktaların en küçük toplamlarından elde ettiğimiz dizi [700,900,824] şeklindedir.

Bu dizinin de en küçük elemanı 700 olduğundan 1. ve 2. bölgelerindeki karşılıklı noktaları inceliyoruz. Sonuç olarak 1. ve 2. bölgedeki karşılıklı uzaklıkları 300 ve 400 olan iki noktadan elde ettiğimizi görüyoruz. Bu noktaların mutlak farkı 100 olduğundan kritik noktanın derecesini 100 olarak belirliyoruz. Bunu tüm noktalar için yaptığımızda ise kritik noktaların tamamını belirlemekle birlikte aynı zamanda derecelerini de belirlemiş oluyoruz. Karşılıklı iki uzaklık farkının 0 olduğu yerlerde sabit “80” değerini kullanmaktayız. Tüm kritik değerlerde birleşme durumu için negatif değerler kullanırken ayrılma durumu için pozitif değerler kullanmaktayız. Bunun sebebi her iki durumu da ayrı ayrı analiz edebilmek içindir.

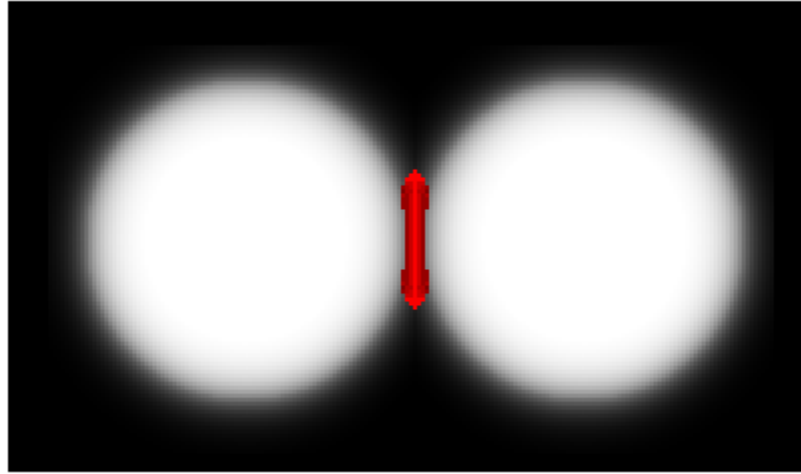
1	1	-446	-400	-200	-80	-200	-400	-446	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	1	-400	-200	-80	-200	-400	1	1	1
1	1	-446	-400	-200	-80	-200	-400	-446	1	1

Şekil 5.34 Birleşme durumu için derecelendirilmiş kritik noktalara örnek

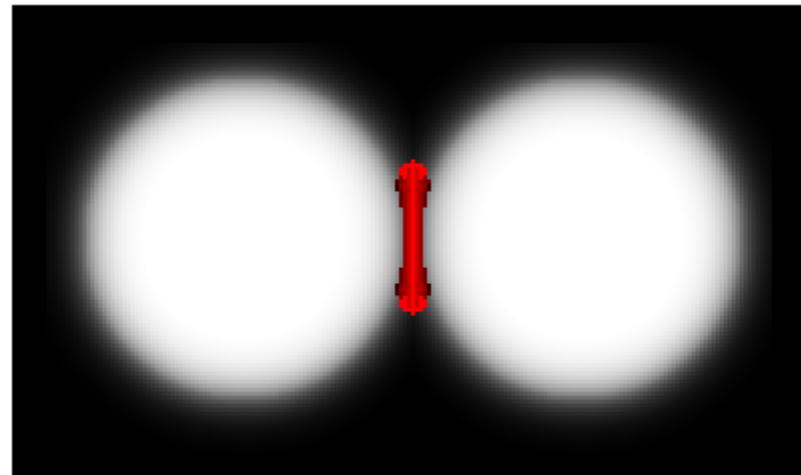
Şekil 5.4'teki görüntü için 11x11 13x13 ve 15x15'lik lokal parçalar olarak uygulamamızı çalıştıralım ve sonuçları analiz edelim.



11x11 çıktısı



13x13 çıktısı



15x15 çıktısı

Şekil 5.35 Birleşme durumu için uzaklık filtresine dayalı farklı pencere boyutlarına göre kritik alanlar

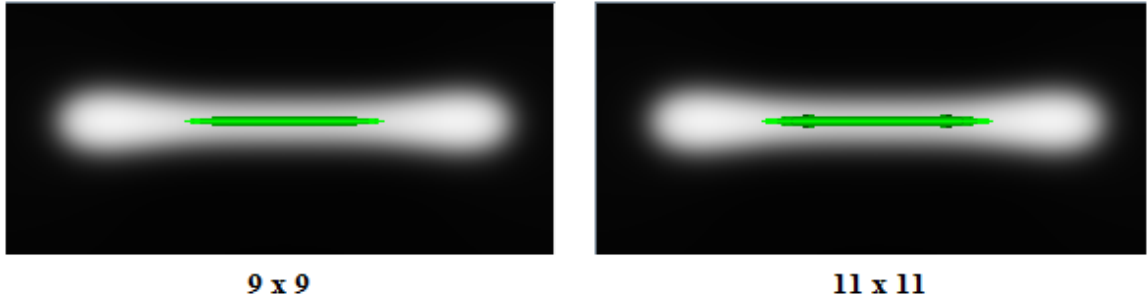
Ele aldığımız üç örnekte kritik alanları farklı lokal bölgelere göre hesaplattık ve de üç farklı çıktı aldık. Burada da görüldüğü üzere kritik alan ele alınan lokal bölgelere göre değişkenlik göstermiş ve de uygulama yaptığımız pencere büyüdükçe kritik alan genişlemiştir. Her üç örnekte de kritik alanın en orta noktası aynı kritiklik derecesine sahip olup, kritiklik derecesi ise bağlantılı bileşenlere yaklaştıkça azalmıştır. Buradaki amacımız daha karmaşık yüzeylerde ele aldığımız $n \times n$ 'lik pencere haricinde kritiklik derecesini de ele alarak gerekli kısıtlamaları uyguluyor olabilmemizdir.

Ayrılma durumu için de işlem adımlarımız aynen geçerlidir. Kritiklik derecelerini, ayrılma ve birleşme olarak ayırt etme amacı ile ayrılma durumları için pozitif değer atanmaktadır.

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
400	400	400	400	400	400	400	400	400	400	400	400	400	400
200	200	200	200	200	200	200	200	200	200	200	200	200	200
80	80	80	80	80	80	80	80	80	80	80	80	80	80
200	200	200	200	200	200	200	200	200	200	200	200	200	200
400	400	400	400	400	400	400	400	400	400	400	400	400	400
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

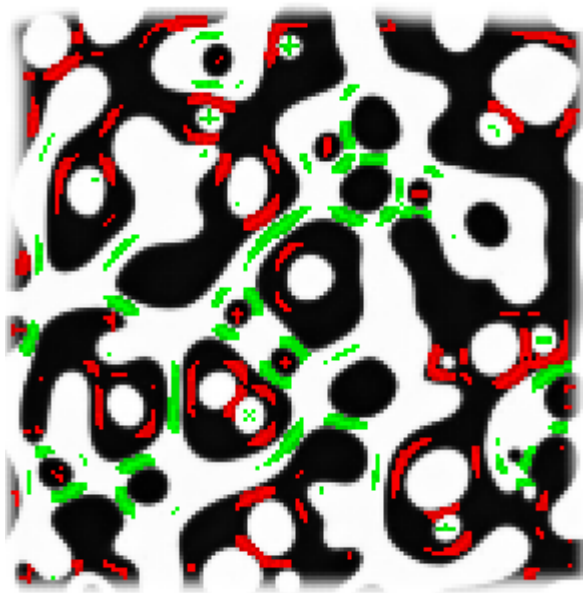
Şekil 5.36 Ayrılma durumu için derecelendirilmiş kritik noktalara örnek

Şekil 5.10'daki görüntü için 9x9 ve 11x11'lik lokal parçalar alarak uygulamamızı çalıştıralım ve sonuçları analiz edelim.

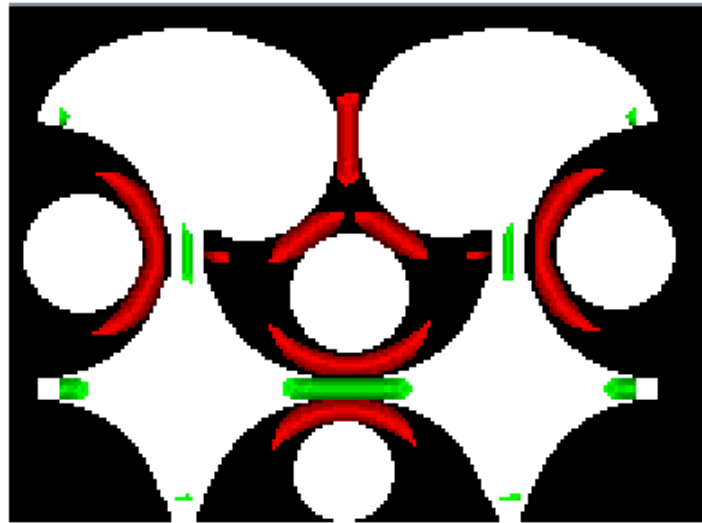


Şekil 5.37 Ayrılma durumu için derecelendirilmiş kritik alan örnekleri

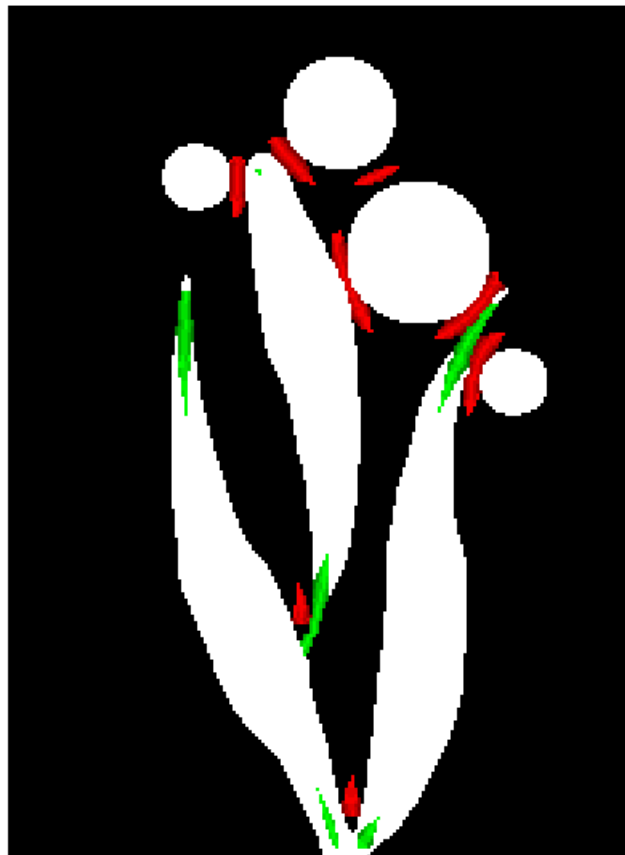
Yukarıdaki örnek çıktılarına bakıldığında birleşme durumundaki sonuçları yine aynen elde ettik. Ele aldığımız lokal pencereyi büyüttükçe kritik alanın genişlediğini görmekteyiz. Kritiklik derecesinin ise bağlantılı bileşenlere yaklaştıkça azaldığını görmekteyiz. Daha karmaşık yüzeylerde birleşme ve ayrılma durumları için derecelendirilmiş kritik alanlar farklı görüntüler ele alınarak aşağıda sunulmuştur.



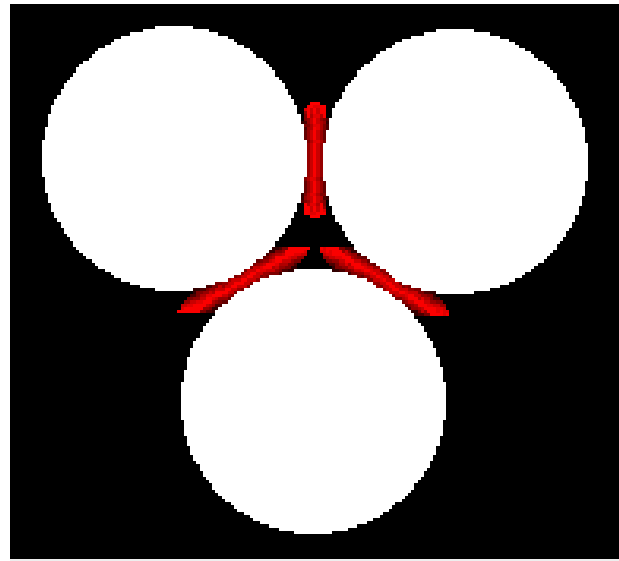
7 x 7



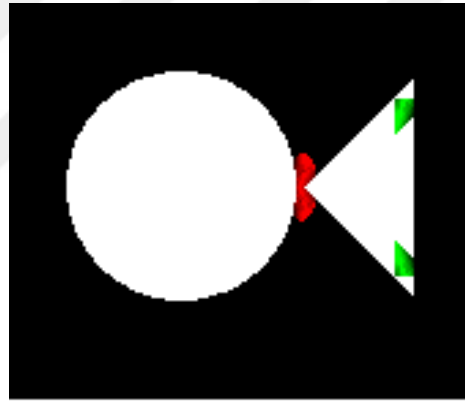
13 x 13



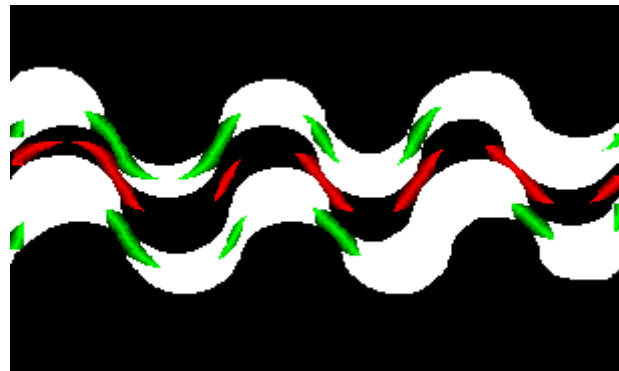
11 x 11



15x15



15x15



Şekil 5.38 Karmaşık yüzeyler için uzaklık filtresine dayalı farklı pencere boyutlarına göre kritik alanlar

VI. SONUÇ VE GELECEK ÇALIŞMALAR

Görüntü modellemelerinde önemli bir araç olan faz alan denklemlerini kullanırken topolojik evrimin sınırlandırılması öncelikli düşüncemiz olmuştur. Bununla birlikte dijital topolojideki basit olmayan nokta fikrinden yola çıkarak topolojik yapının bozulmaması adına bileşenler arasında kritik alan hesaplamaları için uygulamalar geliştirdik. Bu hesaplamalar görüntü üzerinden alınan lokal pencereler içerisinde kritik noktalar belirlenerek yapılmaktadır. Lokal pencerelerde kritik noktaları bulurken öncelikle bağlantılı bileşen sayılarına göre hesaplama yaptık. Seçilen herhangi bir pikseli merkez kabul eden lokal pencerede birden fazla bileşen olması durumunda o noktayı kritik kabul edip kritik alanları oluşturduk. Bir sonraki aşamada dört yönlü mask tanımlayıp lokal pencereler ile kesiştirdik. Bu sayede farklı bağlantılı bileşenlerin karşılıklı bölgelerde olma koşulunu yerine getirerek kritik alanlarımızı daha da sağlamlaştırdık. Böylece faz alanı modellemelerinde kritik bölgeleri istediğimiz şekilde ayırmış olduk. Daha sonra bulduğumuz kritik alanların dereceli olması fikrinden yola çıkarak filtreye dayalı kritik alan hesabını geliştirdik. Uzaklık ve bölge matrisleri tanımlayarak lokal pencereler ile kesiştirdik. Bu işlemlerin sonucunda bileşenlerin karşılıklı olma şartını yerine getirip aynı zamanda uzaklık matrisinden de yararlanarak kritik noktanın derecelendirilmesini sağladık. Bu şekilde derecelendirilmiş kritik bölgelere ulaşarak faz alanı ara yüzey dinamiğini oluşturduk.

Çalışmalarımızdan istediğimiz sonuçları hemen hemen elde edebildik. Fakat odaklanmamız gereken birkaç konu bulunmaktadır. Bunlardan bir tanesi

algoritmalarımızı daha da hızlandırabilmektir. Bağlantılı bileşenleri etiketlemek için iki farklı algoritma (özyineleme algoritması, iki geçişli algoritma) araştırıp two pass algoritmasını tercih etmiştik. Bu aşamada diğer bağlantılı bileşen etiketleme yöntemleri de araştırılarak hız kazandırılabilir. Diğer taraftan kritik alan belirlemede uyguladığımız formüller geliştirilebilir. Filtreye dayalı yöntemde kritik nokta derecelendirmesinde en yakın uzaklığı bulurken uzaklık filtresi uygulamak yerine analitik geometrideki doğrudan uzaklık formülleri kullanılabilir. Bu şekilde en yakın uzaklığı bulup direk algoritmadan çıkılabilir. Bu düşüncenin algoritmaya hız kazandıracağını düşünüyoruz fakat hız kıyaslamaları yapılarak daha hızlı olması durumuna göre doğru denklemleri kullanılabilir.

Lokal pencerenin büyüklüklerini, görüntünün biçimine göre veya uygulanmak istenilen alana göre ayarlamak gelecekteki görevimiz olacaktır. Böylelikle daha dinamik bir hesaplama sonucu ortaya çıkacaktır. Bir diğer görevimiz ise yaptığımız uygulamalar ile elde ettiğimiz kritik alanları faz alan denklemlerine uygulamak olacaktır. Kısıtlı faz alan yapısının hem fiziksel hem de grafik üzerinde olmak üzere çeşitli uygulamaları vardır. Bu uygulamalar üzerinde çalışıp geliştirmeler yapmayı hedeflemekteyiz. Ayrıca faz alanı sınırlamalarını yaparken her adımdaki kritik alanları güncellemek işlemlerimizi yavaşlatabilir. Bu sorunu çözmek için de kritik alanların benzerliklerini kullanarak faz-alan uygulaması geliştirmeleri yapmayı hedeflemekteyiz.

VII. KAYNAKLAR

1. Caglar, S. , "*Topoloji Nedir?*", (2017), https://www.matematiksel.org/topoloji-nedir/adresinden_alindi.
2. Acar, R., Sagirli, N., "*Phase Field Topology Constraints*", *Research in Shape Analysis, Wish2 Sirince, Turkey, June 2016, Association for Women in Mathematics Series, Vol:12, Springer, 2018.*
3. Dogan, M. , "*Kompakt İrtibatlı Lie Grupları Üzerinden Demetler ve Homotopi Normalite*", Ankara, 2008, s. 9-10, acikarsiv.ankara.edu.tr/browse/24536/mustafa_dogan_tez.pdf adresinden alındı.
4. Loring, W. , "*Morse Theory*", (January 2001), <http://www.math.harvard.edu/history/bott/bottbio/node9.html> adresinden alındı.
5. "*Morse-Smale Complex*", (t.y.) ,<http://www2.cs.uh.edu/~chengu/Teaching/Spring2013/Lecs/Lec8.pdf> adresinden alındı.
6. Kundu, A. (t.y.) , "*Introduction to Reeb Graphs and Contour Trees*" , <http://web.cse.ohio-state.edu/~wang.1016/courses/788/Lecs/lec15-abhisek.pdf> adersinden alındı.
7. Kong, T., Rosenfeld, A. , "*Digital topology: Introduction and survey*", *Computer Vision, Graphics and Image Provesing, Elsevier, Vol:48(3), 1989, s.357-393.*
8. Pham DL., Bazin, PL. and Prince, JL. , "*Digital Topology in Brain Imaging*", *IEEE Signal Processing, Vol:27, 2010, s.52-55.*
9. Passat, N., Mazo, L., "*An introduction to simple sets*". *Pattern Recognition Letters, Vol:(30)15, 2009, s. 1366-1377.*

10. Ronse, C., "Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images", *Discrete Applied Mathematics*, Vol:(21)1, 1988, s.67-79.
11. Acar, R. , "Simulation of Interface Dynamics: A diffuse Interface Model", *Visual Computer*, Vol:(25)2, 2009, s.101-115.
12. Cayiroglu, İ. (t.y.), "Resmi Alt Bölgelere Ayırma", http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-8.Hafta.pdf adresinden alındı.
13. Shapiro, L. ,Stockman, G. , "Computer Vision", Pearson, 2001, s.69-75.
14. "Connected-component_labeling", (t.y.), https://en.wikipedia.org/wiki/Connected-component_labeling adresinden alındı.

VIII. ÖZGEÇMİŞ

NECATİ SAĞIRLI

e-posta: necati.sagirli@hotmail.com

EĞİTİM

2013 - 2018	OKAN ÜNİVERSİTESİ Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği	İstanbul
2006 - 2011	DUMLUPINAR ÜNİVERSİTESİ Fen Edebiyat Fakültesi Matematik Bölümü	Kütahya

İŞ DENEYİMLERİ

01.10.2012 - Devam	ASSİSTT Operasyonel Planlama Kıdemli Uzaman	İstanbul
01.09.2011 - 01.10.2012	YARGI AKADEMİ DERSHANESİ Matematik Öğretmeni	Erzincan
01.02.2011 - 01.07.2012	T.C. MERCAN LİSESİ Matematik Öğretmeni	Erzincan

YAYINLAR

1. Acar, R., Sagirli, N., "Phase Field Topology Constraints", Research in Shape Analysis, Wish2 Sirince, Turkey, June 2016, Association for Women in Mathematics Series, Vol:12, Springer, 2018.

