T.C.

GEBZE YÜKSEK TEKNOLOJI ENSTİTÜSÜ MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

3-B YAPININ FORM ALAN DIŞ HATLAR KULLANILARAK ELDE EDİLMESİ

Ulaş VURAL

YÜSKEK LİSANS TEZİ BİLGİSAYAR MÜHENDİSLİĞİ

ANABILIM DALI

GEBZE

2006

T.C.

GEBZE YÜKSEK TEKNOLOJI ENSTİTÜSÜ MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

3-B YAPININ FORM ALAN DIŞ HATLAR KULLANILARAK ELDE EDİLMESİ

Ulaş VURAL YÜKSEK LİSANS TEZİ BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TEZ DANIŞMANI Yrd. Doç. Dr. Yusuf Sinan AKGÜL

GEBZE 2006

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 20/07/2006 tarih ve 2006/26 sayılı kararıyla oluşturulan jüri tarafından 01/09/2006 tarihinde tez savunma sınavı yapılan Ulaş VURAL'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Yrd. Doç. Dr. Yusuf Sinan AKGÜL

ÜYE : Yrd. Doç. Dr. Fatih Erdoğan SEVİLGEN

ÜYE : Yrd. Doç. Dr. İlyas KANDEMİR

ONAY

G.Y.T.E.	Mühendislik v	ve Fen	Bilimleri	Enstitüsü	Yönetim	Kurulu'nun
	tarih ve		/	sayılı k	ararı.	

İMZA/MÜHÜR

ÖZET

TEZ BAŞLIĞI: 3 B Yapının Form Alan Dış Hatlar Kullanılarak Elde Edilmesi

YAZAR ADI: Ulaş Vural

Çizge kesme algoritmaları son dönemlerde bilgisayarla görme alanında oldukça popülerdir. Bu sezgisel yöntemlerin kullanılmasıyla birçok problemin çözümünde başarılı sonuçlar elde edilmiştir. Çizge kesme algoritmalarının etiket sırasına ve başlangıç durumuna bağlı oluşu temel sıkıntılarıdır. Bu noktaların aşılması için çizge kesme sezgisel yaklaşımı çift ağ optimizasyon sürecinin içine entegre edilmiştir. Yeni sistem içinde iki çizge kesme optimizasyonu eş zamanlı ve etkileşimli olarak çalışır. Optimizasyon süreçlerinin birlikte çalışması sistemi yerel en iyi noktalara karşı daha kararlı hale getirmektedir. Sistem kararlığın yanı sıra yeni bilgiler üretmektedir. Elde edilen bilgiler ışığında stereo algoritmalarının en çok zorlandıkları kapanmış, dokusuz ve süreksiz bölgeler bulunabilmektedir. Bu problemli resim bölgelerinin çözümü için iki farklı yaklaşım geliştirilmiştir. Yaklaşımlardan ilki basit etkileşim enerjisine dayanır. İkinci yaklaşım ise güvenilirlik analizi yaparak problemli bölgeleri güvenilir bölgeler yardımıyla tahmin eder. Her iki yöntem de arka alanları ile birlikte açıklanmış ve genel kabul görmüş test resimleriyle denenmiştir. Geliştirilen sistemlerin elde ettikleri sonuçlar diğer algoritmaların sonuçlarıyla karşılaştırılmış ve sistemlerin verimli çalıştıkları gösterilmiştir.

SUMMARY

THESIS TITLE: Recovery Of The 3D Structure Using Deformable Surfaces

AUTHOR: Ulaş Vural

Graph cut methods have been becoming very popular among the Computer Vision community after a number of successful applications for some classic problems. However, some fundamental problems, such as dependency of label order and initial label configuration, still remain to be solved. This thesis proposes a new technique to address these shortcomings by integrating the graph cut methods into the dual mesh framework. The dual mesh move synchronously and they always communicate to overcome the local minima problem. As a result, system produces more stable results and there are some very useful side products. Using these side products, some basic problems of stereo analysis are easily addressed. Such problems include occluded, discontinuous, and textureless areas. Two separate dual mesh deformation techniques are proposed. The first uses a simple communication mechanism. The second utilizes a reliability metric to determine the problematic areas of the stereo images. Both techniques are tested against popular ground truth stereo databases and it is observed that the proposed system works efficiently and accurately.

TEŞEKKÜR

Tezime başlamamdan bitirmeme kadar bana devamlı yol gösteren ve destek veren danışmanım Sayın Yrd. Doç. Dr. Yusuf Sinan AKGÜL'e, beni bu alana yönlendiren ve yüksek lisans eğitimime büyük katkısı olduğunu düşündüğüm Sayın Yrd. Doç. Dr. Fatih Erdoğan SEVİLGEN'e ve yardımlarından dolayı arkadaşım Araş. Gör. Türker AKYÜZ'e sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	İV
SUMMARY	V
TEŞEKKÜR	Vİ
İÇİNDEKİLER DİZİNİ	Vİİ
SİMGELER VE KISALTMALAR DİZİNİ	İX
ŞEKİLLER DİZİNİ	X
ÇİZELGELER DİZİNİ	Xİİ
1 GİRİŞ	1
2 STEREO	4
2.1 Stereo Yönteminin Temelleri	4
2.2 Genel Kabuller ve Kısıtlar	5
2.3 Algoritmaların Sınıflandırılması	7
2.3.1 İlk İşleme	9
2.3.2 Eşleme	10
3 AZAMİ AKIŞ / ASGARİ KESME TEORİSİ	12
3.1 Akış Ağları ve Akışlar	12
3.2 Azami Akış Algoritmaları	18
3.2.1 Ford-Fulkerson Yöntemi	18
3.2.2 Edmonds-Karp Algoritması	21
3.2.3 Preflow-Push Algoritması	21
3.2.4 Lift-To-Front Algoritması	21
3.2.5 Boykov-Kolmogorov Algoritması	22
4 ÇİZGE KESME ALGORİTMALARI	27
4.1 Çizge-Kesme Algoritmalarının Tarihçesi	27
4.2 Çizge Kesme Algoritmalarının Kesin Sonuç	Verdiği Durumlar 30
4.3 Çizge Kesme Algoritmalarında Yaklaşımsal	Yöntemler 32

		viii
4.3.1	αβ-Değiştirme Algoritması ve En İyi αβ-Değişimini	
Bulma		33
4.3.2	α -Genişleme Algoritması ve En İyi α -Genişlemeyi Bulma	37
4.3.3	α-Genişleme Algoritmasının Optimallık Analizi	41
4.4 K	apanma Durumlarını Çizge Kesme Yöntemiyle Çözmek	43
4.4.1	Kapanmaları Dikkate Alan Çizge Kesme Algoritmaları	44
4.5 D	ilimleme Tabanlı Çizge Kesme Algoritmaları	47
4.5.1	Hong ve Chen'in Algoritması	48
4.5.2	Bleyer ve Gelautz'un Algoritması	56
4.6 Ç	izge Kesme ve Diğer Algoritmaların Karşılaştırmaları	62
5 ÇİFT A	AĞ YAPISI	65
5.1 A	ktif Dış Hatlar	65
5.2 Ç	ift Ağ Kullanan Stereo Algoritması	68
6 ÇİZGE	KESME TABANLI ÇİFT AĞ YAKLAŞIMI	72
6.1 Ç	ift Ağ ve Çizge Kesmenin Birlikte Kullanılmasının Nedenleri	72
6.2 St	ereo İçin Çizge Kesme Tabanlı Çift Ağ Algoritması	74
6.2.1	Algoritmanın Açıklanması	74
6.2.2	Deney Sonuçları	80
6.3 St	ereo İçin Güvenilirlik Tabanlı Çizge Kesme Yöntemi	83
6.3.1	Algoritmanın Açıklanması	83
6.3.2	Deney Sonuçları	88
7 SONU	ÇLAR VE ÖNERİLER	91
KAYNAK	LAR	92
ÖZGEÇMÎ	Ş	97

SİMGELER VE KISALTMALAR DİZİNİ

BK Boykov ve Kolmogorov Azami Akış Algoritması

BVZ Boykov, Veksler ve Zabih Stereo Algoritması

KZ1 Kolmogorov ve Zabih Stereo Algoritması

KZ2 Kolmogorov ve Zabih Stereo Algoritması

DINIC Dinic Azami Akış Algoritması

Q_PRF Kuyruk Tabanlı Azami Akış Algoritması

H_PRF En Yüksek Seviye Seçen Azami Akış Algoritması

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. İki Kameralı Stereo Yapısı.	4
3.1. Ford-Fulkerson Azami Akış Algoritmasının Çalışma Şeması.	20
3.2. Boykov ve Kolmogorov Algoritmasında Ağaçların Oluşumu.	22
3.3. BK Algoritması ile Resim Yenileme Sonuçları.	23
3.4. KZ2 ve BK Kullanılarak Elde Edilen Fark Haritaları.	26
4.1. Akış Ağı ve Çizgede s-t Kesme.	27
4.2. 2B ve 3B Çizgede Kesme.	29
4.3. Çizgede Resimlerin Konumları ve Oluşabilecek Hatalı Kesme.	29
4.4. Dört Etiket İçin Enerji Formülünün Çizgesi.	31
4.5. αβ-Değiştirme Algoritması İçin Oluşturulmuş Çizge.	34
4.6. αβ-Değiştirme Algoritmasında Oluşabilecek s-t Kesmeleri.	36
4.7. α-Genişleme Algoritması Tek Boyutlu Çizge Örneği.	38
4.8. α-Genişleme Algoritmasında Oluşabilecek s-t Kesmeleri.	40
4.9. (a) Orijinal Sol Resim, (b) KZ2, (c) Gerçek Fark Haritası, (d) Yerel Eşlen	ne, (e)
BVZ (f) Zitnick & Kanade Algoritması.	46
4.10. Hong ve Chen Algoritmasının Karşılaştırılması.	55
4.11. Bleyer ve Gelautz Algoritmasının Genel İşleyişi.	57
4.12. Piksellerin Fark Katmanlarına Atanması.	59
4.13. Teddy İçin Orijinal Resimler ve Gerçek Fark Haritası.	61
4.14. Sol ve Sağ Fark Haritaları, Sol Resimdeki Dilimlere Göre Fark Haritası.	61
4.15. Sol ve Sağ Resimler İçin İlk Katman Atamaları, Sol Resimdeki Dilimler	e Göre
Katman Ataması.	61
4.16. Gerçek Fark Haritası ile Sol Resmin Dilimlerine Göre Bulunan Fark	
Haritasının Karşılaştırması, Üç Boyutlu Hale Getirilmiş Görüntü.	61
4.17 Stereo Algoritmalarının Sonuçları.	64
5.1. Klasik Yılan.	66
5.2. Yeni Geliştirilmiş Bir Yılan.	66
5.3. İki Tane Şekil Alabilen Ağ Kullanarak Yüzeyin Elde Edilmesi.	69
5.4 Nobuhara ve Arkadaşlarının Hareket Takip Uygulaması.	71
6.1. Manhattan Tarzı 4-8-128 Komşuluklu Sistemler	73

6.2. Komşu Sayısının Sonuca Etkileri	74
6.3. Kesme ve Çizgedeki Bağlar.	78
6.4. Ağların Hareketi.	79
6.5. Çift Ağ Algoritmasının Karşılaştırılması.	81
6.6 Problemli Pikseller (Tsukuba).	82
6.7 Güvenilirlik Tabanlı Algoritmanın Sonuçları.	89

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	Sayta
3.1. "Diamond" Resminde Azami Akış Algoritmaları(Sn).	24
3.2. "Bell Quad" Resminde Azami Akış Algoritmaları(Sn).	24
3.3., Etiket Sayılarının Zamana Etkisi (Sn).	24
3.4. Azami Akış Algoritmalarının BVZ İle Test Zamanları (Sn).	25
3.5. Azami Akış Algoritmalarının KZ2 İle Test Zamanları (Sn).	25
3.6. Azami Akış Algoritmalarının KZ1 İle Test Zamanları (Sn).	25
4.1. αβ-Değiştirme Algoritmasında Kenar Ağırlıkları.	34
4.2. αβ-Değiştirme Algoritması	35
4.3. α-Genişleme Algoritması.	38
4.4. α-Genişleme İçin Kenar Ağırlıklarının Hesaplanması.	39
4.5. αβ-Değiştirme Algoritmasında Kenarların Ağırlıkları.	45
4.6. α-Genişleme Algoritmasında Kenarların Ağırlıkları.	45
4.7. KZ1 ve Diğer Algoritmaların Karşılaştırılması.	46
4.8. Algoritmanın λ Katsayısına Bağımlılığı.	47
4.9. Hong ve Chen Algoritmasında Hata Oranları.	55
4.10 Stereo Algoritmalarının Karşılaştırmaları	63
6.1. Çift Ağ Algoritmasının Karşılaştırılması.	81
6.2. Problemli ve Kapanmış Piksellerin Benzerliği	82
6.3. Çift Ağın Taban Optimizasyona Etkisi.	83
6.4. Güvenilirlik Tabanlı Algoritmanın Karşılaştırılması.	89
6.5. Güvenilir Alanlardaki Hata Oranları	90

1 GİRİŞ

Nesnelerinin üç boyutlu yapılarının bulunması için yapılan araştırmaların başlangıcı 1970'li yıllar olmasına rağmen, teknolojideki geniş kullanım alanı ve önceki yöntemlerin bu teknolojik ihtiyaca cevap verebilecek ölçüde başarılı ve genel nitelikte olmayışı günümüzde de bu araştırmaların yoğun bir şekilde sürdürülmesini gerektirmiştir. Bu eski probleminin çözümü için birçok farklı yöntem geliştirilmiş ve hemen hemen her bir yöntem de kendi çalışma alanını yaratmıştır. Bu yöntemlerden bazıları aktif olarak enerji gönderilip alınmasına dayanırken, diğer yöntemler nesnelerinin hareketlerinden, gölgelerden, yüzeydeki dokudan ya da kameranın odaklanma yeteneklerinden yararlanarak derinlik bilgisini elde etmeyi amaçlamıştır. [Trucco, and Verri, 1998]

Stereo, nesnelerin üç boyutlu yapı bilgisine ulaşmak için oldukça yoğun şekilde kullanılan ve araştırılan bir yöntemdir. Bu alanda yapılan araştırmaları karşılaştıran ya da sınıflayan inceleme makaleleri bile sadece stereo yönteminin belli bir sınıfını inceleyebilmektedir. Aggarwal ve Dhond tarafından daha 1989 yılında yazılmış bir inceleme makalesi bile sadece stereo metodunun özellik tabanlı eşlemeye dayanan algoritmalarını karşılaştırabilmiştir [Dhond, and Aggarwal, 1989]. Stereo yöntemini sadece bir eşleme problemi olarak gören bu klasik yaklaşımların temel tıkanma noktası eşleme aşamasının üç boyutlu yapının yeniden elde edilmesi sürecinde ortaya çıkan bilgiden yararlanamamasıdır. Bu nedenle yoğun fark haritaları üretebilecek olan pencere eşleme tabanlı algoritmalar klasik anlayışta başarısız sonuçlar vermektedir. Günümüzde ise bu klasik yöntemlerin, yerel eşleme ve daha sonra üç boyutlu yapının oluşturulmasına dayalı iki basamaklı mantığı genel olarak terk edilmiş ve problemin bir optimizasyon problemi olarak ele alınıp tek bir çatı altında çözülmesi kabul görmüştür. Stereo probleminin çözümünde gelinen bu son aşama, problemi yerel düzgünlük, süreklilik ve verinin durumuna göre ifade edecek bir küresel enerji fonksiyonunun asgarileştirilmesidir [Koschan, 1993] [Scharstein, and Szeliski, 2002] [Brown et. al., 2003].

Stereo problemini bir enerji fonksiyonu olarak ifade eden yapıların hoş ve anlaşılır görünen birleştirilmiş yapılarına rağmen içlerinde kullanılan enerji

fonksiyonlarının kesin sonuç verecek şekilde optimize edilmesi kolay bir iş değildir. Bu nedenle bu tür sistemlerin en iyileştirilmesinde tavlama benzetimi, çizge kesme, inanç yayma, genetik algoritmalar ve hırslı eğim tabanlı optimizasyon gibi birçok sezgisel yöntem kullanılmıştır. Çizge kesme algoritmaları sadece iki etiket değeri [Greig et. al., 1989] ya da konveks enerji fonksiyonları [Ishikawa, 2003] kullanıldığında kesin sonuç verse de son dönemlerde sezgisel çizge kesme algoritmaları bilgisayarla görmenin birçok probleminde sıklıkla kullanılmaya başlamıştır. Sadece stereo probleminin çözümü için bile; çoklu kamera kullanımını destekleyen [Kolmogorov, and Zabih, 2002], kapanma durumlarını bulabilen [Kolmogorov, and Zabih, 2001], dilimlemeyi [Hong, and Chen, 2004] ve tabakalamayı [Bleyer, and Gelautz, 2004] kullanabilen başarılı çizge kesme algoritmaları son dönemlerde ardı ardına duyurulmuştur. Çizge ve akış algoritmalarının derin bilgi birikimi ve sezgisel algoritmanın verdiği hata sınırı çizge kesme algoritmalarının bu popülerliğinin altında yatan başlıca etmenlerdir. Çizge kesme algoritmaları istatistiksel mekanik ve istatistiksel fiziğin kuramlarından da yararlanılarak elde edilen konveks olmayan fonksiyon terimlerini başarılı şekilde polinomal zamanda hesaplayabilir [Boykov et. al. 1999]. Bu yeteneği ile çizge kesme algoritmaları nesnelerin kenarlarında oluşabilecek süreksizlikleri ve ani atlamaları, bozulmaktan ve aşırı yumuşak geçişlerden korumuş olur.

Çift ağ yapısı ise stereo görüntü çiftlerinden derinlik bilgisi elde edilmesinde [Akgul, and Kambhamettu, 1999] ve dil hareketlerinin ultrasonla takibinde kullanılmıştır [Akgul et. al. 1998]. Çift ağ yapısı, bilinen bir optimizasyon yönteminin iki kopyasının arama uzayının farklı yerlerinden aynı anda başlayarak ve birbirleriyle etkileşerek çalışması üzerine kuruludur. Çift ağ yapısını kullanarak yapılan bir optimizasyonun tamamlanması için her iki ağın da aynı pozisyonlara sahip olmaları beklenir. Eğer ağların hareketi birbirlerine kavuşmadan durmuş ise bu durumda enerjisi daha büyük olan ağın yerel iyi noktaya takıldığı düşünülüp, onun enerjisi küçük olan ağa doğru hareket etmesi sağlanabilir. Bu yöntem ile yerel en iyi noktalara karşı, çalışan tek bir optimizasyondan daha fazla bağışıklık kazanılmış olur. Optimizasyon çiftlerinden ilki arama uzayının başından diğeri ise sonunda başlayıp çalıştırıldığında tüm arama uzayı aranmış olur. Bu özelliği ile çift ağ yapısı başlangıç durumuna bağımlılık sorunundan da kurtulur. Çift ağ yapısının bu

avantajlarının yanında çalıştırdığı optimizasyon algoritmasına bağımlı olarak yerel en iyi noktalara takılma ve sadece sürekli yüzeylerde çalışma gibi engelleri mevcuttur.

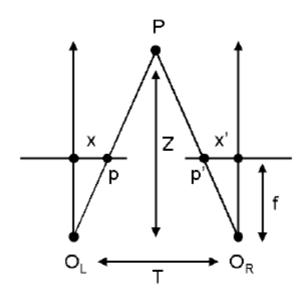
Bu tez çalışmasında çift ağ yapısının içinde çizge kesme ile enerji asgarileştirmesi yapan bir sistemi tanımlanmıştır. Sezgisel çizge kesme algoritmasının çift ağ yapısı ile birlikte tek bir sistemde toplanması hem çizge kesme algoritmasının hem de çift ağ yapısının tek çalıştıkları durumlarda ortaya çıkan eksikliklerini örtmüştür [Vural, and Akgul, "A Graph Cut Based...", 2006] [Vural, and Akgul, "A Reliability Based...", 2006]. Cizge kesme algoritmasının başlangıç konumuna bağlılığı çift ağ yapısı ile aşılmıştır. Çizge kesme algoritmasının içinde kullanılabilen konkav düzgünlük terimleri ile çift ağ yapısının sürekli yüzey isteğine çare bulunmuştur. Sistemlerin tek tek düşünüldüğü bu iki durum haricinde genel olarak sistemin elde ettiği önemli kazanımlar da mevcuttur. Bunlardan ilki çift ağ yapısı ile birlikte kullanılan çizge kesme algoritmasının daha iyi bir enerji seviyesine inebilmesi ve daha doğru fark değerleri elde etmesidir. Elde edilen ikinci başarı ise ağların etkileşimi ile bulunan sonucun güvenilirlik analizinin yapılabilmesidir. Bu analizin sonucunda resimdeki dokusuz, süreksiz ve kapanmaların olduğu bölgeler tespit edilebilmekte ve bu tür problemli yerler için farklı enerji katsayıları ya da terimleri kullanmak mümkün olmaktadır.

Stereo probleminin daha detaylı bir tanımı ile birlikte bu problemin çözümünde genel olarak kullanılan kabuller ve kısıtlar 1. Bölümde anlatılacaktır. 1. Bölümün sonunda stereo algoritmalarının genel bir sınıflandırılması yapıldıktan sonra 2. Bölümde azami akış/asgari kesme teorisi, azami akış/asgari kesme algoritmalarının açıklaması yapılacaktır. 3. Bölümde bu algoritmalara dayanarak geliştirilmiş olan çizge kesme algoritmalarının bir tarihçesi, popüler çizge kesme algoritmalarının ayrıntılı anlatımlarıyla birlikte verilecektir. 4. Bölümde çift ağ yapısı ve bu yöntemi kullanarak derinlik bilgisinin bulunması anlatılacaktır. Yeni sistemin açıklaması 5. Bölüm içinde yapıldıktan sonra 6. Bölümde algoritmalarımızın sonuçları, diğer algoritmalarla karşılaştırmalı olarak yorumlanacaktır.

2 STEREO

2.1 Stereo Yönteminin Temelleri

Stereo, bilgisayarla görme alanında üç boyutlu yapı bilgisini elde etmek için en sık kullanılan ve araştırılan yöntemlerden biridir. Stereo yöntemi temel olarak bir sahnenin aynı anda iki ya da daha fazla görüntüsünün alınmasına ve bu alınan görüntüler üzerindeki kaymaların hesaplanmasına dayanır. Sahne üzerindeki bir nokta iki farklı açıdan çekim yapan kameraların resim düzlemlerinde farklı konumlara düşecektir. Bu yöntem insanların görüşüne de benzemektedir. Her seferinde biri açık olmak kaydıyla gözlerinizi sırasıyla açıp kapatarak bakarsanız parmağımızın arkasındaki basparmağımıza alanın değiştiğini görebilirsiniz. Parmağınızdan gelen ışıklar retina üzerinde farklı konumlara düşmektedir. Bu konum bilgilerini kullanan beyin sahnenin derinlik bilgisini üretmektedir.



Şekil 2.1. İki Kameralı Stereo Yapısı. [Trucco and Veri, 1998.]

Şekil 2.1. üzerinde T kameraların arasındaki uzaklığı, Z nesnenin derinliğini, f odak uzaklığını verdiğinde ve d her iki resimde aynı noktanın arasındaki fark olduğunda derinliği aşağıdaki formül ile elde edebiliriz.

$$Z = f \frac{T}{d} \tag{2.1.}$$

Bu tanımlardan sonra stereo yöntemi içindeki iki temel problemi görebiliriz. Bunlardan ilki noktaların farklı resimler üzerinde nerelere düştüğünün bulunması yani eşleme problemidir. İkinci problem ise kamera parametrelerinin kullanarak gerçek derinlik değerlerinin hesaplanmasıdır. Eşleme problemi oldukça sıkıntılı bir problemdir. Bu yöntemin daha kolay ve hızlı bir şekilde çözülebilmesi için bazı kabul ve kısıtlar geliştirilmiştir. Bu kabul ve kısıtlar algoritmaların içinde sıklıkla kullanıldığından algoritmaların sınıflandırılmasına geçmeden önce bunlardan bazıları anlatılmıştır.

2.2 Genel Kabuller ve Kısıtlar

Eş Kutup Kısıdı (Epipolar Constraint): En güçlü kısıtlardan biridir. Bu kısıt ile iki boyutlu arama uzayını tek boyuta indirgenir ve sadece bir doğru üzerinde arama yapılabilir. Bunu yapabilmek için kameraların ölçümlemesinin yapılmış olması gerekir. Eğer iki kamera doğrultulmuşsa (rectified) bu durumda sadece aynı satır üzerinde arama yapmak yeterli olacaktır.

Kameraların nasıl kalibre edileceği konusunda Boyer ve arkadaşlarının 1991 yılında yazdıkları bir makale mevcuttur. Bununla birlikte ölçümleme yapılmamış kameralar ile de çalışılabilir. Bu konuda Faugeras'ın ve Hartley'in ayrı ayrı yazdıkları makalelerde kalibre edilmemiş kameralardan üç boyutlu yapının elde edilmesi anlatılmaktadır [Koschan, 1993].

Geometrik Benzerlik kısıdı (Geometric Similarity Constraint): Eş doğru parçalarının yönleri ve uzunlukları aynıdır. Bu kısıtta uzunluk kısmı genelde sorun yaratmakla birlikte, açı kısıdı genelde kullanılan bir kısıttır.

Teklik kısıdı (Uniqueness Constraint): Arama alanında benzer aranan alanlardan sadece bir tane mevcuttur.

Fotometrik Kısıt (Photometric Constraint): Eş piksellerin yoğunları benzerdir. Bu kısıt kullanıldığında yüzeyin Lambertian olduğu düşünülebilir. Bu sayede aynı nesnenin farklı açılardaki kameralara yansıma yapıp benzerliği bozmadığı varsayılır.

Renksel Fotometrik Kısıt (Chromatic Photometric Constraint): Jordan ve Bovik 1992 yılında yayınladıkları bir makalede renkli resimler için fotometrik kısıdı değiştirerek kullanmışlardır. Buna göre eş noktaların renk değerleri benzer olmalıdır [Koschan, 1993].

Özellik Uyuşması kısıdı (Feature Combatibility Constraint): İki tanımlanmış elemanın eşleşebilmesi için aynı fiziksel işaretlere sahip olmaları gerekir. Eğer aynı işarete sahip değillerse eşleme olmaz. Bunun için örneğin kenarlar kullanılıyorsa kenarların iyi tanımlanmış olması ve ışık oyunlarının yarattığı gölgelemeleri kenar olarak almaması gerekir. Bu konuda kenardaki renkleri kullanmayı temel alan bir kaç çalışmayı Gershon ve arkadaşları 1987 yılında, Klinker ve arkadaşları 1988 yılında yapmışlardır [Koschan, 1993].

Devamlılık kısıdı (Continuity Constraint): Farklar resimdeki hemen hemen her yerde ufak değişikliklerle dağıtılmıştır. Bu kural özellikle nesne kenarlarında bozulur ama bu tip yerler de resimde az yer tutar. Bu kısıdı D. Marr 1982 yılında insan görüşünü inceleyen bir makalesinde belirtmiştir [Koschan, 1993].

Fark Sınırı (Disparity Limit): Marr ve Poggio 1976 yılında devamlılık kısıdını da kullanarak noktaların resimlerde sınırlı bir farkta olabileceklerini belirtmişlerdir. Bu kısıt arama alanını daraltarak performansın artmasını sağlar.

Sıralılık kısıdı (Ordering Constraint): Eğer sahnedeki nesneler yaklaşık olarak kameraya aynı uzaklıktaysa aranan noktaların ya da özelliklerin resimde belli bir sıra içinde olduğunu söyleyebiliriz.

Bağlılık kısıdı (Connectivity Constraint): Bu güçlü göre eğer bir resimde başka hiç bir bilgi olmadan eğer kenarlar birbirleriyle bağlı şekilde ise bu yapı diğer resimde de aynı şekilde birbirine bağlı olacaktır.

Kapanma kısıdı (Occlusion Constraint): Bir noktanın sadece bir kamerada görünmesi sorunu ise tam olarak çözülememiştir. Bu sorun hem özellik-tabanlı hem de alan-tabanlı algoritmaları etkilemektedir. Bundan en az etkilenmek ve kararlılığı arttırmak için kamera sayısı artırılabilir ya da bu tip engellenme sorunların dar alanda değil geniş alanda olduğunu kabul etmekle aşılır.

2.3 Algoritmaların Sınıflandırılması

Sınıflandırma alanındaki ilk çalışma 1984 yılında Barnard ve Fischer tarafından yapılmıştır 1981 yılından itibaren ve yapılan çalışmaların değerlendirmesini içermektedir. Bu çalışmadan sekiz yıl sonra 1989 yılında bu kez Dhond ve Aggarwal yeni eşleme yöntemleri, özellik tabanlı algoritmaları ve çoklu kamera sistemleri hakkında yapılan çalışmaların bir gözden geçirmesini yayınlamışlardır [Dhond, and Aggarwal, 1989]. Bundan sonra 1993 yılında Koschan ve daha sonra 2003 yılında Brown, Burschka ve Hager bu alanda yapılan çalışmaların bir derlemesini ve sınırlı karşılaştırmalarını vermişlerdir [Koschan, 1993] [Brown et. al., 2003]. Bu alanda çok fazla bilimsel araştırma yapıldığından karşılaştırma yapan makaleler sadece belli bir yöntemi kullanan algoritmaları - onu da sınırlı sayıda - alarak bu işlemi yapmaktadır. 2002 yılında Szeliski ve Scharstein 2-Görüntüden yoğun fark haritaları yaratan algoritmaları inceleyen bir sınıflandırma ve karşılaştırma makalesi yazmışlardır [Scharstein, and Szeliski, 2002]. Bu tez çalışması içindeki sınıflandırma şimdiye kadar yapılan tüm inceleme makaleleri birlikte kullanılarak yapılmıştır ama performans karşılaştırmalarına çok fazla girilmemiştir.

Algoritmalar yerel ve küresel algoritmalar olarak iki genel sınıf altında incelenebilir. Yerel algoritmalar kendi yerellerinde yanlış eşlemelere takılabilirler ve hatalı sonuçlar üretebilirler bununla birlikte arama alanları daha dar olduğu için küresel algoritmalara kıyasla daha az hesaplama gücü istemektedirler. Küresel

yöntemler ise genellikle problemi bir enerji fonksiyonunun optimizasyonu olarak ele alırlar. Bu yöntemler yerel en iyi noktalara karşı daha dirençli olmakla birlikte hesaplama karmaşıklıkları yerel algoritmalara nazaran daha fazla olacaktır. Çizelge 2.1'de stereo algoritmalarının yerel ve küresel olarak sınıflandırılması ve algoritmalarının kısa açıklamaları görülebilir.

Çizelge 2.1. Stereo Algoritmalarının Tasnifi.

Yaklaşım	Kısa Açıklama					
YEREL YÖNTEMLER						
Blok Eşleme (Block Matching)	Ufak bir arama alanında en fazla eşleme skorunu ya da en az hata oranını arar. Genelde çapraz-ilişki veya sağlam rütbe ölçütlerini kullanır.					
Eğim-Tabanlı En Uygun Şekle Sokma (Gradient Based Optimization)	Küçük bir alanda örneğin SSD gibi bir fonksiyonu asgari yapmaya çalışır.					
Özellik Eşleme (Feature Matching)	Işık yoğunlukları yerine belli nesne özelliklerini (örneğin: kenarlar) kullanarak eşleme yapar.					
KÜRESEL YÖNTEMLER						
Dinamik Programlama (Dynamic Programming)	Tablo içinden en iyi yolu bulmaya çalışır. Buradaki sıralılık genelde eş kutup kısıdı sayesinde tanımlanır.					
Çizge Kesme (Graph Cuts)	Fark yüzeyini akış ağındaki asgari kesmeyi kullanarak elde eder.					
Lineer Olmayan Yayınım (Nonlineer Diffusion)	Yerel yayınım süreçlerini kullanarak toplama yapar.					
Güven Yayma (Belief Propagation)	Bir güven ağında mesajlaşarak farkları çözer. Bu güven ilişkisi eşleme skorlarındaki marjinalliklere göre belirlenir.					
Eşlemesiz Yöntemler (Correspondenceless Methods)	Sahnenin modelini bir amaç fonksiyonu kullanarak yaratır.					

Bu genel tasnif enerji fonksiyonunun hesaplanmasında ve en iyileştirilmesinde kullanılan yöntemleri göstermektedir. Bu iki işlev stereo probleminin çözümünde oldukça önemli olmakla birlikte stereo probleminin çözümü sadece bu basamaklardan ibaret değildir. Bir stereo problemi genel olarak üç adımdan oluşur.

Adım–1. İlk işleme (Preprocessing)

Adım–2. Eşleme (Matching)

Adım–2.1. Eşleme Maliyetinin Hesaplanması

Adım-2.2. Maliyetin Toplanması

Adım–2.3. Farkın Hesaplanması / En İyileştirilmesi

Adım–2.4. Farkın Arıtılması

Adım–3 3-Boyutlu Yapının Belirlenmesi (3-D Structure Determination)

2.3.1 İlk İşleme

Bu aşama daha çok özellik temelli algoritmalarda kullanılır ve özelliklerin daha belirgin hale getirilmesini sağlar. İlk işleme basamağı sinyal işleme alanına daha yakındır. Bu alanın oldukça eski ve üzerinde çalışılmış olması hızlı çalışabilen algoritma ve maskelerin kullanılabilmesine olanak sağlamıştır.

Marr- Hildreth yöntemi Laplacian of Gaussian'a yakalaşan bir yöntem sunar. Grimson, Mayhew ve Frisby, Kim ve Aggarval, Ayache ve Faverjon ile diğer birçok kişi algoritmalarında Marr-Hildreth yöntemi ile özellikleri belirginleştirmeyi tercih etmiştir. Bu yöntemin dışında Canny, Deriche, Sobel, Prewitt ve Haralick matematiksel türev operatörlerine yaklaşan maskeler kullanmışlardır. Medioni ve Nevatia basamak-kenar maskeleri kullanarak kenarların yönlerini esas olarak almışlardır. Bu tür maskeler kenarların doğrultusunu ve büyüklüğünü verirler. Üçüncü tür kenar algılama yöntemleri ise yerel gri seviye yoğunluklarını kullanarak kenar elemanlarının (edgel) bilgilerini elde etmeyi amaçlar. Bu operatörler sonuç olarak kenar elemanının yönünü ve büyüklüğünü verirler [Dhond, and Aggarwal, 1989] [Scharstein, and Szeliski, 2002].

2.3.2 Eşleme

Eşleme maliyetlerinin hesaplanmasında kullanılan belli başlı yöntemleri şu şekilde listeleyebiliriz.

a) Yoğunluk Farklarının Kareleri: Pikseller arasındaki ışık yoğunluklarının farklarının karelerine göre maliyeti belirler. Bu yöntem video işleme alanında "ortalama-kare hatası" olarak da bilinir.

$$|C| = \frac{1}{n^2} \left(\sum_{i=1}^n \sum_{j=1}^n (A_{ij} - B_{ij})^2 \right)$$
 (2.2.)

b) Yoğunluk Farklarının Mutlak Değeri: Pikseller arasındaki ışık yoğunluklarının farklarının mutlak değerlerinin toplamına göre eşleme maliyeti belirlenir. Bu yöntem video işleme alanında "ortalama-mutlak değer farkı" olarak da bilinir.

$$|C| = \frac{1}{n^2} \left(\sum_{i=1}^n \sum_{j=1}^n |A_{ij} - B_{ij}| \right)$$
 (2.3.)

- c) İkili Eşleme Maliyeti: Bu yöntemde pikseller eşlendi/eşlenmedi olarak iki tür değer alırlar. Bu yöntem daha çok özellik tabanlı stereo algoritmalarında kullanılmaktadır.
- d) Normalleştirilmiş Çapraz İlişki Maliyeti: Bu yöntem ise yoğunluk farklarının toplamına benzer özellikte bir maliyet çıkartacaktır.

$$|C| = \frac{\sum_{i,j} (A_{ij} - \bar{A}) (B_{ij} - \bar{B})}{\left[\sum_{i,j} (A_{ij} - \bar{A})^{2}\right]^{\frac{1}{2}} \left[\sum_{i,j} (B_{ij} - \bar{B})^{2}\right]^{\frac{1}{2}}}$$
(2.4.)

e) Yeni Ölçütler: Bu yöntemlerin bir kısmı, pikseller için tek tek çıkartılmış maliyetlerin toplanması sırasında oluşacak eşlenmemiş piksellerin hesaba olan olumsuz etkisini sınırlandırmaya yararlar. Truncated quadratics ve contaminated Gaussians gibi yöntemleri kullanırlar.

Yeni geliştirilmiş diğer yöntemlerin bir kısmı kamera kazancı ya da sapmalarından etkilenmeyecek şekilde geliştirilmişken (eğim-tabanlı ölçütler) diğer bir bölümü (Birchfield ve Tomasi tarafından geliştirilmiştir.) ise pikselleri birbirleriyle karşılaştırmak yerine piksellere göre oluşturulmuş fonksiyonları karşılaştırarak örnekleme hatalarından maliyet ölçütünü yalıtmıştır.

Bu şekilde oluşturulan maliyetler daha sonra toplanarak ve/veya ortalaması alınarak kullanılır. Maliyetler çıkartıldıktan sonra eşleme bölümünün en önemli kısımlarından olan en iyileştirme süreci gelir. Bunun için kullanılan yöntemleri ise şu şekilde sınıflandırabiliriz:

- Yerel Yöntemler: Hızlı hesaplama sağlarlar. Yerel en iyi noktalarda kalırlar.
- 2. **Küresel Yöntemler:** Görece daha yavaştırlar. Düzgünlük kısıdı ile daha iyi sonuçlar üretirler. Çizge kesme, dinamik programlama, tavlama benzetimi, rastlantısal Markov alanları, ortalama alan tavlaması gibi yöntemleri küresel yöntemler için de sayabiliriz.
- 3. **Birlikte Çalışan Algoritmalar:** Önce yerel yöntemleri daha sonra küresel yöntemleri kullanarak yerel yöntemlerden iyi, küresel yöntemlerden hızlı çözümler üretirler.

Elde edilen çözümün daha da iyi olması için son adımda tabakalama, kama (spline) ve düzlem uydurma gibi adımlar kullanılabilir. Bu yöntemler alt piksel konumlarının bulunmasına yardımcı olurlar.

Bundan sonraki bölümde farkların ya da bir başka değişle eşleme maliyetlerinin optimizasyonu için kullanılan çizge kesme algoritmalarına değinilecektir. Bunun için öncelikle çizge kesme algoritmalarının altındaki çizge teorisi anlatılacaktır.

3 AZAMİ AKIŞ / ASGARİ KESME TEORİSİ

Çizge kesme algoritmaları ile problemler çözülürken problem önce bir çizge olarak ifade edilir. Bu çizgenin düğümleri arasındaki benzerlik ilişkisi bir maliyet fonksiyonun hesaplanmış değerleri olarak bu düğümler arasındaki kenarlara yerleştirilir. Oluşturulan çizgeler algoritmadan algoritmaya farklılıklar göstermekle birlikte çizge kesme algoritmaları oluşturdukları çizgenin maliyetini optimize etmek için bu maliyetleri akış kapasitesi olarak düşünüp azami akış/asgari kesme algoritmalarını kullanırlar. Çizge kesme algoritmalarını incelerken öncelikle azami akış/asgari kesme algoritmalarını ve bunların altındaki çizge teorisini incelenecektir. Bir sonraki kısımda ise çizge kesme algoritmalarının tarihçesi içinde çizge kesme algoritmalarının çizgeleri nasıl oluşturdukları anlatılacaktır.

3.1 Akış Ağları ve Akışlar

Akış ağı G=(V,E), hiç bir $f:VxV\rightarrow R$ kenarı negatif olmayan, bağlı ve yönlü bir çizgedir. Her bir kenar, negatif olmayan ağırlığa sahiptir. **Kapasite** olarak adlandırılan bu ağırlığı $c(u,v) \ge 0$ olarak gösterilebilir. Akış ağı iki tane özel nokta içerir. Bu noktalardan birincisi **kaynak** (s), ikincisi ise **hedef** (t) noktasıdır.

Bu ağdaki **akış** ise şu şekilde tanımlanabilir: G=(V,E) akış ağı, s kaynak noktası ve t de hedef noktası olsun. G çizgesi içindeki bir akış, aşağıdaki şartları sağlayan gerçel değerli $f: VxV \to R$ fonksiyonudur:

1. **Kapasite Sınırı:** Her $u,v \in V$ için $f(u,v) \le c(u,v)$ olmalıdır. Bu sınıra göre bir kenardan geçirebileceğimiz azami akım o kenarın kapasitesi ile sınırlıdır. Bu kapasite hiç bir zaman negatif olamaz. Çizgede bulunmayan kenarlar için kapasitenin sıfır olduğu düşünülür.

- 2. Çarpık Simetriklik: Her $u, v \in V$ için f(u, v) = -f(v, u) olmalıdır. Bu durumda bir noktadan kendine olan akış sıfır olur.
- 3. **Akışların Korunumu:** Her $u \in V \{s, t\}$ için $\sum_{v \in V} f(u, v) = 0$ olmalıdır. Bu Kirchhoff akım kanunu gibi düşünülebilir. Bir noktaya gelen akımlar ile çıkan akımların toplamı sıfırdır.

Net Akış: Pozitif ya da negatif olabilecek olan f(u,v) miktarı, u noktasından v noktasına olan net akışı gösterir.

Akış Değeri: Bir f akışının değeri kaynak (s) noktasından o noktaya olan net akışı gösterir.

$$|f| = \sum_{v \in V} f(s, v) \tag{3.1.}$$

Azami Akış Problemi: G akış ağında s kaynak noktasından t hedef noktasına olabilecek azami akışı bulmayı amaçlayan problemdir.

Süper Kaynak ve Süper Hedef: Azami akış problemi tek bir hedef ve kaynak için tanımlanmıştır ama birden fazla kaynak ve hedef için de çözülebilir. Bunun çözümü için çizgeye bir tane süper kaynak ve bir tane süper hedef eklenir. Süper kaynak ile diğer kaynaklar arasında sonsuz akışa izin verecek şekilde kenarlar eklenir. Aynı şekilde hedef noktalar ile süper hedef arasında da sonsuz miktarda akışın geçmesine izin verilirse, azami akış problemi tek bir hedef ve tek bir kaynağa indirgenmis olur.

Örtülü Toplama Gösterimi: X ve Y iki farklı nokta kümesi x, X kümesindeki bir nokta ve y, Y kümesindeki bir nokta olsun, bu durumda bu nokta kümeleri arasındaki akışın toplamı (3.2.) ile gösterilir.

$$f(X,Y) = \sum_{x \in X} \sum_{v \in Y} f(x,y)$$
 (3.2.)

Önsav–1: G(V,E) bir akış ağı ve f, G akış ağındaki bir akış olsun.

$$X \subseteq V$$
 için, $f(X,X) = 0$.

$$X, Y \subseteq V$$
 için, $f(X, Y) = -f(Y, X)$.

$$X,Y,Z\subseteq V$$
 ve $X\cap Y=\varnothing$ için, $f(X\cup Y,Z)=f(X,Z)+f(Y,Z)$ ve
$$f(Z,X\cup Y)=f(Z,X)+f(Z,Y).$$

Kaynak ve hedef noktalar haricindeki tüm $v \in V - \{s,t\}$ noktaların net akışları, akışların korunumu kısıdı nedeniyle sıfır olacaktır. Kaynaktan çıkan net akış pozitif, hedefteki akış kaynaktaki akışa eşit büyüklükte ve negatif olacaktır.

İspat:

$$|f| = f(V,t)$$

$$|f| = f(s,V) = f(V,V) - f(V - s,V)$$
 (Önsav–1)

$$|f| = f(V, V - s)$$
 (Önsav–1)

$$|f| = f(V,t) + f(V,V-s-t)$$
 (Önsav–1)

$$|f| = f(V, t)$$
 (Akışlarının Korunumu)

Artan Kapasite: u noktasından v noktasına c(u,v) kapasitesini aşmayacak şekilde gönderilebilecek ek net akım miktarına denir.

$$c_f(u,v) = c(u,v) - f(u,v)$$

Artan Ağ: G=(V,E) akış ağı ve f akışı verilmiş olsun, f tarafından indirgenen G ağından oluşturulan **artan ağ** $G_f = (V, E_f)$, $E_f = \{(u,v) \in VxV : c_f(u,v) > 0\}$ şeklinde tanımlanır.

Önsav-2: G=(V,E) bir akış ağı, s ve t bu ağdaki kaynak ve hedef noktalar ve f bu ağ içinde bir akış olsun. G_f , G ağının f akışı tarafından indirgenmiş artan ağı ve f' artan ağ G_f içinde bir akış olsun. Bu durumda |f+f'|=|f|+|f'| olur.

İspat:
$$(f+f')(u,v) = f(u,v) + f'(u,v)$$

 $(f+f')(u,v) = -f(v,u) - f'(v,u)$ (Çarpık Simetriklik)
 $(f+f')(u,v) = -(f(v,u) + f'(v,u)) = -(f+f')(v,u)$

Kapasite kısıdına göre her $u,v \in V$ için $f'(u,v) \le c_f(u,v)$ olmalıdır.

$$(f + f')(u,v) = f(u,v) + f'(u,v)$$
$$(f + f')(u,v) \le f(u,v) + (c(u,v) - f(u,v)) = c(u,v)$$

Akışların korunumu için her $u \in V - \{s, t\}$ için

$$\sum_{v \in V} (f + f')(u, v) = \sum_{v \in V} f(u, v) + f'(u, v)$$

$$\sum_{v \in V} (f + f')(u, v) = \sum_{v \in V} f(u, v) + \sum_{v \in V} f'(u, v) = 0 + 0 = 0$$

Sonuç olarak;

$$|f+f'| = \sum_{v \in V} (f+f')(s,v) = \sum_{v \in V} f(s,v) + \sum_{v \in V} f'(s,v) = |f| + |f'|$$
 (3.3.)

Kesme: G=(V,E) akış ağındaki bir (S,T) kesmesi çizgedeki V noktalarını S ve T=V-S ve f(S,T)=|f| olacak şekilde iki kısma ayırır. Bu kesmenin kapasitesi c(S,T) ile gösterilir.

Önsav–3: f, G akış ağında s kaynaklı ve t hedefli bir akış ve (S,T) G akış ağında bir kesme olsun. Bu durumda, (S,T) boyunca net akış:

$$f(S,T) = |f| \tag{3.4.}$$

İspat: Önsav–1 kullanılarak,

$$f(S,T) = f(S,V) - f(S,S) = f(S,V)$$

$$f(S,T) = f(s,V) - f(S-s,V) = f(s,V)$$

$$f(S,T)=|f|$$

Önermenin Doğal Sonucu: G akış ağındaki herhangi bir f akışı G ağındaki herhangi bir kesmenin kapasitesi ile üstten sınırlıdır.

İspat: (S,T), G ağındaki herhangi bir kesme ve f, G ağındaki herhangi bir akış olsun. Önsav–3 ve kapasite kısıdını kullanarak:

$$|f| = f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) \le \sum_{u \in S} \sum_{v \in T} c(u,v) = c(S,T)$$
 (3.5.)

Teorem 1 (Azami Akış – En Az Kesme Teoremi)

Eğer f, G=(V,E) akış ağında s kaynaklı ve t hedefli bir akış ise aşağıdaki şartlar eşittir:

- 1. f, G akış ağındaki azami akıştır.
- 2. Artan ağ G_f eklenebilecek yol içermez.
- 3. G ağındaki herhangi (S,T) kesmesi için |f| = c(S,T)

İspat: (1) Bunun ispatı yalanlama yöntemi ile yapılabilir. f, G akış ağındaki azami akım olsun fakat artan ağ G_f eklenebilecek bir p yolu içersin. Bu yolun eklenmesiyle elde edilmiş $f+f_f$ akışı yeni azami akımı oluşturur.

- 2) Eğer G_f eklenebilir yol içerirse hala azami akış elde edilmemiş demektir. Bu nedenle azami akışın bulunabilmesi için artan ağda eklenebilir yol olmamalıdır.
- (3) Önsav-3'ün doğal sonucundan her (S,T) kesmesi için $|f| \le c(S,T)$ olduğunu görürüz, eğer |f| = c(S,T) ise bu akış azami akıştır.

Asgari kesme probleminin azami akış probleminin çözüm yöntemleriyle çözülebileceği gösterilmiş oldu. Şimdi ise azami akış problemi için geliştirilmiş yöntemlerden bir kaç tanesinin genel özelliklerine bakılacaktır.

3.2 Azami Akış Algoritmaları

Azami akış probleminin çözümü için birçok algoritma ve yöntem geliştirilmiştir. Bu algoritmalar aşağıdaki şekilde sınıflandırılabilir:

- 1. Ford Fulkerson Yöntemi
- 2. Edmonds Karp Algoritması
- 3. Preflow-Push Algoritmaları
- 4. İleri Kaldırma Algoritmaları
- 5. Boykov-Kolmogrov Yol Ekleme Algoritması

3.2.1 Ford-Fulkerson Yöntemi

Bu algoritmanın birçok farklı zaman sürelerinde çalışan uyarlamaları olduğundan yöntem olarak da anılmaktadır. Artan ağda bulduğu yolları ekleyerek akışı azami hale getirmeye çalışır. İki temel kavramın açıklanması gerekir:

- 1. Artan Ağ: Kapasitesini dolduran yollar çıkartıldıktan sonra geri kalan ağdır. Akış geçen yollar kapasitesinden arta kalanı tutar. Her seferinde geri kalan ağ üzerinde kaynaktan hedefe gidecek yol bulunmaya çalışılır. Eğer artan ağ üzerinde bu şekilde bir yol bulunamazsa azami akış bulunmuş demektir.
- 2. Yolların Eklenmesi: Her aramadan sonra bulunan yollar azami akışa eklenir. Bu ekleme sırasında yollardan geçebilecek akış en küçük kapasiteli yolun izin verdiği kadardır. Her eklenen akım için çizgeye ters yönde aynı akım miktarında bir yol eklenir. Bu yolun eklenmesiyle ilk başta yapılabilecek yanlış seçimlerin daha sonra düzeltilebilmesi sağlanır.

FORD-FULKERSON-METODU (G, s, t)

- 1. İlklendirme **f**=0
- 2. Eklenebilecek **p** yolu olduğu sürece yap
 - 2.1 **p** yolunu **f** akışına ekle
- 3. f akışını döndür

Temel Ford-Fulkerson Algoritması

FORD-FULKERSON(G, s, t)

1. Her $(u,v) \in E[G]$ için yap

$$1.1 f[u,v] \leftarrow 0$$

1.2
$$f[v,u] \leftarrow 0$$

2. G_f artan ağında kaynaktan hedefe p
 yolu olduğu müddetçe yap

2.1
$$c_f(p) \leftarrow min\{c_f(u, v) : (u, v) \in p\}$$

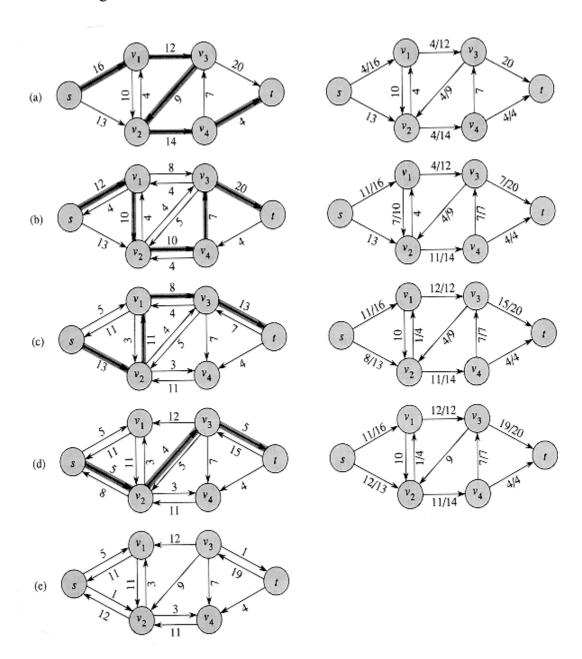
2.2 Her
$$(u,v) \in p$$
 için yap

2.2.1
$$f[u,v] \leftarrow f[u,v] + c_f(p)$$

2.2.2
$$f[v, u] \leftarrow -f[u, v]$$

Bu algoritmanın zaman karmaşıklığı $O(E|f^*|)$ dir. $|f^*|$ azami akış miktarıdır. Bu yöntemde eğer eklenecek yol doğru seçilemezse algoritma sonlanmayabilir ya da

çok uzun zaman sürebilir. Zaman karmaşıklığı akış miktarına bağlı olduğundan azami akış miktarı azsa algoritmanın çalışma zamanı daha iyi olacaktır. Algoritmanın sonlanma sorunu olmasına rağmen, algoritma eğer sonlanırsa kesinlikle doğru sonucu verecektir.



Şekil 3.1. Ford-Fulkerson Azami Akış Algoritmasının Çalışma Şeması. [Cormen et. al., 2001.]

3.2.2 Edmonds-Karp Algoritması

Ford-Fulkerson algoritması gibi yol ekleme tabanlıdır. Temel algoritmanın eksik yönlerini giderir ve daha iyi zaman karmaşıklığına sahiptir. Tamamlanamama sorunu gidermek için rasgele yol bulma yerine genişlik öncelikli arama yapar. Bu yöntemle zaman karmaşıklığı $O(VE^2)$ olur.

3.2.3 Preflow-Push Algoritması

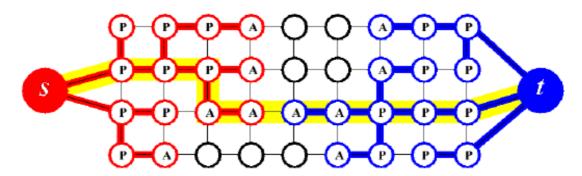
Bu algoritma türleri yol ekleme türü algoritmalardan daha hızlı çalışmakta ve paralelleştirilebilmektedir. Bu sayede GPU (Grafik İşleme Birimi) ile birlikte kullanıldığında çizge kesme algoritmalarının gerçek zamanlı çalışması söz konusu olabilmektedir. Goldberg'in genel Preflow-Push algoritmasının zaman karmaşıklığı $O(V^2E)$ dir. Bu algoritma türünde noktaların rezervuarları vardır ve fazla akışı belli bir süre tutarlar. Akış noktaların yüksekliğine göre olur. Bu nedenle kaynaktan akış çıktıkça aradaki noktaların yükseklikleri değişir. Yüksek olan noktadan alçaktaki noktalara doğru akış devam eder. Algoritma sonlandığında rezervuarlarda hiç akı birikimi kalmamış, gerekirse bu akışların hepsi kaynağa dönmüş olur.

3.2.4 Lift-To-Front Algoritması

Ağ veri yapısını daha verimli kullanarak ve seçim sırasını daha dikkatli yaparak Preflow-Push algoritmasından daha hızlı çalışacak bir algoritma elde edilebilir. Bunun için öncelikle bir akım geçtiğinde rezervuarlarında birikme olan noktaların bu birikimlerini boşaltmaları sağlanır. Bu birikimi komşularına gönderebilmesi için komşuluk listeleri tutulur. Bu yöntem ile maksimum akış bulma algoritmasının zaman karmaşıklığı en az Preflow-Push kadar iyi olan $O(V^3)$ olur.

3.2.5 Boykov-Kolmogorov Algoritması

Algoritmanın temelinde yol ekleme yöntemi vardır. Eklenecek yolları bulmak için arama ağacı oluşturur. Çizge içerisinde s kaynak noktasından ve t hedef noktasından başlayan iki tane arama noktası vardır. Bu arama ağaçlarının ortak elemanları yoktur. Ağaçların elemanları aktif ya da pasif olabilirler. Aktif noktalar ağacın en dışındaki noktalardır. Çizge içindeki her nokta ya T ya S ağacına bağlı ya da bağımsızdır.



Şekil 3.2. Boykov ve Kolmogorov Algoritmasında Ağaçların Oluşumu. [Boykov and Kolmogorov, 2002.]

Algoritmanın temel olarak üç durumu vardır:

- 1. **Büyüme:** Bu aşamada aktif noktalar bağımsız uçlarla doymamış kenarları varsa bağlantı kurarlar. Aktif uç bir yavru edindiğinde pasifleşir ve en dışa eklenen yeni nokta aktif olur. Büyüme işleminde hangi noktanın ekleneceği rasgele belirlenir.
- 2. **Ekleme:** Büyüme aşamasında bulunan yeni yol akışa eklenir. Gönderebileceğimiz azami akışı gönderdiğimizden bazı yollar doyuma ulaşacaktır. Bu durumda önceden ağaca bağlı olan noktaların bazıları ebeveyn noktadan koparlar ve yetim duruma düşerler.
- 3. **Uyum:** Yetim duruma düşmüş noktalara mevcut bulundukları ağaçta yeni bir ebeveyn aranır. Eğer bu ebeveyn nokta ile doymamış bir bağ ile yeniden bağlantı kurulabilirse bu noktalar yaşamaya devam ederler yoksa bağımsız nokta durumuna düşerler.

Algoritmanın, çalışma zamanı teorik olarak preflow-push algoritmalarından daha kötüdür ama bilgisayarla görme alanındaki uygulamalarda diğer algoritmaların tümünden daha iyi durumdadır. Denic'in algoritmasındaki gibi ağaç her seferinde yeni baştan kurulmaz ve eski ağaç kullanılır. Her seferinden yeniden ağacı oluşturmaktan kaçınması uygulama zamanında algoritmaya büyük avantaj sağlar. Teorik olarak algoritmanın çalışma zamanı, |C| asgari kesmenin maliyeti olduğunda $O(VE^2|C|)$ olur.

Algoritmanın diğer algoritmalarla karşılaştırmaları için bir test ortamı yaratılmıştır. Algoritmanın resim yenileme ve stereo gibi bilgisayarla görmenin iki önemli problemindeki çalışma zamanları incelenip, yorumlanmıştır.

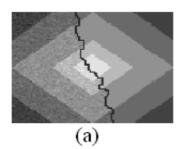
Karşılaştırması yapılan algoritmaların kısa adları ve teorik olarak en kötü çalışma zamanları aşağıda belirtilmiştir.

DINIC: Dinic'in seri, yol ekleme tabanlı algoritması - $O(VE^2)$

H_PRF: En yüksek seviye seçme kurallı Push-Relabel Algoritması - $O(V^3)$

Q_PRF: Kuyruk tabanlı seçme kurallı Push-Relabel Algoritması - $O(V^2 \sqrt{E})$

Resim yenileme problemi için "Diamond" ve "Bell Quad" resimleri kullanılmıştır.







Şekil 3.3. BK Algoritması ile Resim Yenileme Sonuçları. [Boykov and Kolmogorov, 2002.]

Çizelge 3.1. "Diamond" Resminde Azami Akış Algoritmaları(Sn).

Yöntem	Girdi: "Diamond", 210 Çeşit Etiket							
10ntem	35x35	50x50	70x70	100x100	141x141	200x200	282x282	
DINIC	0,39	0,77	3,42	4,19	13,85	43,00	136,76	
H_PRF	0,17	0,34	1,16	1,68	4,69	12,97	32,74	
Q_PRF	0,16	0,35	1,24	1,70	5,14	14,09	40,83	
BK	0,16	0,20	0,71	0,74	2,21	4,49	12,14	

Çizelge 3.2. "Bell Quad" Resminde Azami Akış Algoritmaları(Sn).

Yöntem	Girdi: "Bell Quad", 244 Çeşit Etiket							
Tomem	44x44	62x62	87x87	125x125	176x176	250x250		
DINIC	1,32	4,97	13,49	37,81	101,39	259,19		
H_PRF	0,31	0,72	1,72	3,85	8,24	18,69		
Q_PRF	0,20	1,00	1,70	4,31	10,65	25,04		
BK	0,19	0,48	0,98	2,11	4,84	10,47		

Çizelge 3.3., Etiket Sayılarının Zamana Etkisi (Sn).

Yöntem	Girdi: "Diamond", 100x100			Girdi: "Bell Quad", 125x125				
	L=27	L=54	L=108	L=215	L=32	L=63	L=125	L=250
DINIC	6,89	18,16	50,81	166,27	6,91	17,69	46,64	102,74
H_PRF	3,05	7,38	15,50	47,49	7,47	19,30	58,14	192,39
Q_PRF	2,36	6,41	17,22	43,47	2,39	7,95	15,83	45,46
BK	0,55	1,39	4,34	16,81	1,13	2,95	10,44	41,11

Etiket sayısı arttıkça algoritmanın oransal olarak iyiliği azalmaktadır. Örneğin "Bell Quad" resminde 32 etiket için Q-PRF ile zaman farkı yarı yarıya iken 250 etiket için zamanlar neredeyse aynıdır.

Algoritma, üç farklı stereo algoritmasıyla sınanmıştır. Bunlardan ilki Boykov ve arkadaşlarının önerdikleri α-genişleme yaklaşımıdır [Boykov et. al. 1999]. İkinci algoritma ise Kolmogorov ve Zabih tarafından önerilen kapanma durumlarını

algılayabilen algoritmadır [Kolmogorov, and Zabih, 2001]. Makalede önerilen azami akış algoritması son olarak çoklu kamera sistemleri için geliştirilen bir yöntemle de test edilmiştir [Kolmogorov, and Zabih, 2002].

Çizelge 3.4. Azami Akış Algoritmalarının BVZ İle Test Zamanları (Sn).

Yöntem	"Kafa" (3	384 x 288)	"Ağaç" (2	256 x 233)	"Rasgele" (384 x 288)		
Tonten	N4	N8	N4	N8	N4	N8	
DINIC	104,18	151,32	9,53	19,80	105,93	167,16	
H_PRF	12,00	18,03	1,65	2,86	14,25	18,22	
Q_PRF	10,40	14,69	2,13	3,33	12,05	15,64	
BK	3,41	6,47	0,68	1,42	3,50	6,87	

Çizelge 3.5. Azami Akış Algoritmalarının KZ2 İle Test Zamanları (Sn).

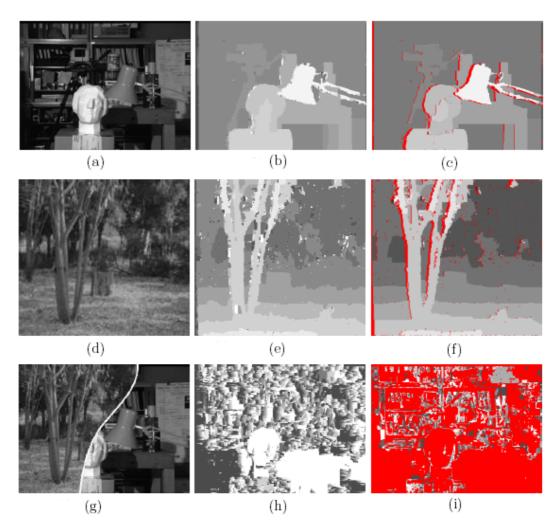
Yöntem	"Kafa" (3	384 x 288)	"Ağaç" (2	256 x 233)	"Rasgele" (384 x 288	
Tomeent	N4	N8	N4	N8	N4	N8
DINIC	376.70	370.94	66.19	102.60	81.70	115.58
H_PRF	35.65	49.81	9.07	15.41	5.48	8.32
Q_PRF	33.12	44.86	8.55	13.64	9.36	14.02
BK	10.64	19.14	2.73	5.51	3.61	6.42

Çizelge 3.6. Azami Akış Algoritmalarının KZ1 İle Test Zamanları (Sn).

	Girdi Dizisi						
Yöntem	"Kafa", 5 Görüntü (384 x 288)	"Bahçe", 8 Görüntü (352 x 240)	"Dayton", 5 Görüntü (384 x 256)				
DINIC	2793.48	2894.85	2680.91				
H_PRF	282.35	308.52	349.60				
Q_PRF	292.93	296.48	266.08				
BK	104.33	81.30	85.56				

Testlerden çıkan sonuçlara göre algoritma kendisinden sonraki en iyi algoritmadan bile 2–5 kez daha hızlıdır fakat algoritmanın diğer algoritmalarla çalışma zamanı oranı çizge karmaşıklaştıkça kötüleşmektedir. Çizgelerinin kesikli

yapıları nedeniyle oluşabilecek geometrik sapmaları gidermenin bir yolu komşuluk sayılarını arttırmaktır fakat daha çok komşu eklendiğinde algoritmanın diğer algoritmalara göre başarı oranı düşmektedir.

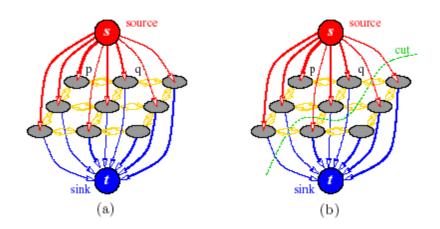


Şekil 3.4. KZ2 ve BK Kullanılarak Elde Edilen Fark Haritaları. [Boykov and Kolmogorov, 2002.]

4 ÇİZGE KESME ALGORİTMALARI

4.1 Çizge-Kesme Algoritmalarının Tarihçesi

D. Greig ve arkadaşlarının 1989 yılında yayınladıkları "Exact maximum a posteriori estimation for binary images" isimli makale çizge-kesme yönteminin bilgisayarla görme alanında kullanımının ilk örneğidir. Bu makalede iki renkli resimlerin yenilenmesinin çizge-kesme yöntemi ile yapılması anlatılmıştır. Makalede verilen örneğin iki renkli resimler için olması onun kısıtlı uygulama alanı olduğunun düşünülmesine neden olmuştur ve makale yayımlandıktan sonra on yıl kadar hiç kimsenin dikkatini çekmemiştir. Buna rağmen Greig ve arkadaşları iki renkli resim yenilemesi gibi basit sayılabilecek bir problemde bile tavlama benzetimi yönteminin en iyiden ne kadar uzak noktalara gidebileceğini göstermiş ve çizge kesme yöntemi ile daha başarılı sonuçlar elde etmiştir. Bu yöntem çok basit olduğu için diğer yöntemlerin anlaşılmasını kolaylaştırabilir. Bu nedenle üzerinde biraz daha fazla durulacaktır.



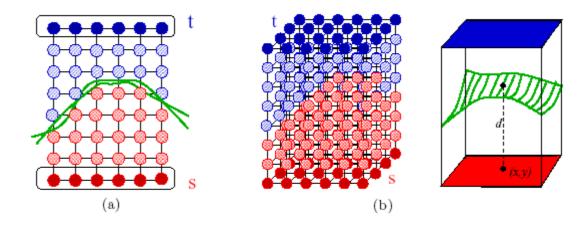
Şekil 4.1. Akış Ağı ve Çizgede s-t Kesme. [Paragios et. al., 2005.]

G=(V,E) çizgesinde V- $\{s,t\}$ noktaları pikselleri gösterirken,l etiketi ya sıfır ya da bir olacaktır $l \in \{0,1\}$. Sadece iki tane etiket olduğundan s kaynak noktası 1, t hedef noktası 0 olarak kabul edilebilir. t-bağlantıları piksellerin eldeki resimdeki yoğunluk değerlerine göre negatif olmayan bir ceza belirtirler. Bu ceza eğer pikselin

görünür değeri 1 ise $D_p(1) < D_p(0)$, görünür değeri 0 ise $D_p(0) < D_p(1)$ olacaktır. Bu bağlantılar enerji fonksiyonunun pikselin mevcut değerine bağlı olmasını sağlarken, n-bağlantıları pikselin değerinin komşularının değerlerine bağlı olmasını sağlar. Bu nedenle n-bağlantıları için farklılıkları koruyucu özelliğe sahip Potts modeli ve bu modelin gücünü belirleyecek bir λ katsayısı kullanılır. Komşuluk için 4 ya da 8 komşu seçilebilir. λ sayısının seçimine göre sonuçta farklılıklar gözükecektir. Çok küçük λ değerleri için pikseller kendi mevcut değerlerini korurken, çok büyük değerleri için de komşularına bağımlılıkları artacaktır. Bu nedenle λ sayısının ayarlanması önemlidir. Sonraları bu katsayının resimdeki piksellerin istatistiksel dağılımına göre adaptasyonunu sağlayan yöntemler geliştirilecektir.

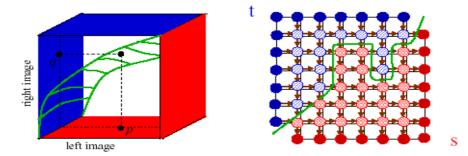
$$E(f) = |C| = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} I(f_p = 0, f_q = 1)$$
(4.1.)

Greig ve arkadaşların yazdıkları bu makaleden sonra çizge kesme algoritmaları iki temel özelliğinden dolayı bilgisayarla görme alanında rağbet görmeye başlamıştır. Bunlardan birincisi hiperyüzey elde etme yeteneği, ikincisi ise enerji asgarileştirmeyi sağlamasıdır. Hiperyüzey elde edilmesi birçok bilgisayarla görme problemi için önemlidir. Çizge kesme yöntemi ile topolojinin kaç boyutlu olduğundan bağımsız olarak yüzey elde edebilir. Bu özelliğinden dolayı resim dilimleme, nesne çıkartma, tıbbi resim analizi, stereo ve video resim dokusunun elde edilmesi gibi alanlarda kullanılmıştır.



Şekil 4.2. 2B ve 3B Çizgede Kesme. [Paragios et. al., 2005.]

Hiperyüzey çıkartma yeteneği ile çizge kesme yöntemi ilk uygulama alanlarından birini stereo probleminde bulmuştur. Bu konuda S. Roy ve I. Cox 1998 yılında n-kamera stereo problemi için eşleme probleminde ve yine onlarla hemen hemen aynı zamanda H. Ishikawa ve D. Geiger stereo problemindeki kapanmalar ve süreksizlikleri çizge kesme yöntemi ile çözmeye çalışmışlardır. Roy ve Cox yukarıdaki şekle benzer bir yapı düşünmüşlerdir. Bu yapıda kesme aralığı olası hiperyüzeyi verecektir. Bu algoritmanın bir sorunu kesmeler sırasında oluşabilecek katlanmalardır. Bu katlanmaları engellemek için ise çizgenin yönlü yapısı kullanılır ve bulunduğu bölüme göre sonsuz maliyetli n-bağları ile bağlantı kurularak katlanmalar engellenir.



Şekil 4.3. Çizgede Resimlerin Konumları ve Oluşabilecek Hatalı Kesme. [Paragios et. al., 2005.]

Ishikawa ve Geiger ise daha farklı bir yapıda düşünmüşler ve sol resimdeki p=(x,y) sağ resimdeki q=(f(x,y),y) noktası ile eşleşir. Ishikawa ve Geiger makalelerinde sıralılık kısıdı kullanmışlar ve bu sayede katlanmaların önüne geçmişlerdir.

4.2 Çizge Kesme Algoritmalarının Kesin Sonuç Verdiği Durumlar

Greig iki etiket için çizge kesme algoritmasının doğru sonuç vereceğini ispatlamıştır fakat birçok bilgisayarla görme problemi çok etiketlidir. İki etiket için optimallığı ispatlanmış olan çizge kesme algoritmaları kullanım alanlarının dar olduğu gerekçesi ile uzun zaman rağbet görmemiştir.

Greig'in bu makalesinden sonra Ishikawa ve arkadaşları çok etiketli problemler için kesin sonuç veren bir yapıyı şu şekilde tanımlamıştır: P piksellerin kümesini, L bu piksellere karşılık gelebilecek olan etiketleri (fark/ hareket vektörü / yoğunluk / vb.) göstersin. Buradaki iş P piksellerine uygun L etiketlerini atayacak f etiketlemesini bulmaktır. f tüm P kümesindeki piksellerin atamalarını göstersin, f_p ise sadece p noktasının atamasını göstersin.

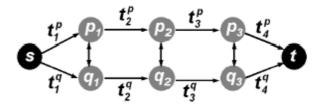
Etiketler tamsayı ve lineer olarak dağılmış olsun. Bu durum hareket problemleri için geçerli değildir çünkü etiketlerin lineer olarak dağılması söz konusu olamaz.

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} \lambda_{pq} |f_p - f_q| \text{ ve } L = \{1,...,k\}, L \in Z^+$$
 (4.2.)

Örnek olarak 4 farklı etiket içeren bir çizgeyi oluşturulsun. G=(V,E) akış ağı ve $\{s,t\} \in V, P = V - \{s,t\}$ olsun. Bu durumda $t_1^p = (s,p_1), t_j^p = (p_{j-1},p_j)$ ve $t_k^p = (p_{k-1},t)$ olur. $t_j^p = (p_{j-1},p_j)$ kenar ağırlıkları

$$K_p + D_p(j)$$
, $K_p = 1 + (k-1) \sum_{q \in N_p} \lambda_{pq}$ şeklinde yazılabilir. G çizgesi üzerindeki

bir C kesmesi en az bir tane t_j^p içerecektir. Eğer ağırlıklar yeteri kadar büyük seçilirse, kesmenin iki tane t_j^p içermesi engellenmiş olur. Bu kesme bir s-t kesme ise asgari kesmedir ve en iyi sonucu verecektir.



Şekil 4.4. Dört Etiket İçin Enerji Formülünün Çizgesi. [Paragios et. al., 2005.]

Ishikawa 2003 yılında yazdığı "Exact optimization for Markov Random Fields with convex priors" isimli makalesinde tüm konveks V_{pq} fonksiyonları için yukarıdaki çizgede kesin sonuç bulunabileceğini belirtmiştir. Buradaki sıkıntı konveks fonksiyonun küresel olarak düzgünlük sağlaması ve parçalı düzgünlük ya da bir başka deyişle nesne sınırlarındaki atlamalara izin vermemesidir. Bu nedenle nesneler arasındaki farklar bir filtreden geçirilmiş gibi olacak ve kesin derinlik bilgileri elde edilemeyecektir [Ishikawa, 2003].

Çizge kesme algoritmalarında kesin sonuç elde etmek için bulunan yöntemlerin ya süreksizlikleri filtrelemesi ya da sadece iki etiket için çalışıyor olması çizge kesme algoritmalarında yaklaşımsal algoritmaları ön plana çıkartmıştır. Bu yaklaşımsal algoritmalar birçok farklı enerji teriminin bir arada kullanılmasına olanak vererek oldukça başarılı sonuçlar elde etmişlerdir.

4.3 Çizge Kesme Algoritmalarında Yaklaşımsal

Yöntemler

Kesin sonuç bulan çizge kesme algoritmalarının sadece iki etiket ya da konveks düzgünlük fonksiyonları ile çalışıyor olması uygulama alanını oldukça daraltmaktaydı. Konveks fonksiyonlar V_{pq} fonksiyonun çok yüksek değerler alabilmesine izin verdiklerinden süreksizliğin olduğu noktaları koruyamazlar. Bu durumu gidermek için V_{pq} fonksiyonunun sınırlanması gerekir. Bu sınırlamayı yapmak için konveks olmayan fonksiyonlar kullanılabilir. Konveks olmayan fonksiyonlar süreksizlik olan bölgelerde daha iyi çalışmalarına rağmen hesaplanmaları konveks fonksiyonlara göre çok daha zordur. Çizge kesme algoritmalarında sıklıkla kullanılan Potts modeli hem oldukça hızlı hesaplanabilen hem de süreksizlikleri koruyan istatistiksel bir modeldir. Potts modelinde $V_{pq} = \lambda_{pq} I(f_p \neq f_q)$ olarak belirlenebilir. Bu modelin çözülmesi ise NP-hard problemdir fakat çizge kesme yöntemi ile optimum noktanın belli bir sınırında yaklaşık çözüm bulmak mümkündür.

Boykov, Veksler ve Zabih "Fast Approximate Energy Minimization via Graph Cuts" makalelerinde iki tane farklı yaklaşımsal çözüm önermişlerdir. Bu yöntemlerin süreksizlikleri koruyarak çalışmaları için α -genişleme algoritması metrik, $\alpha\beta$ -değiştirme algoritması ise yarı metrik iki tane düzgünlük fonksiyonuna ihtiyaç duyar.

Eğer $V(\alpha,\beta) \Leftrightarrow \alpha=\beta$ ve $V(\alpha,\beta)=V(\beta,\alpha)\geq 0$ şartları herhangi $\{\beta,\alpha\}\in L$ için sağlanıyorsa $V(\alpha,\beta)$ yarı-metrik $V(\alpha,\beta) \Leftrightarrow \alpha=\beta$, $V(\alpha,\beta)=V(\beta,\alpha)\geq 0$ ve $V(\alpha,\beta)\leq V(\alpha,\gamma)+V(\gamma,\beta)$ şartları herhangi $\{\beta,\alpha,\gamma\}\in L$ için sağlanıyorsa $V(\alpha,\beta)$ metrik bir fonksiyondur.

Yaklaşımsal yöntem olarak duyurulan yöntemlerden ilki $\alpha\beta$ -değiştirme algoritmasıdır. Tüm piksellerinin ve onlarla eşleşen etiketlerinin oluşturduğu durum f ile gösterilir. $\alpha\beta$ -değiştirme algoritmasında f durumunda bulunan piksellerden α

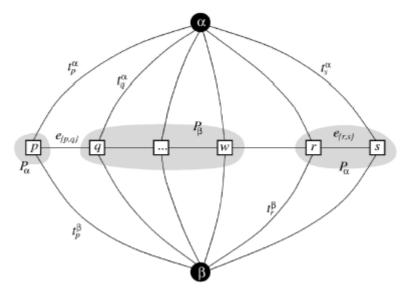
etiketi taşıyanlarla β etiketi taşıyan piksellerin bir kısmının etiketlerini birbirleriyle değiştirerek enerji asgarileştirmeye çalışılır. Bu durum yinelemeli olarak tüm $\alpha\beta$ çiftleri için rasgele olarak yapılır. $\alpha\beta$ -değiştirme algoritması için $V(\alpha,\alpha)+V(\beta,\beta)\leq V(\alpha,\beta)+V(\beta,\alpha)$ olmalıdır. Bunun sağlanması içinse yarı-metrik $V(\alpha,\beta)=min(K,|\alpha-\beta|^2)$ fonksiyonu kullanılabilir.

Yaklaşımsal yöntemlerden ikincisi olan α -genişleme yönteminde ise tüm noktalar ya kendi mevcut değerlerini koruyarak ya da α değeri alarak enerjiyi asgarileştirmeye çalışacaklardır. $V(\alpha,\alpha)+V(\beta,\gamma)\leq V(\alpha,\gamma)+V(\beta,\alpha)$ olması için kullanılabilecek metrik $V(\alpha,\beta)$ fonksiyonu ise $V(\alpha,\beta)=\min(K,|\alpha-\beta|)$ şeklinde ya da Potts model olabilir. Bu model ile birlikte kullanıldığında α -genişleme algoritması oldukça başarılı sonuçlar vermekte ve birçok makalede tercih edilmektedir.

4.3.1 αβ-Değiştirme Algoritması ve En İyi αβ-Değişimini Bulma

Yaklaşımsal yöntem olarak duyurulan yöntemlerden ilki $\alpha\beta$ -değiştirme algoritmasıdır [Boykov et. al., 1999]. Bu algoritmada f durumunda bulunan piksellerden α etiketi taşıyanlarla β etiketi taşıyan bazı pikseller etiket değiştirerek enerjiyi asgarileştirmeye çalışırlar. Bir $\alpha\beta$ -değiştirme sonucunda f etiketi daha az enerjiye sahip \hat{f} etiketi haline getirilmeye çalışılır. Bu durum yinelemeli olarak tüm $\alpha\beta$ çiftleri için rasgele olarak yapılır.

αβ-değiştirme algoritması için $V(\alpha,\alpha)+V(\beta,\beta)\leq V(\alpha,\beta)+V(\beta,\alpha)$ olmalıdır. Bunun sağlanması için yarı metrik $V(\alpha,\beta)=min(K,|\alpha-\beta|^2)$ fonksiyonu kullanılabilir.



Şekil 4.5. αβ-Değiştirme Algoritması İçin Oluşturulmuş Çizge. [Boykov et. al., 2001.]

 α β-değiştirme algoritmasında çizge o an ki α ve β değerlerine göre dinamik olarak oluşturulacaktır. Yukarıdaki örnek resim kolaylık olması için tek boyutlu olarak gösterilmiştir. Örnek çizgede $P_{\alpha}=\{\mathrm{p,r,s}\}$ ve $P_{\beta}=\{\mathrm{q,...,w}\}$ olarak verilmiştir. Çizgedeki t-bağlarının ve n-bağlarının ağırlıkları ise şu şekilde hesaplanmıştır.

Çizelge 4.1. αβ-Değiştirme Algoritmasında Kenar Ağırlıkları.

Kenar	Ağırlık	Açıklama
t_p^{lpha}	$D_p(\alpha) + \sum_{q \in N_p, q \notin P_{\alpha\beta}} (\alpha, f_q)$	$p \in P_{aeta}$
t_p^{eta}	$D_p(\beta) + \sum_{q \in N_p, q \notin P_{a\beta}} V(\beta, f_q)$	$p \in P_{\alpha\beta}$
$e_{\{p,q\}}$	$V(\alpha, \beta)$	$\{p,q\} \in N$ $p,q \in P_{\alpha}\beta$

αβ-Değiştirme Algoritması

- 1. rasgele bir f etiketlemesi ile başla
- 2. Başarı = 0;
- 3. Her $\{\alpha, \beta\} \subset L$ çifti için yap
 - 3.1 f etiketlemesinden bir $\alpha\beta$ -değiştirme uzaklıktaki f' etiketlerinden en küçük enerjili olan $\bar{f} = minE(f')$ etiketlemesini bul.
 - 3.2 Eğer $E(\bar{f}) \le E(f)$ ise $f = \bar{f}$ ve Başarı=1 olarak ata.
- 4. Eğer Başarı=1 ise 2. Adıma git.
- 5. f etiketini döndür.
- 3. Adıma bir yineleme denir ve 3.1 adımı algoritmanın en can alıcı yeridir. Buradaki hesabın doğru olup olmaması sonucu belirler. Enerjide azalma olduğu sürece bu yineleme devam eder. $\alpha\beta$ -Değiştirme algoritmasında 3.1 ve 3.2 numaralı adımlar $O(L^2)$ kere işlenir. Yineleme sayısı ise teorik olarak piksel sayısı ile sınırlıdır ama pratikte bir kaç yineleme sonunda enerji yerel en küçük halini alır. Enerjideki en büyük azalma ise ilk yinelemede gerçekleşir. Bu nedenle testte ilk yineleme sonunda gerçek sonuçlara ne kadar yaklaşıldığı da belirtilecektir.

 $G_{a\beta}$ ağındaki her $p \in P_{a\beta}$ pikseli için C kesmesi içinde sadece tek bir t-bağı bulunabilir.

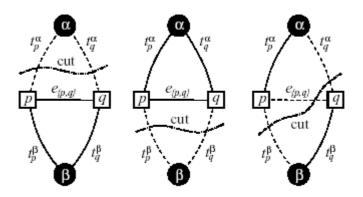
$$f_{p}^{C} = \begin{cases} \alpha & eger \quad p \in P_{\alpha\beta} \quad i\varsigma in \quad t_{p}^{\alpha} \\ \beta & eger \quad p \in P_{\alpha\beta} \quad i\varsigma in \quad t_{p}^{\beta} \\ f_{p} & eger \quad p \in P \quad ve \quad p \notin P_{\alpha\beta} \end{cases}$$

Eğer p pikseli $p \in P_{\alpha\beta}$ ise bu nokta bir sonraki durumda ya α ya da β etiketi alacaktır. Eğer t_p^{α} bağı kesilmişse nokta α , t_p^{β} bağı kesilmişse nokta β etiketi alır. Eğer p pikseli $p \notin P_{\alpha\beta}$ ise önceki etiketini korumaya devam edecektir.

Önsav–4: $G_{\alpha\beta}$ üzerindeki bir C kesmesini gösteren f^C etiketlemesi, f etiketlemesinden sadece bir αβ-değiştirme kadar uzaktır.

N-bağlantıları için ise aşağıdaki durum söz konusu olacaktır. Herhangi bir C kesmesi ve $e_{\{p,q\}}$ n-bağlantısı için:

- 1. Eğer $t_p^{\alpha}, t_q^{\alpha} \in C$ ise $e_{\{p,q\}} \notin C$
- 2. Eğer $t_p^{\beta}, t_q^{\beta} \in C$ ise $e_{\{p,q\}} \notin C$
- 3. Eğer $t_p^{\alpha}, t_q^{\beta} \in C$ ise $e_{\{p,q\}} \in C$
- 4. Eğer t_q^{β} , $t_p^{\alpha} \in C$ ise $e_{\{p,q\}} \in C$



Şekil 4.6. αβ-Değiştirme Algoritmasında Oluşabilecek s-t Kesmeleri. [Boykov et. al., 2001.]

Önsav–5: Herhangi bir C kesmesi ve herhangi $e_{\{p,q\}}$ n-bağlantısı için

$$\left| C \cap e_{\{p,q\}} \right| = V\left(f_p^C, f_q^C\right)$$

İspat: $|C \cap e_{\{p,q\}}| = |e_{\{p,q\}}| = V(\alpha, \beta)$ ve $f_p^C = \alpha, f_q^C = \beta$ olduğundan eşitlik doğrudur. Burada V() fonksiyonunun yarı metrik olduğu düşünülmektedir.

Teorem–2: $G_{\alpha\beta}$ üzerindeki C kesmesi ile f etiketlemesinden bir αβ-değiştirme kadar uzaklıktaki etiketlemeler arasında bire bir ilişki vardır. Bu nedenle de

 $G_{\alpha\beta}$ üzerindeki C kesmesinin maliyeti, f etiketlemesinden bir $\alpha\beta$ -değiştirme kadar uzakta bulunan f^{C} etiketlemelerden herhangi birinin enerjisi ile bir sabitin toplamına eşittir.

$$|C| = E(f^C) + Sabit$$

ISPAT:
$$|C| = \sum_{p \in P_{\alpha\beta}} |C \cap \{t_p^{\alpha}, t_p^{\beta}\}| + \sum_{\{p,q\} \in N_p, \{p,q\} \notin P_{\alpha\beta}} |C \cap e_{\{p,q\}}|$$

$$p \in P_{a\beta} \text{ için } \left| C \cap \{t_p^a, t_p^\beta\} \right| = D_p \left(f_p^C\right) + \sum_{\{p,q\} \in N, q \notin P_{a\beta}} V \left(f_p^C, f_q\right)$$

Önceki önsavdan;

$$\left|C\right| = \sum_{p \in \rho_{\alpha\beta}} D_p \left(f_p^C\right) + \sum_{p \in \rho_{\alpha\beta}} \sum_{q \in N_p} V \left(f_p^C, f_q\right) + \sum_{\{\mathbf{p}, \mathbf{q}\} \in N, \{\mathbf{p}, \mathbf{q}\} \subset \rho_{\alpha\beta}} V \left(f_p^C, f_q^C\right)$$

$$|C| = \sum_{p \in \rho_{\alpha\beta}} D_p(f_p^C) + \sum_{\{p,q\} \in N, p \text{ or } q \in \rho_{\alpha\beta}} V(f_p^C, f_q^C)$$

bu eşitlik $|C| = E(f^C) - K$ şeklinde yazılabilir.

$$K = \sum_{p \notin \rho_{\alpha\beta}} D_p (f_p) + \sum_{\{p,q\} \in N, \{p,q\} \cap \rho_{\alpha\beta} = \emptyset} V (f_p, f_q)$$

tüm C kesmeleri için aynı değere sahip bir sabit olur.

4.3.2 α-Genişleme Algoritması ve En İyi α-Genişlemeyi Bulma

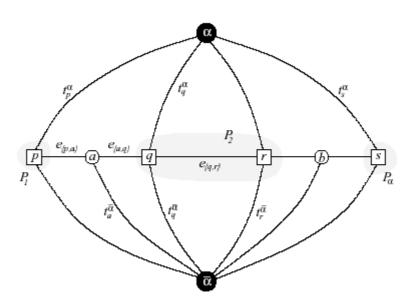
 α -genişleme algoritmasında verilen bir başlangıç f etiketlemesi için tüm noktaları α etiketine götürülmeye çalışılır [Boykov et. al., 1999]. Elde edilecek yeni

f' etiketlemesinde önceki f etiketlemesinde α etiketi taşımayan bazı pikseller artık α etiketi taşıyor, diğerleri ise eski etiketlerini koruyor olacaklardır.

Çizelge 4.3. α-Genişleme Algoritması.

α -Genişleme Algoritması

- 1. Rasgele bir f etiketlemesi ile başla
- 2. Başarı = 0;
- 3. Her $\alpha \in L$ etiketi için yap
 - 3.1 f etiketlemesinden bir α -genişleme uzaklıktaki f' etiketlerinden en küçük enerjili olan $\bar{f} = minE(f')$ etiketlemesini bul.
 - 3.2 Eğer $E(\bar{r}) \le E(f)$ ise $f = \bar{r}$ ve Başarı=1 olarak ata.
- 4. Eğer Başarı=1 ise 2. Adıma git.
- 5. f etiketini döndür.



Şekil 4.7. α-Genişleme Algoritması Tek Boyutlu Çizge Örneği. [Boykov et. al., 2001.]

Bu çizge yapısında iki önemli nokta vardır. Bunlardan ilki hedef ve kaynak noktaların seçimidir. α -genişleme algoritmasında oluşan çizgenin hedef noktası yeni etiket olan α etiketi iken kaynak noktasının etiketi olan $\bar{\alpha}$ ise f etiketlemesindeki

eski etiket değerini verir. İkinci dikkat edilmesi gerek nokta ise farklı etiketler arasında oluşturulan yardımcı noktalardır. Örneğin bu yukarıdaki çizgede a ve b uçları farklı etiketlere sahip bölgeler arasında oluşturulmuş yardımcı noktalardır. Bu yardımcı noktalar α-genişleme algoritmasının ilk uygulamasında [Boykov et. al., 1999] kullanılmıştır fakat daha sonra algoritmanın daha genelleştirilmiş bir uygulaması ile bu yardımcı düğümlere olan gereksinim ortadan kalkmıştır [Kolmogorov and Zabih, 2002].

$$V_{\alpha} = \{\alpha, \overline{\alpha}, \rho, \bigcup_{\{\mathbf{p},\mathbf{q}\} \in N, f_{p} \neq f_{q}} a_{\{\mathbf{p},\mathbf{q}\}}\}$$

Çizgedeki kenarların ağırlıklarını ise aşağıdaki çizelgede olduğu gibi verebiliriz.

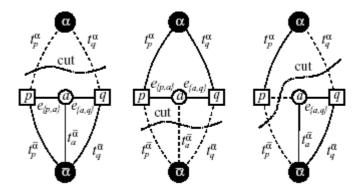
Çizelge 4.4. α-Genişleme İçin Kenar Ağırlıklarının Hesaplanması.

Kenar	Ağırlık	Açıklama
$t_p^{\overline{lpha}}$	œ	$p\in ho_{lpha}$
$t_p^{\overline{lpha}}$	$D_p(f_p)$	$p \notin \rho_{\alpha}$
t_p^a	$D_{p}(\alpha)$	$p \in \rho$
$e_{ m \{p,a\}}$	$V(f_p,\alpha)$	
$e_{ ext{\{a,q\}}}$	$V(\alpha, f_q)$	$\{p,q\} \in N, f_p \neq f_q$
$t_a^{\overline{lpha}}$	$V(f_p, f_q)$	
$e_{ m \{p,q\}}$	$V(f_p,\alpha)$	$\{p,q\} \in N, f_p = f_q$

Önsav–6: G_{α} ağındaki bir C kesmesiyle eşleşen f^{C} etiketlemesi f etiketlemesinden bir α -genişleme kadar uzaktır.

Özellik: Eğer p ve q noktaları komşu $\{p,q\} \in N$ ve $f_p \neq f_q$ ise G_a ağıdaki en küçük kesme C aşağıdakileri sağlar:

- a) Eğer t_p^{α} , $t_q^{\alpha} \in C$ ise $C \cap \varepsilon_{\{p,q\}} = \emptyset$
- b) Eğer $t_p^{\overline{a}}, t_q^{\overline{a}} \in C$ ise $C \cap \varepsilon_{\{p,q\}} = t_a^{\overline{a}}$
- c) Eğer $t_p^{\overline{a}}, t_q^{a} \in C$ ise $C \cap \varepsilon_{\{p,q\}} = e_{\{p,a\}}$
- d) Eğer $t_p^a, t_q^{\overline{a}} \in C$ ise $C \cap \varepsilon_{\{p,q\}} = e_{\{a,q\}}$



Şekil 4.8. α-Genişleme Algoritmasında Oluşabilecek s-t Kesmeleri. [Boykov et. al., 2001.]

Önsav–7: Eğer p ve q noktaları komşu $\{p,q\} \in N$ ve $f_p \neq f_q$ ise G_α ağıdaki en küçük kesme C aşağıdaki eşitliği sağlar.

$$|C \cap \varepsilon_{\{p,q\}}| = V(f_p^C, f_q^C)$$

Teorem–3: G_{α} ağı verilen f ve α değerleri için çizelgede verilmiş ağırlıklara göre oluşturulmuş olsun. Bu durumda G_{α} ağıdaki kesmeler ile f etiketlemesinden bir α -genişleme kadar uzaklıkta olan diğer etiketlemeler arasında bir ilişki vardır. Bu nedenle kesmenin maliyeti bu kesmeye ait en küçük enerjili etiketlemenin enerjisine eşittir.

$$|C| = E(f^C)$$

4.3.3 α-Genişleme Algoritmasının Optimallık Analizi

Bu bölümde α -genişleme algoritmasının en iyi sonuç ile arasındaki hata arasında belli bir oran olduğunu gösterilmiştir [Boykov et. al., 1999]. Bu orandan daha fazla bir hata oluşmayacaktır. $\alpha\beta$ -Değiştirme algoritması için ise bu şekilde bir sınır koymak mümkün olmamaktadır.

Küresel en iyi ile yerel iyi arasındaki hata faktörü için;

$$c = max_{p,q \in N} \left(\frac{max_{\alpha \neq \beta \in L} V(\alpha, \beta)}{min_{\alpha \neq \beta \in L} V(\alpha, \beta)} \right)$$

Teorem 4: \hat{f} en iyi yerel sonuç ve f^* küresel en iyi sonuç olsun. Bu durumda

$$E(\hat{f}) \le 2cE(f^*)$$

olacaktır.

İspat: $\alpha \in L$ ve $\rho_{\alpha} = \{ p \in \rho \mid f_p^* = \alpha \}$ olsun.

Bir α -genişleme sonucunda \hat{f} etiketlemesinden f^{α} elde edilsin:

$$p \in \rho_{\alpha}$$
 ise $f_{\alpha}^{p} = \alpha$

$$p \notin \rho_{\alpha}$$
 ise $f_{\alpha}^{p} = f_{p}$ olur.

 $\hat{\sl}$ α -genişleme Hareketlerinin izin verdiği yerel en küçük nokta olsun. Bu durumda

$$E(\hat{f}) \leq E(f^{\alpha})$$

S herhangi sayıda piksel içeren ve $S \subset \rho$ olan bir küme olsun. f etiketlemesinin S kümesine göre sınırlanmış hali $E(f \mid S)$ şeklinde gösterilir.

$$E(f \mid S) = \sum_{p \in S} D_p(f_p) + \sum_{\{p,q\} \in S} V(f_p, f_q)$$

 I^{α} : ρ_{α} içindeki pikseller ve komşularını göstersin.

 $B^{\scriptscriptstyle \alpha}$: $\rho_{\scriptscriptstyle \alpha}$ nın sınırında bulunan komşu pikselleri göstersin.

 O^a : ρ_a nın dışında bulunan pikseller ve onların komşularını göstersin. Tüm bunlar matematiksel olarak şu şekilde ifade edilebilir:

$$I^{\alpha} = \rho_{\alpha} \cup \{\{p,q\} \in N : p \in \rho_{\alpha}, q \in \rho_{\alpha}\}\$$

$$B^{\alpha} = \{ \{ \mathbf{p}, \mathbf{q} \} \in N : p \in \rho_{\alpha}, q \notin \rho_{\alpha} \}$$

$$O^{\alpha} = (\rho - \rho_{\alpha}) \cup \{\{p, q\} \in N : p \notin \rho_{\alpha}, q \notin \rho_{\alpha}\}\}$$

Bu tanımlardan ve f etiketlemesi ile ilgili kurallardan:

$$E(f^{\alpha} \mid O^{\alpha}) = E(\hat{f} \mid O^{\alpha})$$

$$E(f^{\alpha} | I^{\alpha}) = E(f^* | I^{\alpha})$$

$$E(f^{\alpha} \mid B^{\alpha}) \leq c.E(f^* \mid B^{\alpha})$$

$$E(\hat{f} \mid I^{\alpha}) + E(\hat{f} \mid B^{\alpha}) + E(\hat{f} \mid O^{\alpha}) \le E(f^{\alpha} \mid I^{\alpha}) + E(f^{\alpha} \mid B^{\alpha}) + E(f^{\alpha} \mid O^{\alpha})$$

$$E(\hat{f} \mid I^{\alpha}) + E(\hat{f} \mid B^{\alpha}) \leq E(f^* \mid I^{\alpha}) + c.E(f^* \mid B^{\alpha})$$

Yukarıda elde edilen eşitsizlik toplam enerji formülünde yerine konulur ve her α etiketi için bu toplam hesaplanır.

$$\sum_{\alpha \in L} \left(E(f \mid I^{\alpha}) + E(f \mid B^{\alpha}) \right) \leq \sum_{\alpha \in L} \left(E(f^* \mid I^{\alpha}) + c.E(f^* \mid B^{\alpha}) \right)$$

 $B = \bigcup_{\alpha \in L} B^{\alpha}$ tüm $\alpha \in L$ etiketleri için B^{α} piksellerinin birleşimi olsun.

$$E(\hat{f}) + E(\hat{f} \mid B) \le E(f^*) + (2c - 1)E_B(f^*) \le 2cE(f^*)$$

Bu durumda optimal sonuçtan sapma bir sabitin iki katına eşit yada daha küçük olacaktır.

4.4 Kapanma Durumlarını Çizge Kesme Yöntemiyle Cözmek

Bir önceki bölümde anlatılan α-genişleme ve αβ-değiştirme algoritmalarında oluşturulan çizgeler kapanma durumlarını dikkate almıyorlardı. Kapanma durumları stereo problemi için önemli sorunlardan biridir. Kapanma problemleri genellikle nesnelerin kenarlarında oluşur ve bu alanların gerçek fark değerlerinin bulunmasını engeller. Genel olarak kapanma sorunu bir kameranın gördüğü bir alanın diğer kamera tarafından görülememesi olarak tanımlanır. Bu durumdaki alanın eşlenmesi ya hatalı olacaktır ya da mümkün olmayacaktır. Bu bölümde anlatılan algoritma çapraz kontrol yöntemi ile kapanma olabilecek alanları tespit edip bu alanlara ana enerji fonksiyonunda müdahale etmektedir.

$$E(f) = E(f)_{veri} + E(f)_{diizgiinliik} = \sum_{p \in L} D^p(f^p) + \sum_{p,q \in N} V_{p,q} T(f_p \neq f_q)$$
(4.3.)

Önceki anlatılan algoritmada (4.3.) şeklinde olan enerji fonksiyonu aşağıdaki sekle getirilip, kapanma enerjisi eklenmiştir [Kolmogorov and Zabih, 2001].

$$E(f) = E(f)_{veri} + E(f)_{dizgünlük} + E(f)_{kapanma}$$

$$E(f) = \sum_{p \in L} D^{p}(f^{p}) + \sum_{p,q \in N} V_{p,q} T(f_{p} \neq f_{q}) + \sum_{p \in P} C_{p} T(N_{p}(f) = 0)$$

4.4.1 Kapanmaları Dikkate Alan Çizge Kesme Algoritmaları

Kolmogorov ve Zabih tarafından geliştirilen bu algoritmalar da aynı önceki bölümde tarif edildiği gibi Boykov, Veksler ve Zabih tarafından geliştirilmiş $\alpha\beta$ -değiştirme ve α -genişleme algoritmaları gibi işleyecektir. Bu algoritmanın temel farkı hesaplanacak enerji fonksiyonu ve oluşacak ara çizgelerinin yapılarındaki değişimdir.

Bu hesaplamalarının yapılabilmesi için öncelikle problemin biraz daha farklı bir şekilde yeniden tanımlaması gerekir.

Stereo problemi şu şekilde tanımlanabilir: L sol resimdeki tüm piksellerin, R sağ resimdeki tüm piksellerin ve ρ her iki resimdeki piksellerin kümesi olsun. Bu durumda $\rho = L \cup R$ olacaktır. p pikseli p_x, p_y koordinatlarına sahip olacaktır. A kümesi ise - sırasız - birbirleriyle eşleşebilme olasılığı olan piksel çiftlerinin kümesi olsun. Eğer fark kısıdı kullanılır ve bir pikselin en fazla k piksel kadar uzaktaki bir başka pikselle eşleşeceği söylenebilirse bu durumda s(A) = kL kadar olacaktır.

$$A = \{ < p, q > | p_y = q_y \text{ ve } 0 \le q_x - p_x \le k \}$$

A kümesi bu şekilde tanımladığında amaç A kümesinin alt kümesi olan ve sadece eşlenmiş piksel çiftlerini gösteren bir küme bulmak olacaktır. Bunun için A kümesinin her bir elemanına 1 ya da 0 olarak değer atanır. 1 değerine sahip çiftler birbirlerine atanmış ve "aktif" olarak adlandırılan çiftlerdir. 0 değerine sahip çiftler ise eşleşmemiş ve "pasif" olarak adlandırılan çiftlerdir. A(f) f etiketlemesine göre f kümesi içindeki aktif elemanların kümesi olsun. Eğer her bir piksel sadece bir kez

aktif bir çiftin içinde yer alıyorsa buna "eşsiz yapılandırma" adı verilir. A^{α} , A içindeki α etiketine sahip çiftlerinin kümesi, A^{0} ise f_{0} etiketlemesindeki çiftlerin kümesi olsun.

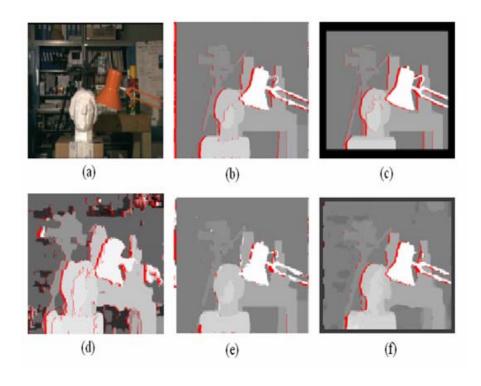
Bu durumda önceki bölümde anlatılan çizgelerin yapıları aynı kaldığında, ağırlıkların şu şekilde belirlenmesi gerekecektir.

Çizelge 4.5. αβ-Değiştirme Algoritmasında Kenarların Ağırlıkları.

Kenar	Ağırlık	Açıklama
(s,a)	$D_{\scriptscriptstyle kapanma}(a)$	$a \in A^{\alpha}$
(a,t)	$D_{kapanma}(a)$	$a \in A^{\beta}$
(a,t)	$D_{veri}(a) + D_{d\ddot{u}zg\ddot{u}nl\ddot{u}k}(a)$	$a \in A^{\alpha}$
(s,a)	$D_{veri}(a) + D_{d\ddot{u}zg\ddot{u}nl\ddot{u}k}(a)$	$a \in A^{\beta}$
$(a_{1,}a_{2})$	V_{a_1,a_2}	$\{a_1, a_2\} \in N, a_1 \in A^a, a_2 \in A^\beta$
$(a_{2,}a_{1})$	C_p	$p \in \rho, a_1 \in A^{\alpha}, a_2 \in A^{\beta}, a_{1,\alpha_2} \in N_p(\tilde{f})$

Çizelge 4.6. α-Genişleme Algoritmasında Kenarların Ağırlıkları.

Kenar	Ağırlık	Açıklama
(s,a)	$D_{\scriptscriptstyle kapanma}(a)$	$a \in A^0$
(a,t)	$D_{\scriptscriptstyle kapanma}(a)$	$a \in A^{\alpha}$
(a,t)	$D_{veri}(a) + D_{d\ddot{u}zg\ddot{u}nl\ddot{u}k}(a)$	$a \in A^0$
(s,a)	$D_{veri}(a)$	$a \in A^{\alpha}$
$(a_{1,}a_{2})(a_{2,}a_{1})$	V_{a_1,a_2}	$\{a_1,a_2\} \in N, a_1,a_2 \in \tilde{A}$
$(a_{1,}a_{2})$	∞	$p \in \rho, a_1 \in A^0, a_2 \in A^\alpha, a_{1,a_2} \in N_p(\tilde{f})$
$(a_{2,}a_{1})$	C_p	$p \in \rho, a_1 \in A^0, a_2 \in A^\alpha, a_{1,a_2} \in N_p(\tilde{f})$



Şekil 4.9. (a) Orijinal Sol Resim, (b) KZ2, (c) Gerçek Fark Haritası, (d) Yerel Eşleme, (e) BVZ (f) Zitnick & Kanade Algoritması. [Kolmogorov and Zabih, 2001.]

Algoritmanın sonuçları hem α -genişleme ve $\alpha\beta$ -değiştirme algoritmaları arasındaki farkı gösterirken hem diğer çizge kesme yöntemleri hem de kapanma durumlarını ele alan çizge kesme yöntemi dışındaki algoritmalarla da bir karşılaştırma sunmaktadır.

Çizelge 4.7. KZ1 ve Diğer Algoritmaların Karşılaştırılması.

Yöntem	Hata	Büyük Hata	Negatif Yanlış	Pozitif Yanlış
α-Genişleme	6,7 %	1,9 %	42,6 %	1,1 %
αβ-Değiştirme	20,7 %	13,6 %	50,6 %	3,4 %
Boykov, Veksler & Zabih	6,7 %	2,0 %	82,8 %	0,3 %
Zitnick & Kanade	12,0 %	2,6 %	52,4 %	0,8 %
Yerel Eşleme	28,5 %	12,8 %	87,3 %	6,1 %

α-genişleme algoritması diğer çizge kesme algoritması olan Boykov,

Veksler & Zabih α-genişleme algoritmasından daha az negatif yanlış üretmiştir. Zitnick & Kanade tarafından geliştirilen yerel ve küresel yöntemleri birlikte kullanarak kapanmaları bulmaya çalışan algoritmadan ise neredeyse iki kat kadar daha az hata üretmiş, negatif yanlışlarla bir pikselden büyük hatalarda da iyileşme sağlamıştır.

Algoritma λ katsayısından görece bağımsız olarak çalışmaktadır. Bu katsayının belirlenmesi sonuç üzerinde çok büyük oranlarda etki göstermemektedir. Algoritmalarda kullanılan istatistiksel yöntem ile resimdeki renk değişimlerine ve gürültüye göre parametreler ayarlanabilmektedir. Bu yöntem sayesinde bir çok stereo algoritmasının sorunu olan parametre bağımlılığı büyük ölçüde aşılmış olmaktadır.

Çizelge 4.8. Algoritmanın λ Katsayısına Bağımlılığı.

λ	1	3	10	30
Hata	10,9 %	6,7 %	9,7 %	11,1 %
Büyük Hata	2,4 %	1,9 %	3,1 %	3,6 %
Negatif Yanlış	42,2 %	42,6 %	48,0 %	51,4 %
Pozitif Yanlış	1,4 %	1,1 %	1,0 %	0,8 %

4.5 Dilimleme Tabanlı Çizge Kesme Algoritmaları

Çizge kesme yöntemi ile dilimlemeyi bir arada kullanarak oluşacak çizgelerin karmaşıklığını azaltmak hem de düzgünlüğü dilimler içinde birbirine yakın ifade edilebileceğini kabul etmek stereo probleminde oldukça iyi sonuçlar vermişlerdir. Bu alanda önce Hong ve Chen tarafından yazılmış dilimleme tabanlı makale ve daha sonra bu makaleye ek olarak tabakalama yöntemi [Hong and Chen, 2004] ile iyileştirme yapan Bleyer ve Gelautz'un makaleleri bu bölümde incelenecektir [Bleyer and Gelautz, 2005].

4.5.1 Hong ve Chen'in Algoritması

Hong ve Chen tarafından yazılan ilk makalede zor resim bölgeleri için iyi sonuçlar veren dilim tabanlı algoritma anlatılmıştır. Sunulan makale, Tao ve arkadaşlarının eski çalışmalarıyla birlikte, Boykov ve arkadaşları ile Kolmogorov ve Zabih'in çizge kesme algoritmalarından ilham almıştır[Tao et. al., 2003][Boykov et. al., 1999][Kolmogorov and Zabih, 2001][Kolmogorov and Zabih, 2002].

Tao ve arkadaşları homojen renk dilimleri içinde çok büyük fark olamayacağı kabulünü temel alan renk dilimlerine dayalı bir stereo sistem tanımlanışlardır. Ana fikir; referans resim fark değerleri kadar **ötelendiğinde** diğer resmin elde edileceğidir. Bu, stereo eşleme probleminin küresel resim eşleme enerjisinin en aza indirgenmesi ile çözülebileceği anlamına gelir. Renklere göre dilimlere ayırmak ise çok büyük olan çözüm kümesinin azaltılmasını ve homojen alanlardaki düzgünleştirmeyi sağlamaktadır. Hırslı yerel arama mekanizması ile birlikte komşu farklar hipotezinin kullanılması çok maliyetli olan öteleme işleminin maliyetinin azaltılmasını ve eşlenemeyen bölgeler için açıklanabilir farkların oluşmasını sağlar.

Boykov ve arkadaşlarıyla, Kolmogorov ve Zabih verimli çalışan ve düzgün fark haritaları bulan çizge kesme algoritmaları duyurmuşlardır. Her iki yaklaşımda da stereo eşleme problemi içinde:

- 1. Komşu piksel arasındaki düzgünlüğü ölçen düzgünleştirme enerjisi E_s ,
- 2. Kabul edilen farklar için eşleşen pikseller arasında oluşan ihtilafı ölçen veri enerjisi E_d ,
- 3. Stereo problemi sadece Kolmogorov ve Zabih tarafından, kapanmış alanları adresleyen *kapanma enerjisi* $^{E_{occ}}$ de bulunduran enerji azaltma problemi olarak görülüp çizge kesme yöntemi ile çözülmüştür.

Dilim tabanlı yöntemde her iki tekniğin de avantajlarından yararlanılmıştır. İlk olarak referans resim birbirleriyle çakışmayan homojen renk dilimlerine ayrılmış ve sahne yapısı yaklaşımsal olarak fark uzayında düzlemlerin kümesi olarak

gösterilmiştir. Stereo eşleme problemi, eşleşen fark düzlemini bir dilime atamak haline gelmiş ve dilim alanında kolayca enerji azaltma problemi olarak formülleştirilebilmiştir. Daha özelde enerji fonksiyonu iki kısım içermektedir:

Komşu dilimler arasındaki düzgünlüğü ölçen düzgünleştirme enerjisi E_{smooth}

Varsayılan fark düzlemi temelinde dilimler ve onların eşleşen alanları arasındaki ihtilafı ölcen *veri enerjisi* $E_{\it data}$

 E_s ve E_d , E_{smooth} ve E_{data} enerjilerinin sadeleştirilmiş halleridir. Her homojen renk dilimi içindeki farkın aynı olacağı kabul edildiğinde E_s sadece değişik dilimler arasındaki farkı ölçmek için kullanılacaktır. Her bir dilim ile onun eşleştiği bölgenin ihtilafını dilimin içindeki tüm piksellerin ihtilafları toplamı olarak göstermek oldukça kolaydır. Bu nedenle veri enerjisi E_d tüm resimdeki eşleşen piksellerin ihtilaflarını ölçmektedir.

Enerji fonksiyonunun en uygun sonucu vermesi için çizge kesme tekniğini Boykov ve arkadaşlarınınkine benzer şekilde kullanılmıştır. Burada dikkat edilmesi gereken çizge düğümlerinin artık pikselleri değil dilimleri göstermesidir. Genelde dilim sayısı piksel sayısına göre çok daha küçüktür. Bu küçüklük sade bir çizge yapısı ve kolay hesaplama sağlar. Ek olarak çizge etiketleri fark menzilindeki her bir ayrık değeri içermek yerine sadece düzlemin yaklaşık fark düzlemlerini tutmaktadır. Birchfield ve Tomasi ayrık fark değerleri üzerinde arama yapıldığında korunanın parçalıların-sabitliğinden (piecewise-constancy) ziyade parçaların-sürekliliği (piecewise-continuity) olduğuna işaret etmişlerdir. Bu yaklaşımda parça-sürekliliği her dilimin içinde düzlem gösterimi ile zorlanmaktadır. Bununla birlikte dilimler arasında fark düzgünlestirmesi de E_{smooth} ile sağlanmaktadır. Bu düzgün farklara sahip dokulu alanlar için özellikle önemlidir çünkü bu alanlarda çok fazla sayıda küçük dilimler oluşacaktır. Bu bölgelere düzgünlük kısıdını zorlayarak bu alanlar arasındaki düzgünlük korunmuş olur.

Algoritma, Birchfield ve Tomasi'nin algoritması ile bazı benzerlikler içermektedir. Her iki algoritma da parçaların-sürekliliğini göstermek için düzgün fark fonksiyonu kullanmışlardır. Bununla birlikte Birchfield ve Tomasi dilimlemek için çizge kesme yöntemini kullanırlarken burada sadece renk dilimlemesi yapılmıştır. Birchfield ve Tomasi tarafından yazılan algoritmada dilimlerin doğruluğu fark fonksiyonun doğru tahmin edilmesine bağlı iken bu algoritmada sadece orijinal resme bağlıdır. İyi bir dilimleme yapıldığında daha keskin sınırlar elde edilebilir fakat Birchfield ve Tomasi algoritması kavisli nesnelerin dış hatlarında sapmalar gösterebilmektedir. Birchfield ve Tomasi algoritmasında çizge kesme piksel alanında gerçekleştirilirken, Hong ve Chen'in geliştirdiği algoritma dilim alanında çalışmaktadır.

Renk Dilimleme

Temel yaklaşım, büyük fark değişimlerinin sadece homojen renk dilimlerinin sınırlarında olabileceği kısıdı üzerine kuruludur. Bu nedenle homojen renk bölgelerinin elde edilmesini sağlayan herhangi bir renk dilimleme algoritması bu algoritmanın işini görecektir. Dilimler içinde düzgünlük zorlandığından az sayıda dilimlemeden (under-segmentation) kaçınılmalıdır. Aşırı sayıda dilimleme ise düzgünlük terimi sayesinde giderilmektedir. Makale içindeki gerçekleştirmede ortalama-kaydırma renk dilimleme algoritması kullanılmıştır.

Fark Düzleminin Tahmin Edilmesi

Sahne yapısı farkları sürekli düzlemlerin bir koleksiyonu olarak sunulup, her yüzey ona yaklaşık olan bir düzlem ile ifade edilmiştir. Düzlem yaklaşımdan ötürü bir miktar hassaslık kaybı olacağının not edilmesi gerekir fakat bu basit model birçok uygulama için yeterlidir. Sahne içindeki olası tüm fark düzlemleri aşağıdaki yol takip edilerek bulunabilir. Öncelikle ilk piksel farkını yerel eşleme analizi ile bulup daha sonra çok küçük dilimler hariç her renk dilimi için ilk düzlem parametreleri hesaplanmıştır. En son olarak ise düzlem parametrelerini, kümelenmiş komşu dilimlere eşlemeye çalışarak küçük dilimler bir büyük dilim haline getirilmiş ve sonuç rafine edilmiştir.

Piksel Alanında Yerel Eşleme

Standart doğrultulmuş stereo için, I referans resmindeki p(x,y) pikseli diğer resimde p'(x',y') pikseline x'=x+d(x,y) ve y'=y olacak şekilde eşlenecektir. Buradaki d(x,y), $[d_{min},d_{max}]$ arasında ayrık değerler alabilir ve p(x,y) noktası için farkı verir. c(x,y,d), p(x,y) noktasının d fark ötedeki bir noktaya eşlenmesinin maliyetini veren fonksiyon olsun. Bu makale boyunca 3x3 boyunda bir pencere için ortalama piksel yoğunluk değerlerine göre maliyeti belirlenmiştir.

$$c(x,y,d) = \frac{1}{9} \sum_{i=-1}^{1} \sum_{i=-1}^{1} |I(x+i,y+j) - J(x+i+d,y+j)|$$

Eşleme maliyeti en küçük olan farkı başlangıç farkı $\hat{d}(x,y)$ olarak belirlenmiştir.

Tek Bir Dilimden İlk Düzlemi Uydurmak

İlk farklardan yararlanarak dilimlere düzlem uydurmak daha önce Tao ve arkadaşları tarafından detaylı şekilde tartışılmıştı. Burada sadece özet şeklinde onların yaklaşımı kullanılacaktır ve bu yaklaşım sadece özet olarak anlatılacaktır. Bir düzlem, her dilim için sürekli farkı modellemek için kullanılır. Örneğin:

 $d=c_1x+c_2y+c_3$, p(x,y) pikseli için d farkını ve c_1,c_2,c_3 ise düzlem parametrelerini göstermektedir. c_1,c_2,c_3 , A'nın i. satırı $[x_i \ y_i \ 1]$ ve B'nin i. satırı $d(x_i,y_i)$ olan $A[c_1,c_2c_3]^T=B$ lineer sisteminin en küçük kare çözümüdür. İlk düzlemin uydurulmasından sonra sistem düzlemi güncellemeye başlar. Her iterasyonda pikselin fark değeri uydurulmuş düzlemin izin verdiği aralıkta değiştirilir ve düzlem parametreleri yapılan değişikliklere göre yeniden düzenlenir. Düzlem uydurma algoritmasının kararlılığını arttırmak için bir kaç farklı şema önerilmiştir:

- 1. $A[c_1, c_2, c_3]^T = B$ lineer sisteminin çözümü güvenilir olmayan ilk farklara sahip piksel tarafından saptırılabilir. Bu piksellere aykırı değer denir. Çapraz kontrol yöntemi ile bu noktaları algılanmaya çalışılır ve eğer $\hat{a}(x,y) \neq \hat{a}'(x',y')$ ise bu nokta aykırı değer olarak belirlenir.
- 2. İterasyon sürecinde, ağırlıklandırılmış en küçük kare şeması benimsenmiştir. İlk düzlem parametreleri ${}^A[c_1,c_2c_3]^T=B$ ile hesaplanmakta ve daha sonra iterasyon süresince ${}^A[c_1,c_2c_3]^T=B$ içindeki her eşitliğin ağırlığı pikselin ilk farkının oraya uydurulmuş düzleme olan yakınlığına göre değiştirilmektedir. Örneğin $\omega(\beta_i)=e^{-2*\beta_i}$ ve $\beta_i=\left|c_1x_i+c_2y_i+c_3-\hat{a}(x_i,y_i)\right|$. Düzlem parametreleri düzenlenmiş ağırlıklara göre güncellenir.
- Çok küçük dilimler düzlem tahmini için yeterli veriyi içermedikleri için ihmal edilirler. Bir dilim için hesap edilen düzlem eğer düzlem kümesinde bu düzlem yoksa kümeye eklenir. Düzlem kümesinde dilimlerden daha az sayıda eleman olacaktır.

Gruplanmış Dilimlerden Rafine Edilmiş Düzlem Uydurması

Burada her dilim için en iyi düzlem bulunmuyor fakat resim için olası tüm düzlemler çıkartılmaktadır. Bu nedenle sahne yapısını gösteren fark düzlemlerinin bir kümesini oluşturmak oldukça önemlidir. Bu küme, gruplanmış dilimlerin düzlem uydurmalarının rafinesi sırasında çıkartılmaktadır. Bu yaklaşımın arkasındaki neden; küçük dilimlerin gruplanarak daha güvenilir dilimler oluşturmasıdır. Bunu sağlamak için şu adımlar işlenmektedir:

- Fark düzlem kümesi içindeki her bir düzlem için dilim eşleme maliyeti ölçülür.
- 2. Her dilime en küçük eşleme maliyetini verecek şekilde düzlem tanımlayıcısı atanır.
- 3. Aynı düzlem tanımlayıcısına sahip komşu dilimler birleştirilir.
- 4. Tüm yeni oluşmuş dilimler için yeniden ilk düzlem uydurulur.

2. ve 4. adımlar kolaydır. Her fark düzlemi içim dilim eşleme maliyetinin nasıl ölçüleceği ayrıca bakılması gerekir. Bu işlemi dilimler içindeki piksellerin eşlenme maliyetlerinin toplamı olarak yapmak mümkündür.

$$C(S,P) = \sum_{(x,y)\in S} c(x,y,d)$$

Fakat bu yaklaşımla ilgili çeşitli problemler söz konusudur. İlk olarak, kapanmış pikseller dilim eşleme maliyetini saptırabilir. İkinci olarak ise piksel eşleme maliyetlerinin toplanması eşlemedeki muğlâklıkları giderirken bu sorun dokusuz bölgelerde aynen devam eder.

Bu sorunların giderilmesi için iki çare önerilmiştir. Öncelikle kapanmış olma ihtimali olan pikseller hesaplamanın dışında tutulmuştur. İkinci olarak piksel eşleme maliyetinin toplamına fark düzlemi tarafından desteklenmeyen piksellerin oranını da katılmıştır.

Daha açık ifade etmek gerekirse; sadece dokulu alanlardaki aykırı değerlerin kapanmış noktalar olabileceğini düşünülmüş ve bu aykırı değerler için çapraz kontrol yöntemini uygulanmıştır. Kapatılmış piksellerin aykırı piksel olarak tanımlanan pikseller içinden çıkacağı açıkça görülebilir. Dokusuz alanlarda aykırı değerlere ise iki nedenden ötürü bakılmamıştır: (1) Çapraz kontrol sırasında kapanma olmasa bile hatalı sonuç verebilir, (2) kapanmalar genelde çok dokulu alanların sınırlarında oluşur.

Tüm kapanmamış pikseller için, eğer bir piksel ilk tahmin edilmiş fark düzleminin çok küçük bir komşuluğunda ise o piksel desteklenen, değilse desteklenmeyen piksel olarak adlandırılmıştır.

S dilimindeki kapanmamış piksellerin sayısı n ve S dilimindeki P fark düzlemini destekleyen piksellerin sayısı da s olsun. O bölgesi S dilimindeki kapanmış bölgeyi göstersin. Bu durumda dilim eşleme maliyetini şu şekilde tanımlanabilir:

$$C(S,P) = \sum_{(x,y)\in S-O} c(x,y,d)e^{-\frac{S}{n}}$$

Eşleme fonksiyonu iki kısımdan oluşmaktadır:

- 1. Kapanmamış piksellerin maliyetlerinin toplamı
- 2. Desteklenmeyen piksel oranına göre artan üstel kısım

Dokusuz bölgeler için ilk kısım birçok düzlemin dilime uygun görünmesi sebep olacaktı. İkinci kısım çok fazla sayıda destekleyen piksel sayısı sayesinde bundan kurtulur. Dokulu bölgelerde ise ikinci kısmın ağırlığı bu kadar olmayacaktır ama doğru düzlem ile diğer düzlemler arasındaki eşleme maliyeti farkını arttıracaktır.

Çizge Kesme Yöntemiyle Fark Düzleminin Etiketlenmesi

Etiketleme işlemi için Boykov'un makalesindeki α-genişleme algoritması kullanılmıştır. Tek fark ise R referans resimdeki dilimleri, D ise tahmini fark düzlemlerinin kümesini göstermektedir. Eşlemeler bu iki küme arasında yapılmış ve Potts modeli kullanılmıştır. Genel enerji formülü

$$E(f) = E_{veri}(f) + E_{diizoiinliik}(f)$$

Deneyler

Algorithms		Tsukuba			Sawtooth			Venus		Map	
	B_{all}	B_{untex}	B_{disc}	B_{Pall}	B_{untex}	B_{disc}	B_{all}	B_{untex}	B_{disc}	B_{all}	B_{disc}
Proposed	1.23	0.29	6.94	0.30	0.00	3.24	0.08	0.01	1.39	1.49	15.46
Layered [5]	1.58	1.06	8.82	0.34	0.00	3.35	1.52	2.96	2.62	0.37	5.24
Belief [9]	1.15	0.42	6.31	0.98	0.30	4.83	1.00	0.76	9.13	0.84	5.27
MultiCam [13]	1.85	1.94	6.99	0.62	0.00	6.86	1.21	1.96	5.71	0.31	4.34
GC+occl [11]	1.19	0.23	6.71	0.73	0.11	5.71	1.64	2.75	5.41	0.61	6.05
Improved Coop [6]	1.67	0.77	9.67	1.21	0.17	6.90	1.04	1.07	13.68	0.29	3.65
Symbiotic [4]	2.87	1.71	11.90	1.04	0.13	7.32	0.51	0.23	7.88	0.50	6.54
Var.win. [15]	2.35	1.65	12.17	1.28	0.23	7.09	1.23	1.16	13.35	0.24	2.98
Graph Cuts [2]	1.86	1.00	9.35	0.42	0.14	3.76	1.69	2.30	5.40	2.39	9.35
Multiw Cut [1]	8.08	6.53	25.33	0.61	0.46	4.60	0.53	0.31	8.06	0.26	3.27
Cooperative [16]	3.49	3.65	14.77	2.03	2.29	13.41	2.57	3.52	26.38	0.22	2.37
Bay. diff [7]	6.49	11.62	12.29	1.45	0.72	9.29	4.00	7.21	18.39	0.20	2.49

Şekil 4.10. Hong ve Chen Algoritmasının Karşılaştırılması. [Hong and Chen, 2005.]

P4 2.4Ghz işlemcili makinede ve 384x288 boyutlu resimlerin 3 saniye içinde fark haritası hesaplanmıştır. Map resminde (çok fazla dokulu) kötü sonuç vermesini dilimleme algoritmasının hatalarına bağlanmıştır. Bu resimde diğer algoritmalara göre 7 kat kadar kötü sonuç elde edilmiş. Dilimler içinde arka alan ve ön alan ayrımı yapamadıklarını ve bu ayrımın yapılması ile daha iyi sonuçlar elde edilebileceğini belirtmişlerdir.

Çizelge 4.9. Hong ve Chen Algoritmasında Hata Oranları.

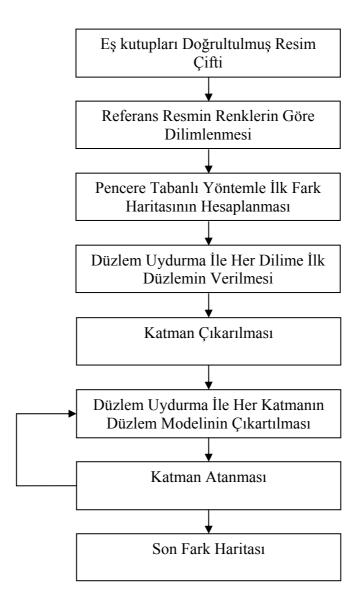
Adım	$HATA_{t\"um\ddot{u}}$	$HATA_{dokusuz}$	HATA _{değişim}
1	39,51	65,96	30,87
2	6,35	6,25	12,30
3	2,61	4,22	8,91
4	0,08	0,01	1,39

Venüs resmi için yukarıdaki verilen çizelgede her bir adımdan sonra hataların değişimi gösterilmiştir.

- 1. Adım: Yerel eşleme yöntemi sonucunda elde edilen hatalar.
- 2. Adım: İlk düzlem uydurulması sonrasında elde edilen hata oranları
- 3. Adım: Dilimlerin gruplanıp, düzlemlerin rafine edilmesi sonrasında elde edilen hatalar
- 4. Adım: Çizge kesme ile enerji en azaltma sonucu elde edilen hatalar.

4.5.2 Bleyer ve Gelautz'un Algoritması

Bleyer ve Gelautz tarafından Hong ve Chen algoritmasını temel alınarak geliştirilmiş olan algoritma test resimleri üzerinde en iyi sonuç veren ikinci küresel algoritma ve çizge kesme algoritması olarak ise birinci en iyi algoritma durumdadır [Bleyer and Gelautz, 2005].



Şekil 4.11. Bleyer ve Gelautz Algoritmasının Genel İşleyişi. [Bleyer and Gelautz, 2005.]

Algoritma iki adet eş kutup çizgilerine göre düzeltilmiş resmi alır, önce referans resmi renk dilimlerine ayırır ve ilk fark haritasını pencere tabanlı eşleme yöntemi ile hesaplar. Bundan sonra her bir dilim için ilk düzlemi uydurmaya çalışır. Bu uydurulan düzlemlere göre resmi katmanlara ayırır ve yeniden düzlem uydurur. Bu son duruma göre dilimlere etiket olarak katmanlar atanır. Atanan katmanlar daha sonra çizge kesme yöntemi ile yeniden değerlendirilir. Eğer daha az enerjili düzlem modeli mevcutsa bu modele göre yeni katman atamaları yapılır ve bu işlem enerji azalmayana kadar devam eder.

Algoritmanın temel adımları ve Hong ve Chen algoritmasından ayrılan yönleri biraz daha ayrıntılı şekilde incelenecektir.

Renk Dilimleme: Resim homojen ve birbiriyle çakışmayan renk dilimlerine ayrılır. Bunun için aynı Hong ve Chen gibi Christoudias ve arkadaşların ortalama-kaydırma tabanlı dilimleme algoritmasını kullanırlar.

İlk Fark Haritası: İlk fark haritasını çıkartmak için yerel maliyeti hesaplamada aynı Hong ve Chen gibi farkların mutlak değerlerinin toplamı (SAD) kullanılmıştır. Yerel yöntemlerin yetersizliği nedeniyle dokusuz bölgelerin daha iyi sonuç vermesi ve kapanmış bölgelerin tespit edilebilmesi için çapraz kontrol yöntemi kullanılmıştır. Test aşamasında Hong ve Chen sadece 3x3 boyutunda pencereler kullanırlarken, Bleyer ve Gelautz 3x3 ve 7x7 boyutunda pencereler kullanmışlardır.

Düzlem Uydurma: Bu aşamada dilim içindeki tüm pikselleri işaretleyebilecek bir düzlem dilime atanır.

$$d(x, y) = a.x + b.y + c$$

Düzlemin dilime atanmasında çok bilinen en küçük kare yöntemi kullanılır. Bu yöntem ile hesaplanan hata $E = \sum_{i=0}^{n-1} (f(x_i, y_i) - z_i)^2 = \sum_{i=0}^{n-1} (ax_i + by_i + c - z_i)^2$ toplamına eşit olacaktır. Bu hatanın en küçük olduğu yer fonksiyonun a,b,c parametrelerine göre alınan kısmi türevlerinin sıfır olduğu yerdir.

Bu kısmi türevler hesaplandığında üç tane vektör ve üç bilinmeyenli üç tane esitlik ortaya çıkar:

$$x = (x_{0}, x_{1}, x_{2}, ..., x_{n-1}), \quad y = (y_{0}, y_{1}, y_{2}, ..., y_{n-1}), \quad z = (z_{0}, z_{1}, z_{2}, ..., z_{n-1})$$

$$ax.x + bx.y + cx.u = x.z$$

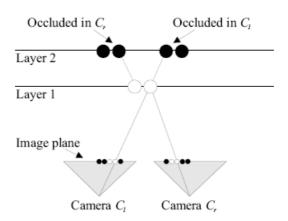
$$ay.x + by.y + cy.u = y.z$$

$$au.x + bu.y + cu.u = u.z$$

Bu eşitliklerden a, b, c parametreleri bulunur. Bulunan değerler d(x,y) = a.x + b.y + c formülünde yerine konulduğunda o dilimi en iyi ifade edebilecek düzlem fonksiyonu bulunmuş olur. Düzlemler bir dilimdeki farkları ifade etmekte yetersiz kalabilirler. Bu nedenle kama uydurma üzerine çalışmalar vardır.

Katman Çıkartma: Bir yüzeye ait düzlemlerin denklemleri birbirlerine çok yakın olacaklardır. Bu nedenle aynı yüzeye ait dilimler gruplanır ve büyük bir grup elde edilir. Bu sayede fazla dilimlemenin neden olacağı sorunlar giderilmiş ve daha düzgün sonuçlar elde edilmiş olur.

Fark Katmanının Atanması: Kapanma durumlarını daha iyi analiz edebilmek için iki seviye belirlenmiştir. Bunlardan ilki "piksel seviyesi" ikicisi ise "dilim seviyesidir". Dilim seviyesi düzgünlük sağlar ama kapanmalar piksellerde olacağından hatalı olabilir. Bu nedenle hem dilimler hem de pikseller katmalara atanmaktadır. Her piksel ya da her dilim sadece tek bir katmana atanabilir.



Şekil 4.12. Piksellerin Fark Katmanlarına Atanması. [Bleyer and Gelautz, 2005.]

Piksel ve dilimlerin katmanlara atanma problemi bir etiketleme problemi olarak düşünülebilir. Bu problem için 0 numaralı katman özel katmandır. Bu katmandaki piksel ya da dilimler kapanmış olarak kabul edilir. Etiket atama problemi için maliyet fonksiyonu şu şekilde yazılabilir:

$$C(f) = min(T_{veri} + T_{uyumsuzluk} + T_{dilimleme} + T_{düzgünlük})$$

Renk Tutarlılığı (T_{veri}): Eğer bir piksel kapanmamışsa onun pencere tabanlı eşleme maliyetini eğer kapanma varsa -1 değerini gösterir.

$$T_{veri} = \sum_{p \in I} \begin{cases} benzemezlik(p, m[f(p)](p)) & : & f(p) \neq 0 \\ -1 & : & diger \end{cases}$$

Görüntü Tutarlılığı ($T_{uyumsuzluk}$): Eğer sol resimdeki p pikseli sağ resimde q pikseline eşlenmişse, sağ resimdeki q pikseli de sol resimdeki p pikseline eşlenmelidir. Bu çapraz kontrol ile saptanır. Bunun olması için teklik kısıdının geçerli olması gerekir.

$$T_{uyumsuzluk} = \sum_{p \in I} \begin{cases} \lambda_{uyumsuzluk} & : \quad f(p) = 0 \lor f(p) \neq f(m[f(p)](p)) \\ 0 & : \quad diger \end{cases}$$

Dilim Tutarlılğı ($T_{\it dilimleme}$): Bu tutarlılık sayesinde hem içsel olarak dilim içindeki farkın düzgünlüğünü zorlamış hem de sağ ve sol resimdeki dilimlerin birbirleriyle tutarlılığını kontrol edilmiş olur. Bu tutarlılık ile bir piksel ya ait bulunduğu dilimin katmanına atanır ya da kapanmıştır.

$$T_{dilimleme} = \sum_{p \in I} \begin{cases} \infty & : & f(p) = 0 \land f(p) \neq f(dilim(p)) \\ 0 & : & diger \end{cases}$$

Düzgünlük ($T_{düzgünlük}$): Nesnelerin sınırları hariç geri kalan yerlerde düzgün fark değerleri elde edileceği düşünülebilir. Bunun için geliştirilmiş formülde hem komşu dilimlerin sınır uzunlukları hem de bu dilimlerin birbiriyle olan renk uyumu göz önüne alınmıştır.

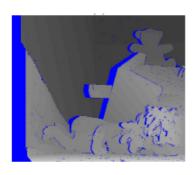
Bu enerjiyi küçültmek için Boykov, Veksler ve Zabih tarafından geliştirilmiş olan α-genişleme algoritması kullanılmıştır. Tüm piksel ve dilimler 0. katmandan başlayarak enerjiyi küçültecek şekilde yeni katman değerlerini alırlar.

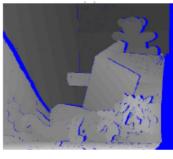






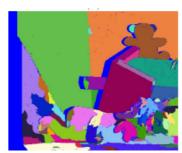
Şekil 4.13. Teddy İçin Orijinal Resimler ve Gerçek Fark Haritası. [Bleyer and Gelautz, 2005.]







Şekil 4.14. Sol ve Sağ Fark Haritaları, Sol Resimdeki Dilimlere Göre Fark Haritası. [Bleyer and Gelautz, 2005.]







Şekil 4.15. Sol ve Sağ Resimler İçin İlk Katman Atamaları, Sol Resimdeki Dilimlere Göre Katman Ataması. [Bleyer and Gelautz, 2005.]





Şekil 4.16. Gerçek Fark Haritası ile Sol Resmin Dilimlerine Göre Bulunan Fark Haritasının Karşılaştırması, Üç Boyutlu Hale Getirilmiş Görüntü. [Bleyer and Gelautz, 2005.]

4.6 Çizge Kesme ve Diğer Algoritmaların

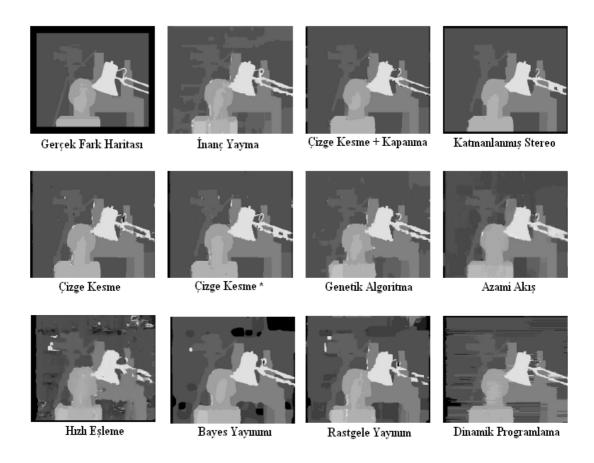
Karşılaştırmaları

Çizge kesme algoritmaları diğer başarılı algoritmalarla karşılaştırıldığında birçok resim üzerinde oldukça iyi sonuçlar vermektedirler. Çizge kesme algoritması içinde kullanılan enerji teriminin kapsamlılığı ve ek kabul/kısıtlar çizge kesme algoritmalarının başarılı üzerinde etkili olmaktadır.

Bu algoritmaların karşılaştırmaları [Scharstein, and Szeliski, 2002] makalelerinde oluşturdukları deney yatağında elde edilen sonuçlardır. Karşılaştırmaların sayısal sonuçları ile birlikte popüler algoritmaların Tsukuba resmi için ürettikleri fark haritaları da verilmiştir.

Çizelge 4.10 Stereo Algoritmalarının Karşılaştırmaları

Algoritma		Tsukuba			Sawtoot	h		Venus		Maj	p
	Tüm	Dokusuz	Fark.	Tüm	Dokusuz	Fark.	Tüm	Dokusuz	Fark.	Dokusuz	Fark.
Sym.BP+occl.	0.97	0.28	5.45	0.19	0.00	2.09	0.16	0.02	2.77	0.16	2.20
Patch-based	0.88	0.19	4.95	0.29	0.00	3.23	0.09	0.02	1.50	0.30	4.08
Segmbased GC	1.23	0.29	6.94	0.30	0.00	3.24	0.08	0.01	1.39	1.49	15.46
Graph+segm.	1.39	0.28	7.17	0.25	0.00	2.56	0.11	0.02	2.04	2.35	20.87
GC + mean shift	1.13	0.48	6.38	1.14	0.06	3.34	0.77	0.70	3.61	0.95	12.83
Segm.+glob.vis.	1.30	0.48	7.50	0.20	0.00	2.30	0.79	0.81	6.37	1.63	16.07
Belief prop.	1.15	0.42	6.31	0.98	0.30	4.83	1.00	0.76	9.13	0.84	5.27
Layered	1.58	1.06	8.82	0.34	0.00	3.35	1.52	2.96	2.62	0.37	5.24
2-pass DP	1.53	0.66	8.25	0.61	0.02	5.25	0.94	0.95	5.72	0.70	9.32
Region-Progress.	1.44	0.55	8.18	0.24	0.00	2.64	0.99	1.37	6.40	1.49	17.11
GC+occl.	1.19	0.23	6.71	0.73	0.11	5.71	1.64	2.75	5.41	0.61	6.05
MultiCam GC	1.85	1.94	6.99	0.62	0.00	6.86	1.21	1.96	5.71	0.31	4.34
GC+occl.	1.27	0.43	6.90	0.36	0.00	3.65	2.79	5.39	2.54	1.79	10.08
Improved Coop.	1.67	0.77	9.67	1.21	0.17	6.90	1.04	1.07	13.68	0.29	3.65
Adapt. weights	1.51	0.65	7.24	1.14	0.27	5.48	1.14	0.61	4.49	1.47	13.58
Graph cuts	1.94	1.09	9.49	1.30	0.06	6.34	1.79	2.61	6.91	0.31	3.88
Reliability-DP	1.36	0.81	7.35	1.09	0.44	4.13	2.35	2.37	13.50	0.55	6.14
Multiw. cut	8.08	6.53	25.33	0.61	0.46	4.60	0.53	0.31	8.06	0.26	3.27
Graph cuts	1.86	1.00	9.35	0.42	0.14	3.76	1.69	2.30	5.40	2.39	9.35
Tree DP	1.77	0.38	9.48	3 1.44	0.84	6.87	1.21	1.41	5.04	1.45	13.00
4-State DP	4.70	3.68	21.05	1.43	0.17	13.93	1.18	0.59	17.98	0.30	4.23
Realtime	4.25	4.47	15.05	1.32	0.35	9.21	1.53	1.80	12.33	0.81	11.35
Cooperative	3.49	3.65	14.77	2.03	2.29	13.41	2.57	3.52	26.38	0.22	2.37
Genetic	2.96	2.66	14.97	2.21	2.76	13.96	2.49	2.89	23.04	1.04	10.91
SSD+MF	5.23	3.80	24.66	2.21	0.72	13.97	3.74	6.82	12.94	0.66	9.35
Dyn. prog.	4.12	4.63	12.34	4.84	3.71	13.26	10.10	15.01	17.12	3.33	14.04

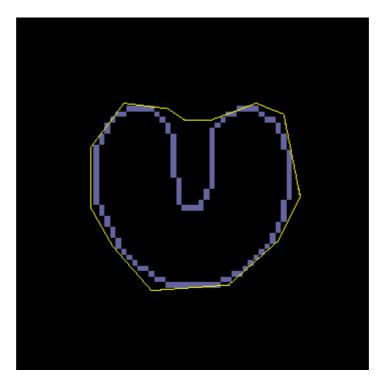


Şekil 4.17 Stereo Algoritmalarının Sonuçları. [Scharstein, and Szeliski, 2002.]

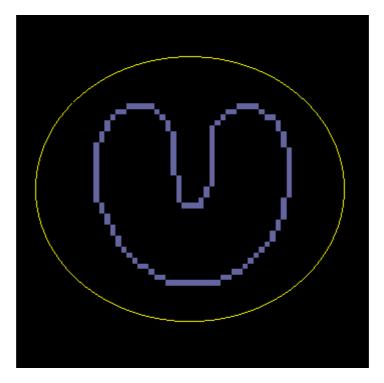
5 ÇİFT AĞ YAPISI

5.1 Aktif Dış Hatlar

Bir aktif dış hat ya da yılan, enerjisini resimdeki durumuna ve kendi şekline göre azaltan bir kama olarak düşünülebilir. Aktif dış hatlar, 1987 yılında Kass ve arkadaşlarının yılanları ilk kez duyurmasıyla bilgisayarla görme alanında yaygın bir şekilde kullanılmaya başlamıştır [Kass et. al., 1987]. Özellikle resim dilimleme konusunda çok hızlı şekilde yaygınlaşmış ve tıbbi bilgisayarla görmede geniş bir uygulama alanı bulmuştur. Bu yöntemin iyi özelliklerinden biri enerji fonksiyonu içindeki süreklilik ve yumuşaklık terimleri sayesinde geometrik sapmaları çok aza indirgemesidir. Bununla birlikte klasik yılan yapısı başlangıç durumuna duyarlıdır ve yerel en iyi noktalarda takılarak deformasyonunu tamamlayamayabilir. Bu gibi sorunlarda yeni geliştirilen aktif dış hat algoritmaları klasik yılan algoritmasına görece üstünlük sağlamıştır fakat bu üstünlük örneklerdeki basit resimler için olsa da her resim için geçerli değildir.



Şekil 5.1. Klasik Yılan.



Şekil 5.2. Yeni Geliştirilmiş Bir Yılan.

$$\begin{aligned} \boldsymbol{E}_{yilan}^* &= \int_0^1 E_{yilan}(\boldsymbol{v}(s)) \, ds \\ &= \int_0^1 \left[E_{ic}(\boldsymbol{v}(s)) + E_{resim}(\boldsymbol{v}(s)) + E_{zorlamalar}(\boldsymbol{v}(s)) \right] ds \end{aligned}$$

Bu genel tanımlamada v(s) = (x(s), y(s), z(s)) şeklinde pikselin parametrik gösterimidir. Bir yılanın enerjisini genel olarak yukarıdaki formülle göstermek mümkündür. Bu enerji terimleri kısaca şu şekilde açıklanabilir:

 $E_{\it ic}$: Kamanın genişleme ve daralmadan dolayı oluşturduğu kendi iç enerjisidir.

 $E_{\it resim}$: Hatlar yâda piksel yoğunlukları gibi resim özelliklerine bağımlılığı belirler.

 $E_{\it zorlamalar}$: Kullanıcı tanımlı diğer enerjiler yâda şekil bilgisi gibi üst seviye bilginin yılanın enerjisine yansıtılmasını sağlar.

Yılanın iç enerjisinde $\alpha(s)$ elastikliği, $\beta(s)$ ise sertliği belirlemek için kullanılır.

$$E_{ic} = \alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{dv^2}{ds^2} \right|^2$$

Resim enerjisinden gelecek etkiler yılanın bırakıldığı resim verisine bağımlı olduğundan genelde değiştirilebilir katsayıya sahip farklı terimler içerirler. Bu terimler resim verisinden bulunabilecek özelliklerin yılan tarafından kullanılabilmesini sağlar.

$$E_{resim} = W_d E_{dogru} + W_k E_{kengr} + W_s E_{sonlanma}$$

Enerji fonksiyonu içindeki terimler sırasıyla resimde bulunan piksellerin açıklık ve koyuluklarına, resimdeki kenarlara ve son olarak da bu kenarların sonlanmasına ve köşelere bakacak terimlerdir.

Genel enerji formülündeki zorlama enerjisi ise kullanıcının üst bilgisini gerektirir ve duruma özeldir.

5.2 Çift Ağ Kullanan Stereo Algoritması

Stereo problemi iki resimdeki piksellerin birbirleriyle eşleşmesi değil aynı anda çekilmiş iki ya da daha fazla fotoğraftan 3-Boyutlu yapının oluşturulmasıdır. Eşleme problemi stereo probleminin içinde oluşmuş ara bir problemdir. Probleme farklı açılardan yaklaşarak bu ara problemden kurtulmak mümkün olabilir. Şekil değiştirebilen aktif dış hatların kullanılmasıyla doğrudan yüzeyin yapısını elde etmek mümkündür [Akgul and Kambhamettu, 1999]. Bunun için sadece enerji fonksiyonun en iyileştirilmesi yeterli olacaktır.

Şekil değiştirebilen ağ yapısı, üç boyutlu uzayda yatay ve dikey olarak birbirlerine bağlı noktalardan oluşur. Ağ elemanlarının x ve y koordinatları resimlerin o konumlardaki piksellerini temsil ederler. Bir ağ elemanı hiçbir zaman bu koordinatlar üzerinde hareket etmez. Ağın z koordinatı ise derinlik bilgisini g. Bu nedenle ağ elemanı uygun derinlik konumuna ulaşıp, küresel enerjiyi asgari değere çekmek için bu koordinat üzerinde hareket eder.

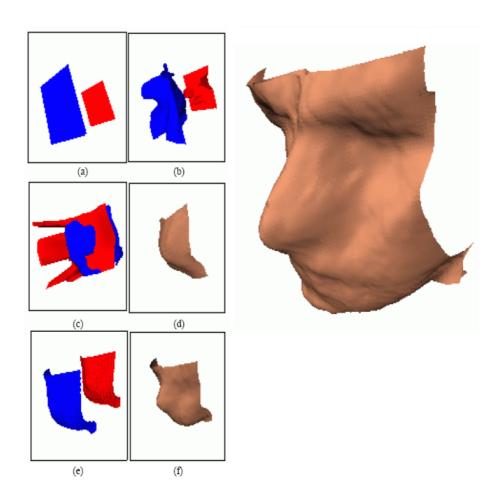
Algoritma temel olarak iki tane şekil değiştirebilen ağın farklı başlangıç durumlarında başlayarak aynı şekli almasını ve bu son şeklin yüzeyin şekli olduğunu kabul eder. Her bir ağ kendi enerjisini diğer ağ ile etkileşerek düşürmeye çalışırlar. Izgaralar birbirlerini çekecekler ve son durumda tüm pikseller birleşmiş olacaktır.

$$E_{\text{Izgara}}(M, N) = \sum_{i=1}^{k} \sum_{i=1}^{l} \alpha E_{\text{Düzgünlük}}(m_{ij}) + \beta E_{\text{Etkilesim}}(m_{ij}, n_{ij}) + \gamma E_{\text{Veri}}(m_{ij})$$

M ve N iki tane şekil alabilen ağ olsunlar. Düzgünlük enerji fonksiyonu ızgara içindeki pikseller için düzgünlüğe bakacaktır. Bu terime iç enerji terimi diyebiliriz. Burada yapılan bir pikselin derinliğinin komşu piksellerin derinliklerine benzetmeye çalışmaktır. Bu yöntem sayesinde geçişler daha düzgün olacaktır.

$$E_{\textit{Dizgiinlik}}\left(m_{ij}\right) = \left(1 - \frac{\left(\vec{m}_{i-1j}m_{ij}\right)\left(\vec{m}_{ij}m_{i+1j}\right)}{\left|\vec{m}_{i-1j}m_{ij}\right|\left|\vec{m}_{ij}m_{i+1j}\right|}\right) + \left(1 - \frac{\left(\vec{m}_{ij-1}m_{ij}\right)\left(\vec{m}_{ij}m_{ij+1}\right)}{\left|\vec{m}_{ij-1}m_{ij}\right|\left|\vec{m}_{ij}m_{ij+1}\right|}\right)$$

Veri enerjisi ya da bir başka deyişle dış enerji ise ağ elemanlarının piksellerin renk yoğunluklarına göre hareket etmesini sağlar. Bu terimin hesaplanması için normalleştirilmiş çapraz ilişki yöntemi kullanılmıştır.



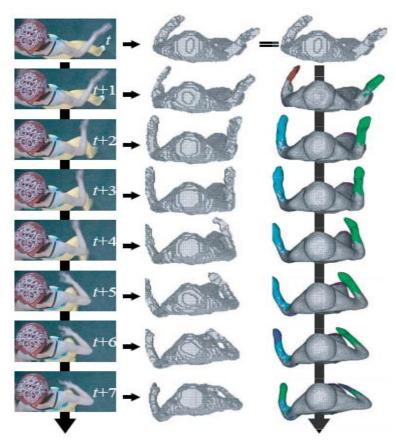
Şekil 5.3. İki Tane Şekil Alabilen Ağ Kullanarak Yüzeyin Elde Edilmesi. [Akgul and Kambhamettu, 1999].

Etkileşme enerjisi ise aktif dış hatların zayıf yanı olan yerel en iyi noktaya takılmayı engellemek için kullanılmaktadır. Bu terim ilk başta kullanılmaz. Eğer hırslı algoritmamız yerel en iyide takılmışsa kullanılmaya başlar. Bunu bir önceki bölümdeki zorlama enerjisine benzetmek de mümkündür. Bu terimin çalıştırılıp çalıştırılmayacağına karar vermek için ağların şekil değiştirmelerinin durduğu durumda birbirleriyle enerjileri karşılaştırılır. Eğer her iki ağ da aynı enerjiye ve konuma sahipseler yüzey bulunmuştur. Farklı enerjilere sahiplerse daha az enerjiye sahip olanın küresel en iyi sonuca daha yakın olduğu düşünülür ve diğer ızgara o tarafa doğru itilir.

$$E_{Etkilestk}(m_{ij}, n_{ij}) = |derinlik(m_{ij}) - derinlik(n_{ij})|$$

Ağ bir kez itildikten sonra şekil değiştirmeye yine iç ve dış enerjinin etkisiyle devam eder. Algoritma sadece ağlar aynı konuma geldiğinde sonlanır.

Algoritmanın önemli avantajlarından biri de hareket takibidir. Hareketlerin birbirini takip eder şekilde olacağını düşünürsek ikinci durum ağların ilk buldukları duruma oldukça yakın olacaktır. Bu nedenle ağlar en son ve en baştan başlatılmak yerine ilk bulunan sonucun belli bir civarından başlatılabilir ve algoritmanın daha kısa çalışma zamanında sonlanması sağlanır [Nobuhara and Matsuyama, 2004].



Şekil 5.4 Nobuhara ve Arkadaşlarının Hareket Takip Uygulaması. [Nobuhara and Matsuyama, 2004].

Bununla birlikte algoritmanın çeşitli yönlerinin iyileştirilmesi gerekmektedir. Bunlardan ilki algoritmanın katsayıları bağımlı olması ve bu katsayıların her resim çifti için farklı olmasıdır. Bu parametreler için belli bir kural bulunamamıştır. Her seferinde denenerek en iyi parametreler bulunur. İkinci olarak iyileştirilmesi gereken kısmı ise düzgünlüğün yeterince iyi sonuçlar vermemesidir. Bu yöntem temel olarak düzgünlüğün kendi içinde zorlandığı yılan yapısını temel alması nedeniyle sadece sürekli alanlarda başarılı sonuçlar vermektedir.

6 ÇİZGE KESME TABANLI ÇİFT AĞ YAKLAŞIMI

6.1 Çift Ağ ve Çizge Kesmenin Birlikte

Kullanılmasının Nedenleri

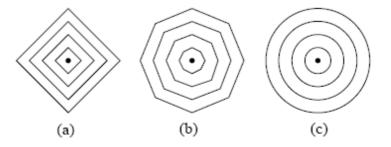
Hem çizge kesme yönteminin hem de çift ağ yönteminin bazı sıkıntıları vardır. Çizge kesme algoritmasının sıkıntıları; başlangıç konfigürasyonuna ve ara çizgelerin oluşturulma sıralarının sonucu değiştirmesi ve çizge yapısının doğasından kaynaklanan ayrıklıktır. Çift ağ yaklaşımının belli başlı sorunlarının ise çok fazla sayıda parametrelerin ayarlanması, çalıştırdığı enerji fonksiyonuna ve optimizasyon yöntemine bağımlılığı ve sadece sürekli yüzeylerde çalışması olduğu söylenebilir.

Çift ağ yapısı içinde çizge kesme ile optimizasyon yapmak oldukça verimli olacaktır. Öncelikle çizge kesme algoritmaları en iyi sonucun belli bir yaklaşıklığında çözüm sunabilmekte ve genel test resimleri üzerinde oldukça başarılı sonuçlar elde etmektedir. Çizge kesme algoritmaları içlerinde kullandıkları Potts stili konveks olmayan düzgünlük terimleri sayesinde sürekli yüzeylerin dışında da çalışabilmektedir. Çift ağ yapısının son önemli sorunu olan parametrelerin ayarlanma sorunu da yine çizge kesme algoritmasının ve istatistiksel yöntemlerin birlikte kullanılmasıyla rahatlıkla aşılabilmektedir. Kapanma durumlarının incelendiği stereo makalesinde çizge kesme algoritmasının parametre bağımlılığı da gözlemlenmiş ve algoritmanın bu parametrenin marjinal değerleri için bile bulduğu sonuçtan fazla sapmadığı görülmüştür [Kolmogorov and Zabih, 2001].

Çizge kesme yönteminin optimum sonuçtan sapması sınırlı olsa da bir birleriyle etkileşim içinde kullanılacak iki adet optimizasyon sürecinin bu sınırda yada ondan daha iyi bir sonuç vermesi sağlanabilir. Bu işlemin tek bir çizge kesme optimizasyonuna göre sisteme ek hesaplama maliyeti getireceği açıktır ama 3-Boyutlu takip problemi gibi durumlarda çift ağ kullanılması hesaplamayı

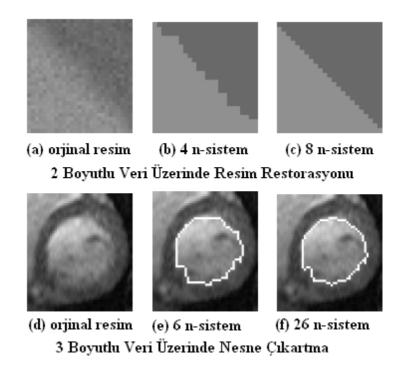
kolaylaştıracaktır. Çizge kesme probleminin ikinci sorunu ve uzunca bir süre bilgisayarla görme alanında kullanılmamasının nedeni ise çizgelerin ayrık yapılarıdır. Bilgisayarla görme alanındaki problemler ise sürekli yapıdadır. Bu nedenle çizgelerin hesaplamaya ek hatalar getireceği düşünülmüştür. Bu geometrik sapmanın ne kadar olabileceği üzerine Boykov ve arkadaşlarının çalışmaları vardır [Boykov and Kolmogorov, 2003]. Makale, hem bu sorunun çözümü hakkında ipuçları vermesi hem de aktif dış hatlarla, çizge kesmenin dilimleme ve nesne çıkartma problemlerinde kullanımını göstermesi nedeniyle daha ayrıntılı olarak incelenecektir.

Kesikli yapının oluşturacağı sonuçları Dijkastra'nın en kısa yol algoritmasını kullanarak irdelemişlerdir. Bir çizge üzerindeki her hangi iki noktanın birbirlerine olan uzaklıkları yol ölçüsünü verir. Bunun için aşağıdaki şekilde Manhattan tarzı yerleşim olan (a) şekli ile Öklid ölçütlerine daha çok yaklaşan 8 ve 128 komşuluklu yapılar gösterilmiştir.



Şekil 6.1. Manhattan Tarzı 4-8-128 Komşuluklu Sistemler. [Boykov and Kolmogorov, 2003.]

Farklı sayıda komşulukların algoritmada kullanılmasıyla bu sonuçlar pratikte doğrulanmış ve komşu sayısının artırılmasının çizge kesmenin geometrik hata oranını düşürdüğü görülmüştür. Komşu sayısının arttırılmasının dezavantajı ise çizgenin karmaşıklaşması ve hesaplama maliyetinin artmasıdır.



Şekil 6.2. Komşu Sayısının Sonuca Etkileri. [Boykov and Kolmogorov, 2003.]

6.2 Stereo İçin Çizge Kesme Tabanlı Çift Ağ Algoritması

6.2.1 Algoritmanın Açıklanması

Sistem çizge kesme optimizasyonu ile şekil değiştirebilen çift ağ yapısının bütünleşik olarak kullanılması üzerine kuruludur. Yeni oluşturulan sistem birbirleriyle etkileşim içinde çalışan iki adet çizge kesme optimizasyon sürecini içermektedir. Resim verisine, farkların komşu pikseller arasında düzgün geçiş yapmasına ve şekil değiştirebilen ağların konumlarına bağlı olan bir küresel enerji fonksiyonu çizge kesme algoritması kullanılarak sistem içinde asgarileştirilmeye

çalışılır. Çift ağ yapısı ve çizge kesme algoritmasının birlikte kullanılması sistemin daha kararlı ve verimli bir şekilde çalışmasını sağlamıştır.

Bir şekil değiştirebilen ağ yapısı, üç boyutlu uzayda yatay ve dikey olarak birbirlerine bağlı noktalar olarak tanımlanabilir. Ağ elemanlarının x ve y koordinatları resimlerin o konumlardaki piksellerini temsil ederler. Resimlerin k, l boyutunda oldukları düşünülürse her bir ağın boyutları da k,l olacaktır. Ağın z koordinatı ise derinlik bilgisini gösterir. Bu nedenle ağ elemanı uygun derinlik konumuna ulaşıp, küresel enerjiyi asgari değere çekmek için bu koordinat üzerinde hareket eder.

Ağların z koordinatı üzerinde aynı konumlara gelmesi ve aynı sonucu göstermeleri beklenir. Optimizasyon süreci içinde bir ağ yerel en iyi noktaya takılırsa ağlar aynı konuma gelmeden dururlar. Bu durumlar için klasik veri ve düzgünlük durumlarına bakan enerji fonksiyonuna ağların etkileşimi sağlayan bir enerji terimi daha eklenmiştir. Bu etkileşim terimi birbirlerine kavuşmamış yada birbirlerini geçmiş ağ elemanlarının aralarında bir enerji oluşturur. Bu enerjiye göre elemanlar aynı konuma gelebilmek için birbirlerini iterler yada çekerler. Bu şekilde tanımlanmış bir sistemde tek bir ağın toplam enerjisi şu şekilde formülleştirilebilir:

$$E_{Ag}(M, N) = \sum_{i=1}^{k} \sum_{j=1}^{l} E_{Veri}(m_{ij}) + E_{Düzgünlük}(m_{ij}) + E_{Etkilesim}(m_{ij}, n_{ij})$$

Veri enerjisi sağ ve sol resimde eşleşecek piksellerin aynı renk yoğunluklarına sahip olup olmadığını kontrol eder. Enerji terimini hesaplarken tek tek pikselleri karşılaştırmak yerine piksellerin belirli bir komşuluk alanı aranıp eşlenmeye çalışılır. Bu yöntemle elde edilen sonuçlar tek bir pikselin vereceği sonuçtan çok daha güvenilir olur. Veri enerjisi daha önce anlatılan maliyet hesaplama yöntemlerinden yoğunluk farklarının karesi kullanılarak hesaplanabilir. Eğer R ve L sağ ve sol resimleri gösterir ve benzerlik penceresinin boyutu s x s olarak alınırsa her bir ağ elemanı için veri enerjisi aşağıdaki şekilde bulunur.

$$E_{Veri}(m) = \frac{1}{s^2} \left(\sum_{i=-s/2}^{s/2} \sum_{j=-s/2}^{s/2} (L_{ij} - R_{ij})^2 \right)$$

Düzgünlük terimi kullanılarak bir pikselin derinliğinin diğer komşu piksellerin derinliklerine bağımlı olması sağlanır. Bu sayede sistem gürültüye karşı daha kararlı olur ve elde edilen fark haritaları nesne kenarları hariç daha doğru sonuçlar verir. Nesne kenarlarındaki ani atlamalar düzgünlük terimi tarafından yumuşatılabilir. Bunu engellemek için konveks olmayan düzgünlük terimleri kullanılır. Bu terimlerin hesaplanmaları oldukça maliyetlidir. Potts tarzı düzgünlük terimi ise oldukça basit hesaplanmakla birlikte çok verimlidir. Ani fark değişimlerini korur ve hesaplama karmaşıklığı düşüktür. Bu nedenle algoritmada düzgünlük terimi olarak 4-komşuluklu Potts tarzı düzgünlük terimi kullanılmıştır.

$$E_{Duzgunluk}(m_{ij}) = V(m_{ij}, m_{i+1j}) + V(m_{ij} + m_{ij+1})$$

 m_{ii} ve m_{kl} iki komşu ağ elemanı olduğunda;

$$V(m_{ij}, m_{kl}) = \begin{cases} 0 & derinlik(m_{ij}) = derinlik(m_{kl}), \\ \lambda_1 & \left| derinlik(m_{ij}) - derinlik(m_{kl}) \right| \le esik_1, \text{Bu} & \text{formulle,} \\ \lambda_2 & diger. \end{cases}$$

derinliğe sahip iki komşu ağ elemanının birbirlerine düzgünlük cezası vermesi engellenmiş olur. Eğer iki ağ elemanının derinlik farkları belli bir eşik değerinin altındaysa daha düşük, bu eşiğin üstündeyse daha yüksek bir düzgünlük cezası alır.

$$E_{\textit{Etkilesim}}(m_{ij}, n_{ij}) = \begin{cases} 0 & derinlik(m_{ij}) = derinlik(n_{ij}), \\ \infty & derinlik(m_{ij}) > derinlik(n_{ij}), \\ \lambda_{3}*(derinlik(n_{ij}) - derinlik(m_{ij})) & (derinlik(n_{ij}) - derinlik(m_{ij})) < esik_{2}, \\ \lambda_{3}*esik_{2} & diger. \end{cases}$$

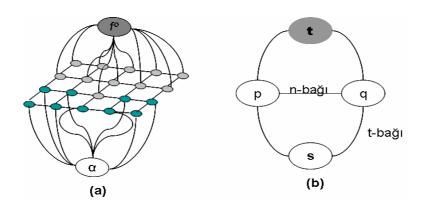
Etkileşim enerjisi her zaman aktif değildir. Bu enerji sadece iki ağ birbirlerine kavuşmadan durdurduklarına etkili hale gelir. İki ağ ilk konfigürasyon olarak olası

fark değer aralığının farklı uçlarından başlarlar. Ağlardan bir tanesi sürekli ve sıralı şekilde asgari fark değerinden azami fark değerine doğru giderken, diğer ağ ise azami fark değerinden asgari fark değerine doğru ilerler. Ağ elemanlarının birbirlerinin içinden geçmeleri bu elemanların iki farklı optimizasyon sürecinde farklı yerlere gitmesine ve farklı konumlarda kalmasına neden olacağından birbirlerini geçmelerini zorlaştırmak için enerji teriminde ceza yüksek tutulmuştur. Bu geçiş tamamen yok edilmemiş ve ağların bir miktar geriye doğru gitmesine izil verilmiştir. İki ağ elemanı birbirlerinden çok uzakta kalmışlarsa bu elemanları çok kuvvetli bir şekilde çekip optimizasyon sürecinin erken tamamlanmasını engellemek için eşik değeri konulmuştur. Bu değer kavuşmama cezasının fazlaca artmasını engeller. Eğer M ağı N ağından daha düşük fark değerleri alacak şekilde yerleştirilmişse, etkileşim enerjisi aşağıdaki şekilde formülleştirilebilir. Formül içindeki sonsuz değeri diğer maliyetlere göre çok yüksek bir maliyettir ama birden çok çizge düğümü veri ve düzgünlük enerjisini azaltacak şekilde diğer ağı geçerse enerjisi yüksek kalan ağ geri doğru esner.

Çizge kesme algoritmaları genel enerji formülünü asgariye indirmek için en iyileştirmeye çalışırken her bir olası etiket / fark değeri için çizgeyi oluşturmalıdır. Bu çizgenin oluşumu algoritmadan algoritmaya değişir. Buradaki algoritmada veri enerjileri t-bağlarına, düzgünlük enerjileri ise n-bağlarına yerleştirilmiştir. Etkileşim enerjisi etkin hale geldiğinde t-bağlarına üzerine maliyet eklemekte ve ağ düğümlerini birbirlerine doğru harekete zorlamaktadır. Bu ara çizgeler tüm olası etiketler için oluşturulur. Tüm etiketler için bir kez sorulduğunda bir iterasyon tamamlanmış olur. Bu tekrarlar ağların enerjisi azalmayana kadar devam eder. Ağlar durduklarında birbirleriyle aynı konumlarda değilseler bu sefer etkileşim enerjisi devreye girer ve ağların birbirlerine doğru hareket etmesi sağlanır.

Çizge kesme tabanlı optimizasyon için öncelikle özel bir çizgenin oluşturulması gerekir. α -genişleme algoritması için çizge, kaynak düğümün ilk konfigürasyonu, hedef düğümün α etiket değerini ve diğer her bir düğümün de ilgili pikseli gösterdiği bir çizge her α etiket değeri için oluşturulacaktır. Bu çizge içerisinde veri enerjileri t-bağlarına, düzgünlük enerjileri ise n-bağlarına yerleştirilecektir. Etkileşim enerjisi aktifleştiğinden ağları birbirlerine doğru çekmek

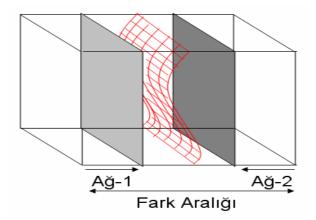
için t-bağları üzerine yerleştirilecektir. Etkileşim sürecinde düzgünlük enerjisi ağların konumlarını korumaya zorlayacaktır ve ağların düzgünlüğü bozarak diğer ağa doğru gitmesine karşı koyacaktır. Bu şekilde oluşturulan çizgeler s-t kesme sonucu sadece tek bir etiket değerine sahip olacak şekilde kaynak yada hedef noktadan ayrılacaklardır. Bu tekrarlar herhangi bir enerji azalması olmayan kadar yada başka bir deyişle ağ elemanları tamamen birbirleriyle aynı konumları aldıklarında sonlanacaktır.



Şekil 6.3. Kesme ve Çizgedeki Bağlar.

α-genişleme algoritması oldukça iyi bir sezgisel yöntem olmasına ve iyi bir hata üst sınırı vermesine rağmen çift ağ yapısı ile birlikte kullanıldığında daha başarılı sonuçlar verecektir. Çizge kesme yöntemini çift ağ ile birlikte kullanan algoritma şu adımlarla tarif edilebilir:

✓ M ağı en düşük fark etiketinden, N ağı en yüksek fark etiketinden başlayacak şekilde konumlanır. İki ağ toplam enerjiyi en aza indirecek şekilde çizge kesme ile optimize edilir. Bu aşamada etkileşim enerjisi devreye girmez.



Şekil 6.4. Ağların Hareketi.

- ✓ Her iki optimizasyon süreci sonlandığında ağ elemanlarının konumları kontrol edilir. M ve N ağlarının aynı konumda olmayan elemanları için etkileşim enerjisi devreye girer. Bu enerji t-bağlarının üzerine yerleştirilir. Aynı konumdaki elemanlar için etkileşim enerjisi sıfırdır.
- ✓ Etkileşim enerjisi, ağları aynı konumları almaya yönlendirir. Bununla birlikte veri ve düzgünlük enerjisi terimleri nedeniyle ağlar birbirlerine yapışmayabilirler.
- ✓ Ağlar aynı konumları aldıklarında optimizasyon tamamlanır. Eğer ağlar aynı konumlara kavuşmamışlarsa önce etkileşim enerjisi kapatılır ve optimizasyon süreci her iki ağ için tekrarlanır. Eğer ağlar yeni ve birbirlerine daha yakın olan bu başlangıç değerlerinden hareket ederek birbirlerine tamamen kavuşmamışlarsa, etkileşim enerjisi yeniden devreye alınır. Bu işlemler ağlar tamamen kavuşana yada optimizasyonda iyileşme durana kadar aynı şekilde tekrarlanır.
- ✓ Bu işlemler tamamlandıktan sonra düşük enerjili ağın z koordinatındaki değerleri o konumun gösterdiği piksel için fark değeri olarak atanır.

6.2.2 Deney Sonuçları

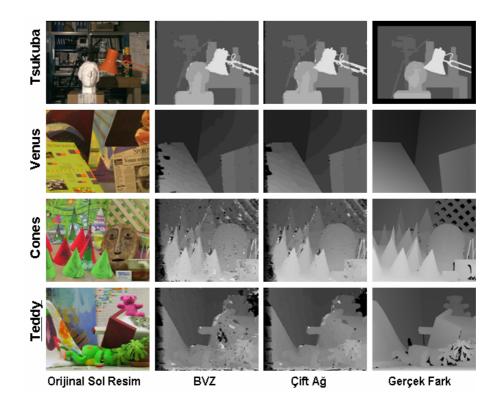
Algoritmanın denemeleri Middlebury resim veritabanındaki resimlerle yapılmıştır[Scharstein and Szeliski, 2003][Scharstein et. al., 2001]. Algoritmanın ürettiği fark haritaları hem diğer algoritmaların sonuçlarıyla hem de gerçek fark haritalarıyla karşılaştırılmıştır. Algoritma C++ dilinde, Kolmogorov ve Zabih tarafından yazılmış kütüphane kullanılarak gerçeklenmiştir. Kütüphane Boykov ve Zabih'in geliştirmiş olduğu azami akış/asgari kesme algoritmasının yanında diğer popüler çizge kesme algoritmalarının da kodlarını içermektedir [Boykov et. al. 1999][Kolmogorov and Zabih, 2001][Kolmogorov and Zabih, 2002][Boykov et. al., 1998][Boykov and Kolmogorov, 2002].

Üç temel deney yapılmıştır. Bu deneylerden ilkinde algoritmanın ürettiği fark haritaları diğer popüler çizge kesme algoritmaları ile karşılaştırılmıştır. Bu algoritmalar BVZ [Boykov et. al., 1998], KZ1[Kolmogorov and Zabih, 2002] ve KZ2 [Kolmogorov and Zabih, 2001] algoritmalarıdır. BVZ algoritması diğer iki algoritmaya göre oldukça basit bir enerji fonksiyonu içerir ve sadece düzgünlük ile veri enerjisi durumlarına bakar. Bu basit yapısı ve kolay uyarlanabilmesi nedeniyle çift ağ kullanan çizge kesme algoritmasının temel algoritması olarak BVZ kullanılmıştır. Bununla birlikte diğer algoritmalar vardır. Deney sonuçları, çift ağ kullanan yeni çizge kesme algoritmasının ikiden çok kameranın elde ettiği görüntülerden yaralanabilme özelliği olan KZ1 ve kapanmaları algılayabilen KZ2 algoritmalarının sonuçlarına yaklaştığı göstermiştir. Yeni algoritmayı bu algoritmalarla doğrudan yapılacak bir karşılaştırma insaflı olmayacaktır, çünkü çift ağ yapısı taban algoritmaya bağımlıdır. Sonuçlarda yeni algoritma her zaman BVZ algoritmasından çok daha iyi sonuçlar üretmiştir. İlerideki uygulamalarda taban algoritma olarak daha gelişmiş algoritmalar kullanılacaktır. Bu konudaki beklenti taban algoritmaya göre çok daha başarılı sonuçların elde edilmesidir.

Karşılaştırma çizelgesinde K kapanma olmayan bölgeleri, T tüm pikselleri ve S süreksizliklerin olduğu alanları gösterir.

Çizelge 6.1. Çift Ağ Algoritmasının Karşılaştırılması.

	7	Sukub	a	Venüs			Teddy			Cones		
Algoritma	K	T	S	K	T	S	K	T	S	K	T	S
BVZ	1,96	4,20	9,71	2,03	3,69	12,1	17,3	25,8	28,8	19,2	28,3	25,7
KZ1	1,83	2,48	6,42	1,06	1,52	5,53	12,0	17,9	22,4	5,78	12,9	13,2
KZ2	1,33	2,15	6,92	1,22	1,78	5,99	12,5	18,8	22,1	6,08	13,2	13,3
Çift Ağ	1,91	4,13	9,50	1,64	3,29	10,5	13,0	21,9	25,2	9,37	19,6	17,5

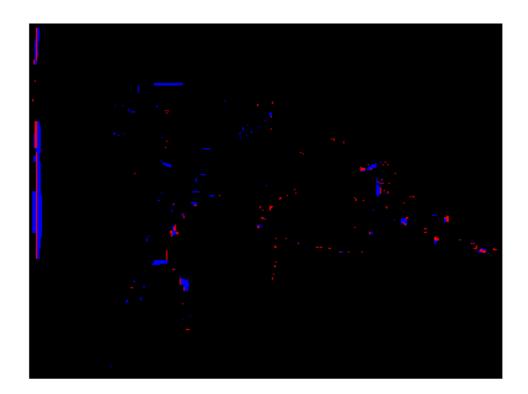


Şekil 6.5. Çift Ağ Algoritmasının Karşılaştırılması.

İkinci deneyde, resimlerdeki dokusuz yâda kapanmış bölgelerin sonuçları nasıl etkiledikleri araştırılmıştır. Bir iterasyon sonunda ağların kavuşmadığı noktalar problemli alanlar olarak işaretlenmiştir. Tsukuba resminde, problemli alanların nerede ise yarısını kapanmış bölgeler oluşturmaktadır. Kavuşmayan diğer noktaların ise nesnelerin kenarları gibi süreksiz bölgeler yâda dokusuz alanlar olduğu düşünülmektedir. Sonuçlar hem gerçek kapanma haritalarından hem de görsel olarak yapılan karşılaştırmalardan elde edilmiştir.

Çizelge 6.2. Problemli ve Kapanmış Piksellerin Benzerliği

	Piksel Sayısı	Yüzde Oran
Toplam Piksel	109921	100,0
Problemli Ağ Elemanları	1312	1,19
Problemli ve Kapanmış Pikseller	504	0,45



Şekil 6.6 Problemli Pikseller (Tsukuba).

Mavi renkle gösterilen yerler ağların bir iterasyon sonunda kavuşmadığı yerleri gösterirken, kırmızı renkle işaretlenmiş yerler ağların birbirlerini geçtikleri yerlerdir. Tüm bu noktalar nesnelerin kenarlarında yoğunlaşmışlardır.

Son olarak ise BVZ algoritması ile BVZ tabanlı çift ağ algoritmasının Cones resmi üzerinde karşılaştırılması yapılmıştır. BVZ algoritması başlangıç koşuluna ve etiketlerin değişim sırasına bağlı olduğundan 10 kez çalıştırılmıştır. Elde edilen sonuçlardan en iyi, en kötü ve ortalama değerler elde edilmiştir. Bunun dışında BVZ algoritması yeni algoritmadaki ağlar gibi iki farklı uçtan ve sıralı şekilde hareket

etmeye zorlanmıştır. Bu iki süreç birbirinden habersiz çalışan iki ağ olarak düşünülebilir. Algoritma etkileşim içinde çalıştığında tek tek çalışan iki optimizasyon sürecinin en iyisinden çok daha iyi sonuç vermektedir.

Çizelge 6.3. Çift Ağın Taban Optimizasyona Etkisi.

Algoritma	Kapanmamış Alanlar	Tüm Alanlar	Süreksiz Alanlar
BVZ Rasgele (En İyi)	18,9	28,0	25,5
BVZ Rasgele (Ortalama)	19,3	28,4	25,7
BVZ Rasgele (En Kötü)	19,6	28,6	26,1
BVZ Düzenli-1 (Tek Ağ)	19,0	28,1	25,4
BVZ Düzenli-2 (Tek Ağ)	18,9	28,0	25,5
Çift Ağ	9,37	19,6	17,5

6.3 Stereo İçin Güvenilirlik Tabanlı Çizge Kesme Yöntemi

6.3.1 Algoritmanın Açıklanması

Sistem çizge kesme optimizasyonu ile şekil değiştirebilen çift ağ yapısının bütünleşik olarak kullanılması üzerine kuruludur. Yeni oluşturulan sistem birbirleriyle etkileşim içinde çalışan iki adet çizge kesme optimizasyon sürecini içermektedir. Resim verisine, farkların komşu pikseller arasında düzgün geçiş yapmasına ve şekil değiştirebilen ağların konumlarına bağlı olan bir küresel enerji fonksiyonu çizge kesme algoritması kullanılarak sistem içinde asgarileştirilmeye çalışılır. Çift ağ yapısı ve çizge kesme algoritmasının birlikte kullanılması sistemin daha kararlı ve verimli bir şekilde çalışmasını sağlamıştır. Çift ağın sağladığı güvenilirlik analizi kullanılarak resimlerdeki dokusuz yada kapanmamış alanların güvenilir bölgelerden yardım alarak farklarının daha doğru tahmin edilmesi sağlanmıştır.

Bir şekil değiştirebilen ağ yapısı, üç boyutlu uzayda yatay ve dikey olarak birbirlerine bağlı noktalar olarak tanımlanabilir. Ağ elemanlarının x ve y

koordinatları resimlerin o konumlardaki piksellerini temsil ederler. Resimlerin k, l boyutunda oldukları düşünülürse her bir ağın boyutları da k,l olacaktır. Ağın z koordinatı ise derinlik bilgisini gösterir. Bu nedenle ağ elemanı uygun derinlik konumuna ulaşıp, küresel enerjiyi asgari değere çekmek için bu koordinat üzerinde hareket eder.

Ağların z koordinatı üzerinde aynı konumlara gelmesi ve aynı sonucu göstermeleri beklenir. Optimizasyon süreci içinde bir ağ yerel en iyi noktaya takılırsa ağlar aynı konuma gelmeden dururlar. Bu durumlar için klasik veri ve düzgünlük durumlarına bakan enerji fonksiyonuna ağların etkileşimi sağlayan bir enerji terimi daha eklenmiştir. Bu etkileşim terimi birbirlerine kavuşmamış yada birbirlerini geçmiş ağ elemanlarının aralarında bir enerji oluşturur. Bu enerjiye göre elemanlar aynı konuma gelebilmek için birbirlerini iterler yada çekerler. Bu şekilde tanımlanmış bir sistemde tek bir ağın toplam enerjisi şu şekilde formülleştirilebilir:

$$E_{Ag}(M, N) = \sum_{i=1}^{k} \sum_{j=1}^{l} E_{Veri}(m_{ij}) + E_{Düzgünlük}(m_{ij}) + E_{Etkilesim}(m_{ij}, n_{ij})$$

Veri enerjisi sağ ve sol resimde eşleşecek piksellerin aynı renk yoğunluklarına sahip olup olmadığını kontrol eder. Enerji terimini hesaplarken tek tek pikselleri karşılaştırmak yerine piksellerin belirli bir komşuluk alanı aranıp eşlenmeye çalışılır. Bu yöntemle elde edilen sonuçlar tek bir pikselin vereceği sonuçtan çok daha güvenilir olur. Veri enerjisi daha önce anlatılan maliyet hesaplama yöntemlerinden yoğunluk farklarının karesi kullanılarak hesaplanabilir. Eğer R ve L sağ ve sol resimleri gösterir ve benzerlik penceresinin boyutu n x n olarak alınırsa her bir ağ elemanı için veri enerjisi aşağıdaki şekilde bulunur.

$$E_{Veri}(m) = \frac{1}{n^2} \left(\sum_{i=1}^n \sum_{j=1}^n (L_{ij} - R_{ij})^2 \right)$$

Düzgünlük terimi kullanılarak bir pikselin derinliğinin diğer komşu piksellerin derinliklerine bağımlı olması sağlanır. Bu sayede sistem gürültüye karşı daha kararlı olur ve elde edilen fark haritaları nesne kenarları hariç daha doğru sonuçlar verir.

Nesne kenarlarındaki ani atlamalar düzgünlük terimi tarafından yumuşatılabilir. Bunu engellemek için konveks olmayan düzgünlük terimleri kullanılır. Bu terimlerin hesaplanmaları oldukça maliyetlidir. Potts tarzı düzgünlük terimi ise oldukça basit hesaplanmakla birlikte çok verimlidir. Ani fark değişimlerini korur ve hesaplama karmaşıklığı düşüktür. Bu nedenle algoritmada düzgünlük terimi olarak 4-komşuluklu Potts tarzı düzgünlük terimi kullanılmıştır.

$$E_{Duzgunluk}(m_{ij}) = V(m_{ij}, m_{i+1j}) + V(m_{ij} + m_{ij+1})$$

 m_{ij} ve m_{kl} iki komşu ağ elemanı olduğunda;

$$V(m_{ij}, m_{kl}) = \begin{cases} 0 & derinlik(m_{ij}) = derinlik(m_{kl}), \\ \lambda_1 & \left| derinlik(m_{ij}) - derinlik(m_{kl}) \right| \le esik_1, \\ \lambda_2 & diger. \end{cases}$$

Bu formülle, aynı derinliğe sahip iki komşu ağ elemanının birbirlerine düzgünlük cezası vermesi engellenmiş olur. Eğer iki ağ elemanının derinlik farkları belli bir eşik değerinin altındaysa daha düşük, bu eşiğin üstündeyse daha yüksek bir düzgünlük cezası alır.

$$E_{Etkilesim}(m_{ij}, n_{ij}) = \begin{cases} 0 & derinlik(m_{ij}) = derinlik(n_{ij}), \\ \lambda_3 & diger. \end{cases}$$

Etkileşim enerjisi her zaman aktif değildir. Bu enerji sadece iki ağ birbirlerine kavuşmadan durdurduklarına etkili hale gelir. İki ağ ilk konfigürasyon olarak olası fark değer aralığının farklı uçlarından başlarlar. Ağlardan bir tanesi sürekli ve sıralı şekilde asgari fark değerinden azami fark değerine doğru giderken, diğer ağ ise azami fark değerinden asgari fark değerine doğru ilerler. Ağ elemanlarının birbirlerinin içinden geçmeleri bu elemanların iki farklı optimizasyon sürecinde farklı yerlere gitmesine ve farklı konumlarda kalmasına neden olacağından

birbirlerini geçmelerini zorlaştırmak için enerji teriminde ceza yüksek tutulmuştur. Bu geçiş tamamen yok edilmemiş ve ağların bir miktar geriye doğru gitmesine izin verilmiştir. İki ağ elemanı birbirlerinden çok uzakta kalmışlarsa bu elemanları çok kuvvetli bir şekilde çekip optimizasyon sürecinin erken tamamlanmasını engellemek için eşik değeri konulmuştur. Bu değer kavuşmama cezasının fazlaca artmasını engeller. Eğer M ağı N ağından daha düşük fark değerleri alacak şekilde yerleştirilmişse, etkileşim enerjisi asağıdaki sekilde formülleştirilebilir. Formül içindeki sonsuz değeri diğer maliyetlere göre çok yüksek bir maliyettir ama birden çok çizge düğümü veri ve düzgünlük enerjisini azaltacak şekilde diğer ağı geçerse enerjisi yüksek kalan ağ geri doğru esner.

Çizge kesme algoritmaları genel enerji formülünü asgariye indirmek için en iyileştirmeye çalışırken her bir olası etiket / fark değeri için çizgeyi oluşturmalıdır. Bu çizgenin oluşumu algoritmadan algoritmaya değişir. Buradaki algoritmada veri enerjileri t-bağlarına, düzgünlük enerjileri ise n-bağlarına yerleştirilmiştir. Etkileşim enerjisi etkin hale geldiğinde t-bağlarına üzerine maliyet eklemekte ve ağ düğümlerini birbirlerine doğru harekete zorlamaktadır. Bu ara çizgeler tüm olası etiketler için oluşturulur. Tüm etiketler için bir kez sorulduğunda bir iterasyon tamamlanmış olur. Bu tekrarlar ağların enerjisi azalmayana kadar devam eder. Ağlar durduklarında birbirleriyle aynı konumlarda değilseler bu sefer etkileşim enerjisi devreye girer ve ağların birbirlerine doğru hareket etmesi sağlanır.

Çizge kesme tabanlı optimizasyon için öncelikle özel bir çizgenin oluşturulması gerekir. α-genişleme algoritması için çizge, kaynak düğümün ilk konfigürasyonu, hedef düğümün α etiket değerini ve diğer her bir düğümün de ilgili pikseli gösterdiği bir çizge her α etiket değeri için oluşturulacaktır. Bu çizge içerisinde veri enerjileri t-bağlarına, düzgünlük enerjileri ise n-bağlarına yerleştirilecektir. Etkileşim enerjisi aktifleştiğinden ağları birbirlerine doğru çekmek için t-bağları üzerine yerleştirilecektir. Etkileşim sürecinde düzgünlük enerjisi ağların konumlarını korumaya zorlayacaktır ve ağların düzgünlüğü bozarak diğer ağa doğru gitmesine karşı koyacaktır. Bu şekilde oluşturulan çizgeler s-t kesme sonucu sadece tek bir etiket değerine sahip olacak şekilde kaynak yâda hedef noktadan ayrılacaklardır. Bu tekrarlar herhangi bir enerji azalması olmayan kadar yâda başka

bir deyişle ağ elemanları tamamen birbirleriyle aynı konumları aldıklarında sonlanacaktır.

α-genişleme algoritması oldukça iyi bir sezgisel yöntem olmasına ve iyi bir hata üst sınırı vermesine rağmen çift ağ yapısı ile birlikte kullanıldığında daha başarılı sonuçlar verecektir. Çizge kesme yöntemini çift ağ ile birlikte kullanan algoritma su adımlarla tarif edilebilir:

- M ağı en düşük fark etiketinden, N ağı en yüksek fark etiketinden başlayacak şekilde konumlanır. İki ağ toplam enerjiyi en aza indirecek şekilde çizge kesme ile optimize edilir. Bu aşamada etkileşim enerjisi devreye girmez.
- ✓ Her iki optimizasyon süreci sonlandığında ağ elemanlarının konumları kontrol edilir. Bu aşamada fark haritasındaki güvenilir ve güvenilir olmayan bölgeler tespit edilir. M ve N ağlarının aynı konumda olmayan elemanları güvenilmez, aynı konumda olan elemanları ise güvenilir olarak işaretlenir.
- ✓ En az bir komşusu güvenilir olan güvenilmez ağ elemanları yarı güvenilir olarak kabul edilir.
- ✓ Yarı güvenilir ağ elemanları için etkileşim enerjisi devreye girer. Hiçbir güvenilmeyen ağ elemanlarının yer değiştirmesine ve yarı güvenilir elemanları kendileri ile düzgünleştirmeleri engellenir.
- ✓ Etkileşim enerjisi, ağları aynı konumları almaya yönlendirir. Bununla birlikte veri ve düzgünlük enerjisi terimleri nedeniyle ağlar birbirlerine yapışmayabilirler.
- ✓ Ağlar aynı konumları aldıklarında optimizasyon tamamlanır. Eğer ağlar aynı konumlara kavuşmamışlarsa önce etkileşim enerjisi kapatılır ve optimizasyon süreci her iki ağ için tekrarlanır. Eğer ağlar yeni ve birbirlerine daha yakın olan bu başlangıç değerlerinden hareket ederek

birbirlerine tamamen kavuşmamışlarsa, etkileşim enerjisi yeniden devreye alınır. Bu işlemler ağlar tamamen kavuşana yada optimizasyonda iyileşme durana kadar aynı şekilde tekrarlanır.

✓ Bu işlemler tamamlandıktan sonra düşük enerjili ağın z koordinatındaki değerleri o konumun gösterdiği piksel için fark değeri olarak atanır.

6.3.2 Deney Sonuçları

Algoritmanın denemeleri Middlebury resim veritabanındaki resimlerle yapılmıştır[Scharstein and Szeliski, 2003][Scharstein et. al., 2001]. Algoritmanın ürettiği fark haritaları hem diğer algoritmaların sonuçlarıyla hem de gerçek fark haritalarıyla karşılaştırılmıştır. Algoritma C++ dilinde, Kolmogorov ve Zabih tarafından yazılmış kütüphane kullanılarak gerçeklenmiştir. Kütüphane Boykov ve Zabih'in geliştirmiş olduğu azami akış/asgari kesme algoritmasının yanında diğer popüler çizge kesme algoritmalarının da kodlarını içermektedir [Boykov et. al. 1999][Kolmogorov and Zabih, 2001][Kolmogorov and Zabih, 2002][Boykov et. al., 1998][Boykov and Kolmogorov, 2002].

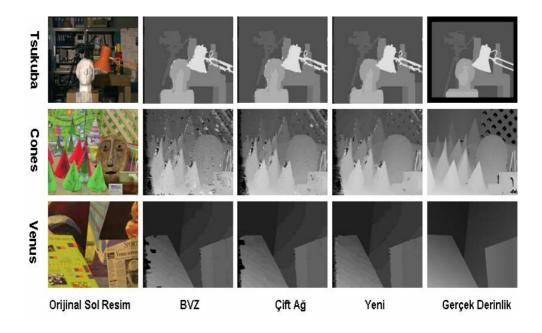
Üç temel deney yapılmıştır. Bu deneylerden ilkinde algoritmanın ürettiği fark haritaları diğer popüler çizge kesme algoritmaları ile karşılaştırılmıştır. Bu algoritmalar BVZ [Boykov et. al., 1998], KZ1[Kolmogorov and Zabih, 2002] ve KZ2 [Kolmogorov and Zabih, 2001] ve çift ağ [Vural and Akgul, "A graph...", 2006] algoritmalarıdır. BVZ algoritması diğer iki algoritmaya göre oldukça basit bir enerji fonksiyonu içerir ve sadece düzgünlük ile veri enerjisi durumlarına bakar. Bu basit yapısı ve kolay uyarlanabilmesi nedeniyle çift ağ kullanan çizge kesme algoritmasının temel algoritması olarak BVZ tercih edilmiştir. Deney sonuçları, çift ağ kullanan yeni çizge kesme algoritmasının ikiden çok kameranın elde ettiği görüntülerden yaralanabilme özelliği olan KZ1 ve kapanmaları algılayabilen KZ2 algoritmalarının sonuçlarına yaklaştığı göstermiştir. Yeni algoritmayı bu algoritmalarla doğrudan yapılacak bir karşılaştırma insaflı olmayacaktır, çünkü çift

ağ yapısı taban algoritmaya bağımlıdır. Sonuçlarda yeni algoritma her zaman BVZ algoritmasından çok daha iyi sonuçlar üretmiştir. İlerideki uygulamalarda taban algoritma olarak daha gelişmiş algoritmalar kullanılacaktır. Bu konudaki beklenti taban algoritmaya göre çok daha başarılı sonuçların elde edilmesidir. Güvenilirlik analizi yaparak çalışan yeni çift ağ algoritması önceki çift ağ algoritmasından daha başarılı sonuçlar elde etmiştir.

Karşılaştırma çizelgesinde K kapanma olmayan bölgeleri, T tüm pikselleri ve S süreksizliklerin olduğu alanları gösterir.

Çizelge 6.4. Güvenilirlik Tabanlı Algoritmanın Karşılaştırılması.

	Tsukuba		Venüs			Cones			
Algoritma	K	T	S	K	T	S	K	T	S
BVZ	1,96	4,20	9,71	2,03	3,69	12,1	19,2	28,3	25,7
KZ1	1,83	2,48	6,42	1,06	1,52	5,53	5,78	12,9	13,2
KZ2	1,33	2,15	6,92	1,22	1,78	5,99	6,08	13,2	13,3
Çift Ağ	1,91	4,13	9,50	1,64	3,29	10,5	9,37	19,6	17,5
Yeni	1,73	3,67	8,75	1,62	3,26	11,4	6,70	17,2	14,9



Şekil 6.7 Güvenilirlik Tabanlı Algoritmanın Sonuçları.

İkinci deneyde ise bir iterasyon sonunda güvenilir bölge olarak adlandırılan alanlardaki hata oranı ölçülmüş ve bu alanlardaki hataların çok az olduğu görülmüştür. Taban algoritması olarak daha kapsamlı bir algoritmanın kullanılması ile bu alandaki sorunlar aşılabilir.

Çizelge 6.5. Güvenilir Alanlardaki Hata Oranları.

Resim Adı	Kapanmamış Alanlar	Tüm Alanlar	Süreksiz Alanlar
Tsukuba	0,87	1,84	4,44
Venus	0,41	0,95	2,47
Cones	2,07	5,39	4,71

7 SONUÇLAR VE ÖNERİLER

Kapanmaların olduğu, dokusuz yada süreksiz bölgeler stereo yönteminde çözülmesi problemli olan yerlerdir. Bu çalışmada bu tür bölgelerin doğru çözülebilmesi için iki farklı sistem önerilmiştir. Önerilen sistemlerden ilki iki eş zamanlı optimizasyon süreci çalıştırarak yerel en iyi noktalara karşı daha kararlı davranabilen bir sistemdir. İkinci sistem ise optimizasyon süreçlerinin birbirleriyle haberleşmesi sırasında ortaya çıkan bilgiyi kullanarak kararlılık analizi yaparak çalışır. Güvenilirlik analizi yapılan resimdeki problemli alanlar eldeki bilgi de kullanılarak daha doğru şekilde tahmin edilebilmektedir.

Sistem yapılan testlerde nesne kenarlarında ve dokusuz alanlarda başarılı sonuçlar vermiştir. Sistem oldukça basit enerji fonksiyonuna sahip BVZ algoritmasında bile güvenilir bölgelerde çok yüksek doğruluk oranları yakalamıştır. Sistemin daha kapsamlı çizge kesme algoritmalarıyla birlikte kullanılmasıyla sonuçlar daha iyi çıkacaktır.

KAYNAKLAR

- 1. Akgul Y., Kambhamettu C., and Stone M. A task-specific contour tracker for ultrasound. In IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 2000.
- 2. Akgul Y. and Kambhamettu C. Recovery and tracking of continuous 3d surfaces from stereo data using a deformable dual-mesh. In International Conference on Computer Vision, pages 765–772, 1999.
- 3. Anandan P. A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision, 2:283–310, 1989.
- 4. BirchField S., and Tomasi C. Depth Discontinuities by Pixel-to-Pixel Stereo. In Proceedings of the Sixth IEEE International Conference on Computer Vision, Mumbai, India, pages 1073-1080, January 1998
- 5. Birchfield S. and Tomasi C. A pixel dissimilarity measure that is insensitive to image sampling. In IEEE Trans. Pattern Analysis and Machine Intelligence, vol: 20, no: 4, pages: 401-406, April, 1998.
- 6. Bleyer M. and Gelautz M. A layered stereo matching algorithm using image segmentation and global visibility constraints. ISPRS Journal of Photogrammetry and Remote Sensing, 59(3):128–150, May 2005.
- 7. Brown M. Z., Burschka D., and Hager G. D. Advances in Computational Stereo. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.25, no.8, pp. 993-1008, August 2003.
- 8. Boykov Y. and Jolly M. P. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D Images. In International Conference on Computer Vision, pages: 105-112, 2001.

- 9. Boykov Y. and Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In EMMCVPR02, page 359 ff., 2002.
- 10. Boykov Y. and Kolmogorov V. Computing geodesics and minimal surfaces via graph cuts. In ICCV International Conference on Computer Vision, pages 26-33, 2003.
- 11. Boykov Y., Veksler O., and Zabih R.. Markov random fields with efficient approximations. In IEEE Computer Vision and Pattern Recognition or CVPR, pages 648–655, 1998.
- 12. Boykov Y., Veksler O., and Zabih R. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222 1239, Nov 2001.
- 13. Cormen T. H., Leiserson C. E., Rivest R. L. and Stein C. Introduction to algorithms second edition. The MIT Press, 2001.
- 14. Dhond U. R., and Aggarwal J. K. Structure From Stereo A Review. In IEEE Trans. Systems, Man, and Cybernetics, vol. 19, pp. 1489-1510, 1989.
- 15. Freedman D. and Drineas P. Energy minimization via graph cuts: settling what is possible. In IEEE Computer Vision and Pattern Recognition, pages II:939–946, 2004.
- 16. Frohlinghaus T. and Buhmann J. M. Regularizing phase-based stereo. In International Conference on Pattern Recognition, 1996.
- 17. Gong M. and Yang Y. H. Fast unambiguous stereo matching using reliability-based dynamic programming. IEEE Trans. Pattern Analysis and Machine Intelligence, 27(6), 2005.

- 18. Greig D., Porteous B., and Seheult A. Exact maximum a posterori estimation for binary images. Journal Royal Statistical Society, B: 51(2):271–279, 1989.
- 19. Hong L. and Chen G. Segment-based stereo matching using graph cuts. In IEEE Computer Vision and Pattern Recognition, pages: 74-81, 2004.
- 20. Hannah M. J. Computer matching of areas in stereo imagery. In Ph.D. Thesis, page Stanford University, 1978.
- 21. Ishikawa H. Exact optimization for markov random fields with convex priors. IEEE Trans. Pattern Analysis and Machine Intelligence, 25(10):1333–1336, October 2003.
- 22. Ishikawa H. and Geiger D. Occlusions, discontinuities, and epipolar lines in stereo. In European Conference on Computer Vision, 1998.
- 23. Kolmogorov V. and Zabih R. Computing visual correspondence with occlusions via graph cuts. In International Conference on Computer Vision, pages II: 508–515, 2001.
- 24. Kolmogorov V. and Zabih R. Multi-camera scene reconstruction via graph cuts. In European Conference on Computer Vision, page III: 82 ff., 2002.
- 25. Kolmogorov V. and Zabih R. What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2):147–159, 2004.
- 26. Koschan A. What is New in Computational Stereo Since 1989: A Survey of Current Stereo Papers. In Technical Report 93-22, Technical University of Berlin, August 1993.

- 27. Matthies L., Kanade T., and Szeliski R. Kalman filter-based algorithms for estimating depth from image sequences. International Journal of Computer Vision, 3(3):209–238, September 1989.
- 28. Nobuhara S. and Matsuyama T. Heterogeneous deformation model for 3D shape and motion recovery from multi-viewpoint images. In 3DPVT, pages 566-573, 2004.
- 29. Paragios N, Chen Y. and Faugeras O. The Handbook of Mathematical Models in Computer Vision, Springer, 2005.
- 30. Scharstein D. and Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1-3):7–42, April 2002.
- 31. Scharstein D., Szeliski R. and Zabih R. A taxonomy and evaluation of dense two-frame stereo methods. Workshop on Stereo and Multi-Baseline Vision, 2001.
- 32. Scharstein D. and Szeliski R. High-accuracy stereo depth maps using structured light. IEEE Computer Vision and Pattern Recognation, pages: 195-202, 2003.
- 33. Sun C. Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques. In International Journal of Computer Vision. 47(1/2/3):99-117, May 2002.
- 34. Szeliski R. and Zabih R.. An Experimental Comparison of Stereo Algorithms. In *IEEE Workshop on Vision Algorithms*, pages 1-19, September 1999.
- 35. Tao H., and Sawhney H. "Gloabal matching criterion and color segmentation based stereo," in Workshop on Applications of Computer Vision, pp. 246-253, 2000.

- 36. Tappen M. F. and Freeman W. T. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In International Conference on Computer Vision, pages 900–907, 2003.
- 37. Terzopoulos D. Regularization of inverse visual problems involving discontinuities. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(4):413–424, 1986.
- 38. Trucco E., and Verri A. Introductory Techniques for 3-D Computer Vision, Prentice Hall, New Jersey, 1998.
- 39. Vural U. and Akgul Y. S. A Multiple Graph Cut Based Approach For Stereo Analysis, LNCS 4174, pages 677–687, Springer, 2006.

ÖZGEÇMİŞ

1980 yılında İstanbul'da doğdu. İlk ve orta öğretimini 1988–1996 yılları arasında sırasıyla Gebze Osmangazi İlkokulu ve Gebze Anadolu Lisesi'nde tamamladı. Ordu Fen Lisesi'nde bir yıl okuduktan sonra 1999 yılında Gebze Sarkuysan Lisesi'nden 1999 yılında mezun oldu. Aynı yıl girdiği Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2003 yılında mezun oldu. 2004 yılında Gebze Yüksek Teknoloji Enstitüsü Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde yüksek lisans eğitimine başladı. Halen Gebze Yüksek Teknoloji Enstitüsü Mühendisliği Bölümü'nde araştırma görevlisi olarak çalışmaktadır.