

**ANKARA ÜNİVERSİTESİ
FENBİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**VİDEO DİZİLERİNDE KENAR BULMA ALGORİTMASI
İLE ÖBEK EŞLEME YÖNTEMİNE DAYALI
HAREKET KESTİRİMİ**

Fatih BOYACI

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2010**

Her hakkı saklıdır

TEZ ONAYI

Fatih BOYACI tarafından hazırlanan “**Video Dizilerinde Kenar Bulma Algoritması İle Öbek Eşleme Yöntemine Dayalı Hareket Kestirimi**” adlı tez çalışması 16/02/2010 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Ziya TELATAR

Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı

Jüri Üyeleri :

Başkan : Yrd. Doç. Dr. Hakan TORA

Atılım Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı

Üye : Doç. Dr. Ziya TELATAR

Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı

Üye : Yrd. Doç. Dr. Hakkı Alparslan ILGIN

Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. Orhan ATAKOL

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

VIDEO DİZİLERİNDE KENAR BULMA ALGORİTMASI İLE ÖBEK EŞLEME YÖNTEMİNE DAYALI HAREKET KESTİRİMİ

Fatih BOYACI

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ziya TELATAR

Hareket kestirimi özellikle video kodlama ve işleme sistemlerinde görüntü iletiminde bant genişliği ve verici gücünden tasarruf sağlamak için kullanılır. Hareket kestirimi amacıyla, günümüze kadar farklı yapıda birçok çalışma yapılmış olup, bu çalışmalardan öbek eşleme ISO, MPEG-1, MPEG-2, MPEG-4, ITU H.261, H.263 ve H.26L v.b. birçok uluslararası video kodlama standartları tarafından desteklenmiştir. Bu tez çalışmasında, imge dizilerinde bir veya daha fazla hareketli objenin öbek eşleme metoduna dayalı hareket kestirimi ve hareket vektörlerinin belirlenmesi konusu ele alınmıştır. Çerçeveler arasında çoklu değişen özellikler ve geçici oluşumlar tanımlanmış, bu bilgilere dayanarak hareket değerlendirmesi yapılmıştır. Hareket kestiriminin video kalitesine yaklaştırılabilmesi amacıyla hareketli nesnenin sınırlarının bulunarak yalnızca bu bölgelerin referans çerçevesinde aranması gerçekleştirilmiştir. Bu işlem, başlangıçta olası hareketli bir nesnenin, tüm çerçevede aranması yerine sadece nesnenin bulunduğu yerin belirlenerek ve daha sonraki çerçevede yalnızca bu öbeğin yakınlarında aranmasını sağlayarak, arama süresi açısından belirgin bir kazanç elde edilmiştir.

Şubat 2010, 109 sayfa

Anahtar Kelimeler: Hareket kestirimi teknikleri, öbek eşleme kriterleri, öbek arama algoritmaları, kenar algılaması yöntemleri.

ABSTRACT

Master Thesis

BLOCK MATCHING BASED MOTION ESTIMATION IN VIDEO SEQUENCES BY EDGE DETECTION ALGORITHM

Fatih BOYACI

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electronics Engineering

Supervisor: Assoc. Prof. Dr. Ziya TELATAR

Motion estimation is used to save transmitter power and bandwidth in image transmission, especially in video coding and video processing systems. Up to now lots of studies in different constitution were performed for motion estimation and block matching, one of which studies was supported by lots of international video coding standarts such as ISO, MPEG-1, MPEG-2, MPEG-4, ITU H.261, H.263, H.26L etc. In this study, motion estimation of one or more moving object based on the block matching method and determination of motion vectors were discussed. Multi-varient features and temporary formations between frames were introduced and motion evaluation based on these information was performed. After getting the edges of the moving object, scanning of only these regions in the referance frame was implemented for the approximation of the motion estimation to video quality. This process made considerable profit in scanning time, by instead of first scanning of a probable moving object in entire frame, just by determination of the place of the object where it was and providing scanning of this object only close to this block.

February 2010, 109 pages

Key Words: Motion estimation techniques, block matching criterions, block search algorithm, edge detection methods.

TEŐEKKÜR

Çalıőmamın her aőamasında önerileri ile beni yönlendiren, , ihtiyaç duyduėum her an yanımda olarak bana bilimsel danıőmanlık veren, beni aydınlatan ve çalıőmalarımı yönlendiren sevgili danıőmanım Doç. Dr. Ziya TELATAR'a teőekkürü bir borç bilirim.

Tez çalıőması ve tezin yazımı süresince sevgilerini esirgemeyen, birçok fedakârlık göstererek beni destekleyen anneme, babama ve eőime en derin duygularımla teőekkür ederim.

Fatih BOYACI

Ankara, Őubat 2010

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
SİMGELER DİZİNİ.....	viii
ŞEKİLLER DİZİNİ.....	xi
ÇİZELGELER DİZİNİ	xv
1. GİRİŞ.....	1
2. KURAMSAL TEMELLER.....	4
2.1 İki Boyutlu Video Dizilerinde Öbek Eşleme Temelli Hareket Kestirimi ve Yöntemleri.....	4
2.1.1 Giriş	4
2.1.2 Hareket vektörleri.....	5
2.1.3 Eşleme ölçütleri.....	6
2.1.3.1 Farkların karelerinin ortalaması (mean squared difference, MSD)	9
2.1.3.2 Hataların karelerinin toplamı (sum of squared error, SSE).....	10
2.1.3.3 Mutlak farkların ortalaması (mean absolute difference, MAD).....	10
2.1.3.4 Mutlak farkların toplamı (sum of absolute difference, SAD).....	11
2.1.4 Öbek eşleme arama teknikleri.....	11
2.1.4.1 Tam arama, TA (full search, FS)	12
2.1.4.2 İki boyutlu logaritmik arama, İBLA (2-D logarithmic search, 2DLS).....	14
2.1.4.3 Üç adımda arama, ÜAA (three step search, TSS)	16
2.1.4.4 Yeni üç adımda arama, YÜAA (new three step search, NTSS).....	17
2.1.4.5 Çapraz arama, ÇA (cross search, CS)	19

2.1.4.6 Eşlenik arama, EA (conjugate direction search, CDS)	20
2.1.4.7 Dört adımda arama, DAA (four step search, 4SS).....	21
2.1.4.8 Baklava desenli arama, BDA (diamond search, DS).....	22
2.1.4.9 Uyarlanıır kök örüntü arama, UKÖA (adaptive root pattern search, ARPS)	24
2.1.4.10 Basit ve etkili üç adımda arama, BEÜAA (simple and efficient three step search, SETSS)	25
2.1.5 Tepe sinyal gürültü oranı, TSGO (Peak Signal To Noise Ratio, PSNR).....	27
2.2 Kenar Algılaması (Edge Detection)	28
2.2.1 Gradyana dayalı yöntemler	29
2.2.2 Laplace işlecine dayalı (Laplacian based) yöntemler	34
2.2.3 Canny kenar algılayıcılar	38
3. MATERYAL ve YÖNTEM	40
3.1 Materyal	40
3.2 Yöntem	43
3.2.1 Yöntem-1	44
3.2.2 Yöntem-2.....	48
3.2.3 Yöntem-3.....	51
3.2.4 Yöntem-4.....	53
3.2.5 Literatür karşılaştırması	57
4. BULGULAR.....	59
4.1 “Pazarlamacı” Görüntü Dizisi İçin Bulgular	61
4.1.1 “Pazarlamacı” görüntü dizisi için hesaplamalar (aday öbek sayısı).....	61
4.1.2 “Pazarlamacı” görüntü dizisi için öbek sayısı (ÖS).....	63

4.1.3 “Pazarlamacı” görüntü dizisi için HVOÖ	64
4.1.4 “Pazarlamacı” görüntü dizisi için HVBY	66
4.1.5 “Pazarlamacı” görüntü dizisi için TSGO (PSNR).....	68
4.1.6 “Pazarlamacı” görüntü dizisi için arama süresi	70
4.2 “Haberler” Görüntü Dizisi İçin Bulgular	72
4.2.1 “Haberler” görüntü dizisi için hesaplamalar (aday öbek sayısı)	72
4.2.2 “Haberler” görüntü dizisi için öbek sayısı (ÖS)	74
4.2.3 “Haberler” görüntü dizisi için HVOÖ.....	75
4.2.4 “Haberler” görüntü dizisi için HVBY.....	77
4.2.5 “Haberler” görüntü dizisi için TSGO (PSNR)	79
4.2.6 “Haberler” görüntü dizisi için arama süresi	81
4.3 “Mobil” Görüntü Dizisi İçin Bulgular	83
4.3.1 “Mobil” görüntü dizisi için hesaplamalar (aday öbek sayısı)	83
4.3.2 “Mobil” görüntü dizisi için öbek sayısı (ÖS)	85
4.3.3 “Mobil” görüntü dizisi için HVOÖ.....	86
4.3.4 “Mobil” görüntü dizisi için HVBY.....	88
4.3.5 “Mobil” görüntü dizisi için TSGO (PSNR).....	90
4.3.6 “Mobil” görüntü dizisi için arama süresi.....	92
4.4 Literatür Karşılaştırması.....	94
4.4.1 “Pazarlamacı” görüntü dizisi için TSGO (PSNR) karşılaştırması	94
4.4.2 “Pazarlamacı” görüntü dizisi için arama süresi karşılaştırması	94
4.4.3 “Haberler” görüntü dizisi için TSGO (PSNR) karşılaştırması	95
4.4.4 “Haberler” görüntü dizisi için arama süresi karşılaştırması	95
5. TARTIŞMA VE SONUÇ	96

5.1 300 Çerçevelik Görüntü Dizileri İçin Sonuçlar.....	97
5.2 100 Çerçevelik Görüntü Dizileri İçin Sonuçlar.....	103
KAYNAKLAR.....	106
ÖZGEÇMİŞ.....	109

SİMGELER DİZİNİ

2DLS	Two Dimension Logarithmic Search
4SS	Four Step Search
ARPS	Adaptive Rood Pattern Search
B_k	k. Çerçeve Üzerinde Çalışılan Öbek
BDA	Baklava Desenli Arama
BEÜAA	Basit ve Etkili Üç adımda Arama
CDS	Conjugate Direction Search
CS	Cross Search
ÇA	Çapraz Arama
$Ç_k$	k. Çerçeve
DAA	Dört Adımda Arama
DS	Diamond Search
dx,dy	Arama konum hareket vektörleri
EA	Eşlenik Arama
FS	Full Search
GÇKH	Güncel Çerçeve Kenar Haritası
HKO	Hataların Karelerinin Ortalaması
HVBY	Hareket Vektörünü Bulma Yüzdesi
HVOÖ	Hareket Vektörünün Olduğu Öbek
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITU	International Telecommunications Union

İBLA	İki Boyutlu Logaritmik Arama
İKH	İşlenmiş Kenar Haritası
LoG	Laplacian of Gauss
MAD	Mean Absolute Difference
MAE	Mean Absolute Error
MB	Mega Bayt
maksP	Çerçevede Kullanılan Maksimum Parlaklık Değeri
$m_f(n_1, n_2)$	Yerel Ortalama Kestirim Değeri
MPEG	Moving Picture Experts Group
MSD	Mean Squared Difference
MSE	Mean Squared Error
NTSC	National Television Standards Committee
NTSS	New Three Step Search
ÖS	Öbek Sayısı
p	Arama Parametresi
PAL	Phase Alternating Line
PSNR	Peak Signal To Noise Ratio
QCIF	Quarter CIF
RAM	Random Access Memory
RÇKH	Referans Çerçeve Kenar Haritası
s	Adım Büyüklüğü
SAD	Sum of Absolute Difference
SAE	Sum of Absolute Error
SETSS	Simple and Efficient Three Step Search

sn	Saniye
SQCIF	Sub-QCIF
SSD	Sum of Squared Difference
SSE	Sum of Squared Error
TA	Tam Arama
TSGO	Tepe Sinyal Gürültü Oranı
TSS	Three Step Search
UKÖA	Uyarlanır Kök Örüntü Arama
ÜAA	Üç Adımda Arama
YD	Yerel Değişinti
YK1	Birinci Yöntem
YK2	İkinci Yöntem
YK3	Üçüncü Yöntem
YK4	Dördüncü Yöntem
YÜAA	Yeni Üç Adımda Arama

ŞEKİLLER DİZİNİ

Şekil 2.1 Hareket vektörleri gösterimi	5
Şekil 2.2 Geriye ve ileriye hareket kestirimi	6
Şekil 2.3 Arama öbeğinin önceki veya sonraki çerçevede gösterimi	8
Şekil 2.4 Arama penceresi parametreleri	9
Şekil 2.5 Tam arama	13
Şekil 2.6 İBLA algoritmasının ilk adımı.....	14
Şekil 2.7 İki boyutlu logaritmik arama	16
Şekil 2.8 Üç adımda arama	17
Şekil 2.9 Yeni üç adımda arama.....	18
Şekil 2.10 Çapraz arama	19
Şekil 2.11.a. Son adım "+", b. Son adım "X"	20
Şekil 2.12 Eşlenik arama.....	20
Şekil 2.13 DAA örüntüsü.....	22
Şekil 2.14 Dört adımda arama	22
Şekil 2.15 Dört adımda arama –merkezdeki hareketler	22
Şekil 2.16.a. Baklava desenli arama- Eşleme ölçütü köşede en küçük, b.Baklava desenli arama- Eşleme ölçütü ortada en küçük.....	23
Şekil 2.17 Uyarlanırlık kök örüntüsü.....	24
Şekil 2.18 Basit ve etkili üç adımda aramada çeyrekler	26
Şekil 2.19 Basit ve etkili üç adımda aramada MAD.....	26
Şekil 2.20 Parlaklık değişim fonksiyonu	29
Şekil 2.21 Bir boyutlu (1-D) kenar algılaması blok çizimi	30
Şekil 2.22 Bir boyutlu (1-D) kenar algılaması eşik seçimi	30

Şekil 2.23 İki boyutlu (2-D) gradyana dayalı kenar algılaması blok şeması	31
Şekil 2.24 Yönlü kenar algılayıcılarında kullanılan dürtü tepkeleri.....	32
Şekil 2.25 Sobel kenar algılamasının dürtü tepkeleri	33
Şekil 2.26 Robert kenar algılamasının dürtü tepkeleri.....	34
Şekil 2.27 Laplace işlecine dayalı kenar algılamasının dürtü tepkelerine örnekler	34
Şekil 2.28 İki boyutlu (2-D) Laplace işlecine dayalı kenar algılaması blok şeması.....	36
Şekil 2.29 LoG operatöründe kullanılan maskenin ağırlık gösterimi.....	37
Şekil 2.30.a. Gauss fonksiyonu, b. Birinci türevi, c. Gauss fonksiyonun Laplace'ı.....	38
Şekil 3.1 Matlab arayüzü.....	40
Şekil 3.2 Microsoft Office Excel arayüzü.....	41
Şekil 3.3.a. Test görüntüleri-Pazarlamacı, b. Test görüntüleri-Haberler, c. Test görüntüleri-Mobil.	42
Şekil 3.4 Çalışmaların ana blok şeması	44
Şekil 3.5 Birinci yöntem blok şeması	45
Şekil 3.6.a. Referans çerçeve, b. Güncel çerçeve.....	46
Şekil 3.7.a. Referans çerçeve-öbeklere ayrılmış, b. Güncel çerçeve-öbeklere ayrılmış.....	46
Şekil 3.8.a. Hareket vektörleri, b. Dengelenmiş çerçeve	47
Şekil 3.9.a. LoG kenar haritası, b. Canny kenar haritası, c.Robert kenar haritası, d.Sobel kenar haritası	49
Şekil 3.10.a. İkinci yöntem RÇKH, b. İkinci yöntem GÇKH.....	49

Şekil 3.11 İkinci yöntem blok şeması	50
Şekil 3.12.a. VE kapısı çıkışı kenar haritası, b. Üçüncü yöntem kenar haritası.....	51
Şekil 3.13 Üçüncü yöntem blok şeması	52
Şekil 3.14 Dördüncü yöntem blok şeması.....	54
Şekil 3.15.a. Dördüncü yöntem RÇKH, b. Dördüncü yöntem GÇKH, c. Dördüncü yöntem İKH	56
Şekil 3.16 Hareket değişimini gösteren ikili görüntü.....	57
Şekil 4.1 Pazarlamacı.....	61
Şekil 4.2 Pazarlamacı-hesaplamalar (aday öbek sayısı)	62
Şekil 4.3 Pazarlamacı-öbek sayısı	63
Şekil 4.4 Pazarlamacı-HVOÖ	65
Şekil 4.5 Pazarlamacı-HVBY.....	67
Şekil 4.6 Pazarlamacı-TSGO	69
Şekil 4.7 Pazarlamacı-arama süreleri.....	71
Şekil 4.8 Haberler	72
Şekil 4.9 Haberler – hesaplamalar (aday öbek sayısı)	73
Şekil 4.10 Haberler- öbek sayısı.....	74
Şekil 4.11 Haberler - HVOÖ.....	76
Şekil 4.12 Haberler - HVBY	78
Şekil 4.13 Haberler - TSGO.....	80
Şekil 4.14 Haberler – arama süresi	82
Şekil 4.15 Mobile.....	83
Şekil 4.16 Mobil – hesaplamalar (aday öbek sayısı)	84
Şekil 4.17 Mobil- öbek sayısı.....	85

Şekil 4.18 Mobil - HVOÖ	87
Şekil 4.19 Mobil - HVBY	89
Şekil 4.20 Mobil - TSGO	91
Şekil 4.21 Mobil – arama süresi	91
Şekil 5.1.a. Öbek sayısı farkı (pazarlamacı-haberler), b. Öbek sayısı farkı (mobil-haberler).....	98
Şekil 5.2.a. Öbek eleme oranı-Haberler, b.Öbek eleme oranı-Pazarlamacı, c.Öbek eleme oranı-Mobil.....	99
Şekil 5.3.a. Arama süresi hızlanması-Haberler, b. Arama süresi hızlanması-Pazarlamacı, c. Arama süresi hızlanması-Mobil	101

ÇİZELGELER DİZİNİ

Çizelge 3.1 Video çerçeve formatları	42
Çizelge 3.2 Üçüncü yöntem İKH değerler tablosu.....	53
Çizelge 3.3 Dördüncü yöntem İKH değerler tablosu-RÇKH ağırlıklı	55
Çizelge 3.4 Dördüncü yöntem İKH değerler tablosu-GÇKH ağırlıklı	55
Çizelge 4.1 Bulgular kısaltma-arama teknikleri.....	60
Çizelge 4.2 Bulgular kısaltma-yöntemler	60
Çizelge 4.3 Pazarlamacı-hesaplamalar (aday öbek sayısı)	61
Çizelge 4.4 Pazarlamacı-öbek sayısı	63
Çizelge 4.5 Pazarlamacı-HVOÖ.....	64
Çizelge 4.6 Pazarlamacı-HVBY.....	66
Çizelge 4.7 Pazarlamacı-TSGO.....	68
Çizelge 4.8 Pazarlamacı-arama süreleri.....	70
Çizelge 4.9 Haberler-hesaplamalar (aday öbek sayısı).....	72
Çizelge 4.10 Haberler-öbek sayısı.....	74
Çizelge 4.11 Haberler - HVOÖ	75
Çizelge 4.12 Haberler - HVBY	77
Çizelge 4.13 Haberler - TSGO	79
Çizelge 4.14 Haberler – arama süresi	81
Çizelge 4.15 Mobil-hesaplamalar (aday öbek sayısı)	83
Çizelge 4.16 Mobil- öbek sayısı.....	85
Çizelge 4.17 Mobil - HVOÖ	86
Çizelge 4.18 Mobil - HVBY	88
Çizelge 4.19 Mobil - TSGO	90

Çizelge 4.20 Mobil – arama süresi	92
Çizelge 4.21 Pazarlamacı- TSGO, Nasır (2007)	94
Çizelge 4.22 Pazarlamacı- TSGO, kenar algılama	94
Çizelge 4.23 Pazarlamacı- arama süresi, Nasır (2007)	94
Çizelge 4.24 Pazarlamacı- arama süresi, kenar algılama	94
Çizelge 4.25 Haberler- TSGO, Nasır (2007) yöntemi	95
Çizelge 4.26 Haberler- TSGO, kenar algılama temelli yöntem	95
Çizelge 4.27 Haberler– arama süresi, Nasır (2007) yöntemi	95
Çizelge 4.28 Haberler– arama süresi, kenar algılama temelli yöntem	95

1. GİRİŞ

Video, ardışık çerçevelerin hızlı bir şekilde gösterilmesiyle elde edilen “hareketli görüntü” ile “sesin” aynı anda oynatılmasıyla oluşan çoklu ortam uygulamasıdır. Bu hız PAL (Phase Alternating Line) sisteminde 25 görüntü/saniye, NTSC (National Television Standards Committee) sisteminde 30 görüntü/saniyedir. Video sinyalinde bir çerçevede komşu pikseller birbiri ile ilintilidir. Bu ilintiye **uzamsal ilinti** (spatial correlation) denir. Uzamsal ilinti, bir görüntünün birbirlerinden bir kenarla ayrılmış bölgelerden oluşmasından kaynaklanmaktadır (Lim 1990) .

İnsan görsel sistemi, kenar sinyallerini geçici süzgeçleyerek hareketleri tanır ve kenarların piksel yoğunluğu üzerine bir avantajı da nesnelerin fiziksel özelliği ile yakın bir bağın olmasıdır (Marr vd. 1980). Nesnelerin kenarlarının algılama ile görüntü onarımı, görüntüyü bölümlerine ayırma, nesne tanıma, görüntüde nesne ile arka planı birbirinden ayırma, hedef takibi vb çeşitli alanlarda çalışma yapmak mümkündür.

Video sinyallerinde komşu çerçevelerdeki pikseller, hareket yoksa aynıdır. Buna da **zamansal ilinti** (temporal correlation) denir. Ardışık çerçevelerin zamansal ilintisinden faydalanarak sabit, değişmeyen çerçeve elamanları iletilmeyerek sadece hareket eden görüntü elemanları iletilerek görüntü kalitesinde mümkün olduğunca az kayıpla en verimli iletim yapılır.

Bir görüntü dizisindeki çerçeveler arasındaki hareketleri algılayabilmek, hareketlerin yönünü ve büyüklüğünü (**hareket vektörlerini**) kestirebilmek için **hareket kestirimi** yöntemleri kullanılmaktadır. Hareket kestirimi yöntemleri yalnızca video çerçeve iletiminde bant genişliğinde verim sağlamak için değil, video sıkıştırması, hedef takibi, trafik ve güvenlik uygulamaları için de önem teşkil eder.

Hareket kestirimi için genel tabanlı (global based), piksel tabanlı (pixel based), öbek tabanlı (block based), bölge tabanlı (region based) yöntemler kullanılmaktadır (Bovik 2000).

Öbek tabanlı yöntemler; sadeliği, donanım karmaşıklığının daha az olması, hareket vektörlerinin kodlanmasındaki verimliliği gibi avantajlarından ötürü ITU-T H.261 (ITU-T SG15 1993), H.263 (ITU-T SG15 1996), ISO MPEG–1 (ISO/IEC JTC 1/SC29 1993), MPEG–2 (ISO/IEC JTC 1/SC29 1996), MPEG–4 (ISO/IEC 14496–2 1998) v.b. birçok uluslararası video kodlama standartları tarafından hareket kestirimi yöntemleri arasında benimsenmiştir (Bovik 2000).

Öbek eşleştirmeyi temel alan yöntemlerde, üzerinde çalışılan **güncel çerçeve** (current frame) birbiri ile örtüşmeyen öbeklere ayrılır. Uygulamalarda genellikle karesel öbekler kullanılır. Her bir öbek bir önceki veya bir sonraki **referans çerçevede** (reference frame) oluşturulan **arama penceresinde** (search window) belirli bir eşleme ölçütü değerine göre aranarak en iyi eşleşen öbek bulunur. En iyi eşleşen öbek ile güncel çerçevede çalışılan öbek arasındaki konumsal fark **hareket vektörünü** (motion vector) belirtir.

Öbek eşlemede, genellikle öbeklerin yatay ve dikey yönde yer değiştirmeleri öteleme biçiminde olduğu ve dönme (rotation) olmadığı varsayılmıştır. Öbek eşlemede, öbekteki yapaylık, hareketli nesnenin kenar kısımlarının tahmin edilmesinde zayıflık gibi dezavantajlar görülmektedir. Öbekteki yapaylık sorunun üstesinden gelmek için, öbek büyüklüğü sabit değil değişken seçilir. Değişken öbek büyüklükleri, sabit öbek büyüklüğü temelinde çalışan yaygın kodlama standartları ile doğrudan uyumluluk göstermemektedir (CCITT Report SG xV 1990, ISO/IEC Report JTCI CD 11172 1991).

Arama tekniklerinin en sadesi ve arama zaman maliyeti en fazla olan **tam aramada, TA** (full search, FS) çalışılan öbek ile arama penceresi içerisindeki tüm aday öbekler karşılaştırılır. Tam aramanın zaman maliyetini azaltmak için çeşitli hızlı arama teknikleri geliştirilmiştir. Hızlı arama teknikleri arama penceresindeki aday öbek

sayısını azaltma amacı ile kendilerine özgü **arama örüntüsüne** (search pattern) sahiptirler ve arama penceresindeki pek çok öbek göz ardı edilir. Bu nedenle hızlı arama tekniklerinin bulduğu en iyi eşleşen öbeğinin tam arama tekniği ile bulunan en iyi eşleşen öbek olmama ihtimali vardır.

Koga (1981) tarafından üç adımda en iyi eşleşen öbeği bulan **üç adımda arama, ÜAA** (three step search, TSS) tekniği, Jain (1981) tarafından **iki boyutlu logaritmik arama, İBLA** (two dimension logarithmic search, 2DLS) Srinivasan ve Rao (1985) tarafından **eşlenik arama, EA**(conjugate direction search,CDS), Reoxiang, Bing ve Liou (1994) tarafından **yeni üç adımda arama, YÜAA** (new three step search, NTSS) , Po ve Ma (1996) tarafından **dört adımda arama, DAA** (four step search, 4SS) ve **çapraz arama, ÇA** (conjugate direction search, CDS), Jianhua ve Ming (1997) tarafından **basit ve etkili üç adımda arama, BEÜAA** (simple and efficient search , SES), Zhu ve Ma (2000) tarafından **baklava desenli arama, BDA** (diamond search, DS) teknikleri geliştirilmiştir.

Araştırılan bloğun çevre öbeklerdeki hareket yönünde harekete sahip olma ihtimalinin yüksek olmasını dikkate alarak Nie ve Ma (2002) tarafından **uyarlanır kök örüntü arama, UKÖA** (adaptive root pattern search) algoritması geliştirilmiştir.

Bu tez çalışmasında video dizisindeki nesnelerin hareket kestirimi için kenar algılama yöntemlerinden yararlanarak video dizilerinin kenar haritaları elde edilir. Araştırılan güncel çerçeve hareketli nesne, hareketin olmadığı geri plan bölgesi ve ardışık çerçevelerde sabit olan yani hareketsiz nesnelere şeklinde üç kategoride incelenir.

2. KURAMSAL TEMELLER

2.1 İki Boyutlu Video Dizilerinde Öbek Eşleme Temelli Hareket Kestirimi ve Yöntemleri

2.1.1 Giriş

Doğada nesnelere üç boyutlu oldukları için hareketleri de üç boyutludur. İki boyutlu görüntü dizilerinde, bu üç boyutlu hareket çeşitli izdüşümü sistemleri ile iki boyutlu harekete dönüştürülür. Görüntü dizilerinde hareketin modellenmesine **hareket kestirimi** (motion estimation) denir. Bir başka anlatımla hareket kestirimi, bir görüntü dizisindeki nesnelere bir çerçeveden diğer çerçeveye olan yer değişiminin vektörel (yön ve büyüklük) olarak ölçüldüğü tekniktir. Hareket kestirimi gürültüyü süzgeçleme, bulanıklık giderme, görüntü sıkıştırma, görüntü iletimi, hedef izleme, trafik ve güvenlik kamerası uygulamaları gibi pek çok alanda uygulama bulmuştur.

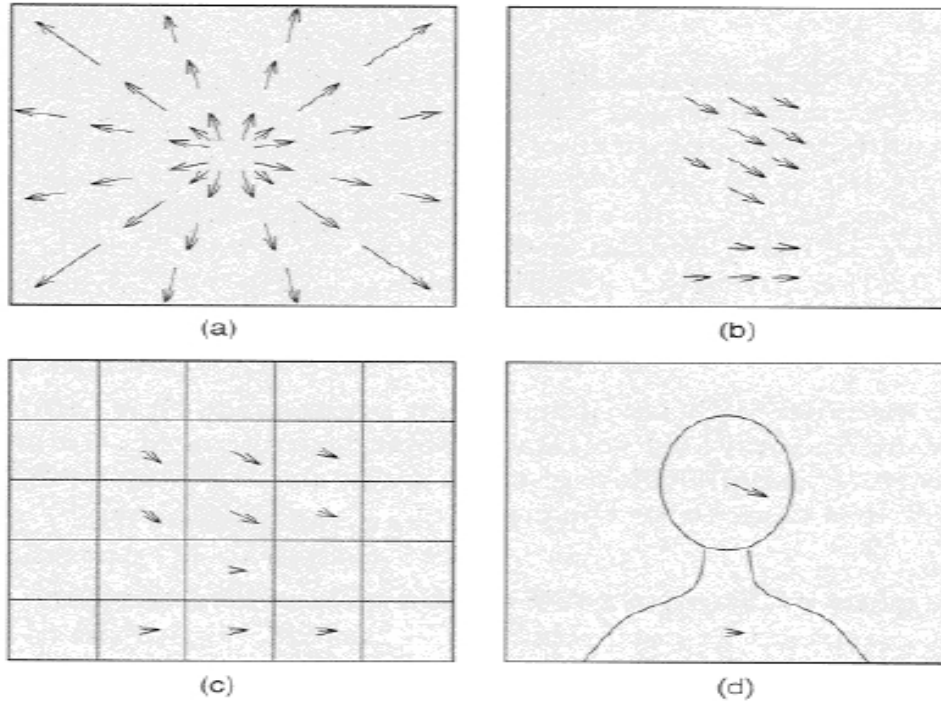
Görüntü dizisindeki hareket, nesnenin ve kameranın hareketi olmak üzere iki temelde ele alınır. Nesne hareketinin modellenmesi için kamera hareketinden kaynaklanan hareketin analizi ve ele alınması önemlidir.

Hareket vektörleri (motion vectors) şekil 2.1'de görülen (şekil 2.1'de a,b,c,d sırasıyla) genel tabanlı (global based), piksel tabanlı (pixel based), öbek tabanlı (block based), bölge tabanlı (region based) yöntemlerden uygun olanı kullanılarak bulunur (Bovik 2000). Bu uygunluk ifadesi hareket vektörlerini nerede ve ne amaçla bulmak istediğimize bağlı olarak değişir.

Örneğin görüntü dizisinde genel hareketin önemli olduğu görüntü dengeleme (stabilization) için genel tabanlı hareket kestirimi kullanılabilir. Kamera hareketi daha baskındır. Genel tabanlı yöntemde öbek tabanlı yöntemdeki gibi tüm çerçeve öbeklere ayrılmaz, tek bir öbek kullanılır. Öbek büyüklüğü hareket kestiriminde önem teşkil eder.

Piksel tabanlı yöntemde optik akış yöntemi de denilmekle birlikte bu yöntemde her bir piksel için bir hareket vektörü bulunur. Yüksek hesaplama karmaşıklığına karşın uygulama alanları yaygındır. Piksel tabanlı yöntemler de kendi içinde fark teknikleri, frekans temelli teknikler ve eşleştirme teknikleri olmak üzere üç grup altında toplanır.

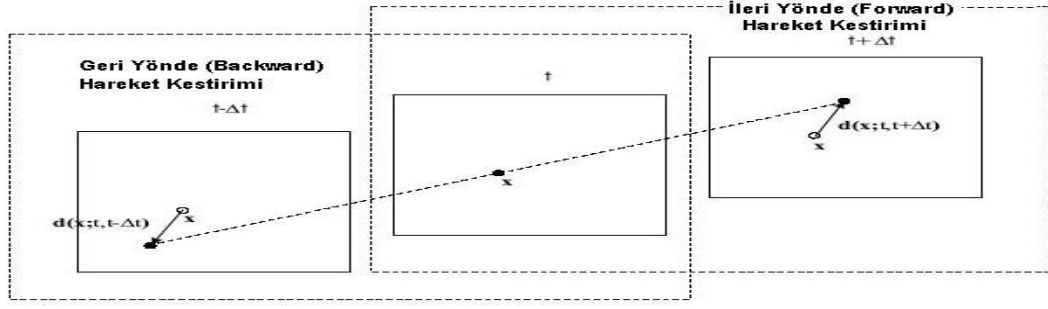
Bölge tabanlı yöntemde çerçevenin bölütleri (segment) bulunur ve özgül (specific) bölütler belirlenir. Özyineli (iterative) bölütleme ve kestirim yapma durumu söz konusudur.



Şekil 2-1 Hareket vektörleri gösterimi

2.1.2 Hareket vektörleri

Görüntü dizilerinde öngörüm, **çerçeve içi** (intra frame) veya **çerçeveler arası** (inter frame) yapılabilir. Çerçeve içi öngörümde uzamsal (spatial), çerçeveler arası öngörümde zamansal (temporal) öngörüm denir. Zamansal öngörümde hareket vektörleri tespit edilir.



Şekil 2-2 Geriye ve ileriye hareket kestirimi

Şekil 2.2’de (Borshukov vd. 1997) geriye ve ileriye hareket kestirimi verilmiştir. Δt zaman aralıkları ile ardışık gelen çerçevelerde, t anındaki güncel çerçeve ile referans olarak kabul edilen çerçeveden yararlanarak ileri veya geri yönlü hareket kestirimi yapılabilir. Referans çerçeveyi $t+\Delta t$ anındaki çerçeve olarak alırsak yapılan kestirime ileri yönde hareket kestirimi; $t-\Delta t$ anındaki çerçeve olarak alırsak geri yönde hareket kestirimi denir.

2.1.3 Eşleme ölçütleri

Birçok imge eşleme metodunda, iki imge veya imgelerin iki parçası piksel-piksel temelinde incelenir. Bu iki imge veya imge parçası uzamsal bir imge dizisinden seçilebilir. Örneğin, aynı zamanda aynı nesneye yönlendirilmiş iki farklı sensor tarafından alınmış iki çerçeve veya aynı sensor tarafından farklı zamanlarda alınan iki çerçeve gibi. Burada amaç iki imge veya imge parçası arasındaki farkı belirlemektir. Bu tip uygulamalar arasında imge kayıt ve şablon (template) eşleme verilebilir. Bunlardan birincisi imgelerin uzamsal kayıtları ile ilgilidir. İkincisi ise, imgedeki bir nesneyi algılamada kullanılır.

Benzerlik ölçüsü yani ilinti, eşleme sürecinde önemli bir yere sahiptir. t_n ve t_{n-1} anlarındaki (p,q) piksel büyüklüğündeki iki çerçeve arasındaki ilinti fonksiyonu aşağıdaki gibi tanımlanır (Anuta, 1969):

$$C(s, t) = \frac{\sum_{j=1}^p \sum_{k=1}^q f_n(j, k) * f_{n-1}(j + s, k + t)}{\sqrt{\sum_{j=1}^p \sum_{k=1}^q f_n^2(j, k)} * \sqrt{\sum_{j=1}^p \sum_{k=1}^q f_{n-1}^2(j + s, k + t)}} \quad (2.1)$$

Bu aynı zamanda normalize iki boyutlu çapraz ilinti fonksiyonu olarak da bilinir.

Öbek eşlemede maksimum benzerliği ya da ilintiyi bulmak yerine, buna eşdeğer fakat daha hesaplamalı bir yöntem, minimum benzersizliği yani bir başka bir deyişle eşleme hatasını bulmak tercih edilir. t_n ve t_{n-1} anlarındaki iki çerçeve arasındaki benzersizlik aşağıdaki gibi tanımlanır.

$$D(s, t) = \frac{1}{\text{Im}} \sum_{j=1}^p \sum_{k=1}^q M(f_n(j, k), f_{n-1}(j + s, k + t)) \quad (2.2)$$

Im: İmge boyutu (Im=pq piksel),

M(u,v):İki argüman u ve v arasındaki benzersizliği belirten bir metriktir.

D(s,t): Eşleme ölçütü veya bozulma olarak adlandırılır

f_n : n. çerçeve,

f_{n-1} : (n-1). çerçeve

Literatürde farkların karelerinin ortalaması (mean squared difference, MSD) (Jain vd. 1981), hataların karelerinin toplamı (sum of squared error, SSE) (Chan vd., 1990), mutlak farkların ortalaması (mean absolute difference, MAD) (Koga vd. 1981), mutlak farkların toplamı (sum of absolute difference, SAD) gibi birçok eşleme ölçütü vardır. Bunlar en çok kullanılan eşleme ölçütleridir.

Şekil 2.3’de üzerinde çalışılan arama öbeğiyle en iyi eşleşen öbeğin, önceki veya sonraki çerçevenin referans çerçeve seçilmesi durumundaki hareket vektörleri verilmiştir. Burada;

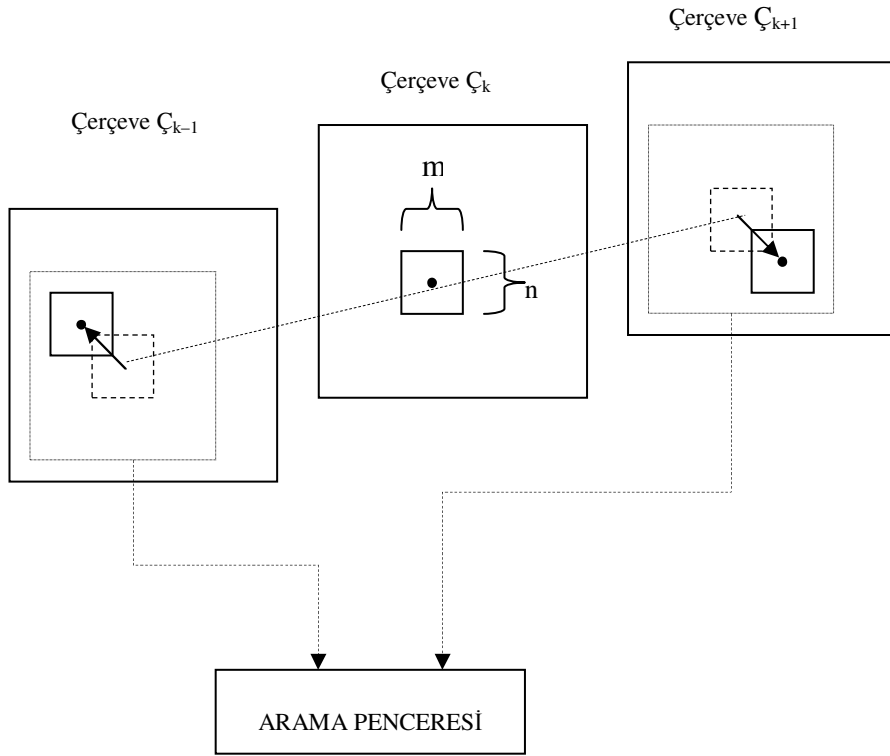
mn : Öbek büyüklüğü (piksel)

B_k : Güncel çerçevedeki (\mathcal{C}_k) $m \times n$ piksel çözünürlüğündeki öbek

B_{k-1} : Önceki çerçevedeki (\mathcal{C}_{k-1}) $m \times n$ piksel çözünürlüğündeki öbek

B_{k+1} : Sonraki çerçevedeki (\mathcal{C}_{k+1}) $m \times n$ piksel çözünürlüğündeki öbek

dx, dy : Arama konum hareket vektörleridir.



Şekil 2-3 Arama öbeğinin önceki veya sonraki çerçevede gösterimi

Güncel çerçevedeki B_k öbeği için önceki veya sonraki çerçevedeki (B_k öbeğini içine alacak şekilde seçilmiş) arama penceresindeki B_{k-1} öbeklerinden veya B_{k+1} öbeklerinden iyi eşleşen öbeği bulmak amaçtır. Benzersizliğinin yani hatanın, başka bir deyişle öbekler arasındaki farkın minimum olması istenir. Aşağıdaki 2.1.3.1, 2.1.3.2, 2.1.3.3 ve 2.1.3.4 başlıkları altında verilen denklemler ileri yönde hareket kestirimi için verilmiştir.

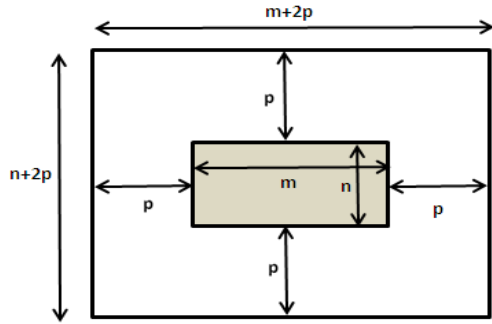
Geri yönde hareket kestirimi yapmak istersek, bu denklemlerdeki B_{k-1} yerine B_{k+1} yazılır.

2.1.3.1 Farkların karelerinin ortalaması (mean squared difference, MSD)

MSD, hataların karelerinin ortalaması (mean squared error, MSE) olarak da bilinir. En iyi sonuç veren eşleme ölçütlerinden biridir. Şu şekilde tanımlanır:

$$MSD(dx, dy) = \frac{1}{mn} \sum_{x=-m/2}^{m/2} \sum_{y=-n/2}^{n/2} (B_k(x, y) - B_{k-1}(x + dx, y + dy))^2 \quad (2.3)$$

Göz önüne alınan (dx, dy) değerleri, arama penceresinin tanımına göre belirlenir. Arama penceresinin parametresi p ile gösterilirse, dolayısıyla arama penceresi $(m+2p)(n+2p)$ piksellik bir alanı kapsar. Yani işlenen öbeğin, p aşağısını, p yukarısını, p sağını, p solunu kapsayan bir alandır. Bu arama penceresi $dx=(-p, +p)$ ve $dy=(-p, +p)$ ile gösterilir.



Şekil 2-4 Arama penceresi parametreleri

MSD hesabı, öbeğin her bir pikseli başına bir çarpma gerekmesine rağmen, MSD fonksiyonu, göz önüne alınan öbekler arasındaki Euclid uzaklığı olarak yorumlanabileceğinden, arama penceresindeki her bir öbek için MSD eşleme ölçütü çoğunlukla kullanılmaktadır.

2.1.3.2 Hataların karelerinin toplamı (sum of squared error, SSE)

SSE, farkların karelerinin toplamı (sum of squared difference, SSD) olarak da bilinir. Şu şekilde tanımlanır:

$$SSE(dx, dy) = \sum_{x=-m/2}^{m/2} \sum_{y=-n/2}^{n/2} (B_k(x, y) - B_{k-1}(x + dx, y + dy))^2 \quad (2.4)$$

MSD'den farkı $\frac{1}{mn}$ çarpanının olmamasıdır.

2.1.3.3 Mutlak farkların ortalaması (mean absolute difference, MAD)

MAD, mutlak hataların ortalaması (mean absolute error, MAE) olarak da bilinir. Şu şekilde tanımlanır:

$$MAD(dx, dy) = \frac{1}{mn} \sum_{x=-m/2}^{m/2} \sum_{y=-n/2}^{n/2} |B_k(x, y) - B_{k-1}(x + dx, y + dy)| \quad (2.5)$$

MAD gibi fonksiyonların avantajı basit olmalarıdır. Fonksiyonun donanımda doğrudan uygulaması vardır ve bu sebeple uygulamalarda daha çok tercih edilir (Musmann vd., 1985). Basitliğinin bir sonucu olarak, farkların karelerinin ortalaması (MSD) eşleme ölçütünün bazı avantajlarından yoksundur. MAD ölçütü, büyük farklardan ziyade küçük farkları vurgulamaktadır. Dolayısıyla, hızlı arama tekniklerinin daha fazla zamanda gerçekleşmesine sebep olur. Ayrıca, farkların karelerinin ortalaması (MSD) eşleme ölçütünden farklı sonuçlar verebilir.

MSD eşleme ölçütü MAD eşleme ölçütüne çok benzemekle birlikte önemli bir farkı vardır. Fark teriminin karesi, fonksiyonun daha doğru ve daha karmaşık olmasına sebep

olur. Kare almanın karmaşıklığı, çarpımların toplamlardan daha çok olmasının bir sonucudur.

2.1.3.4 Mutlak farkların toplamı (sum of absolute difference, SAD)

SAD, mutlak hataların toplamı (sum of absolute error, SAE) olarak da bilinir. Şu şekilde tanımlanır:

$$SAD(dx, dy) = \sum_{x=-m/2}^{m/2} \sum_{y=-n/2}^{n/2} |B_k(x, y) - B_{k-1}(x + dx, y + dy)| \quad (2.6)$$

MAD'dan farkı $\frac{1}{mn}$ çarpanının olmamasıdır.

2.1.4 Öbek eşleme arama teknikleri

İmge dizilerinde ardışık çerçeveler arası ilişkinin ortaya konması için çeşitli yöntemler geliştirilmiştir. Bu yöntemlerin en popüler ve en kullanışlı olanı öbek eşleme yöntemidir. Bu yöntemde imge öbeklere bölünür. Öbeklerin karmaşık hareketleri basit küçük öbeklere dönüştürülür ve çerçeveler arasındaki ilişki bu öbekler yardımıyla bulunur. Burada amaç k . çerçeveden seçilen bir öbeğe en iyi karşılık gelen $(k-1)$. çerçevedeki öbeği bulmaktır. Bunun için eşleme ölçütleri yardımıyla hareket yönünü belirleyen vektörler ortaya konur ve çerçeveler arasındaki değişimler bu vektörler yardımıyla izlenir.

Öbek eşleme tabanlı hareket kestirimlerinde birçok arama tekniği geliştirilmiştir. Bunlardan bazıları:

1. Tam Arama, TA (Full Search, FS)
2. İki Boyutlu Logaritmik Arama, İBLA (2-D Logarithmic Search, 2DLS)
3. Üç Adımda Arama, ÜAA (Three Step Search, TSS)
4. Yeni Üç Adımda Arama, YÜAA (New Three Step Search, NTSS)
5. Çapraz Arama, ÇA (Cross Search, CS)
6. Eşlenik Arama, EA (Conjugate Direction Search, CDS)
7. Dört Adımda Arama, DAA (Four Step Search, 4SS)
8. Baklava Desenli Arama, BDA (Diamond Search, DS)
9. Uyarlanırl Kök Örüntü Arama, UKÖA (Adaptive Root Pattern Search, ARPS)
10. Basit ve Etkili Üç Adımda Arama, BEÜAA (Simple and Efficient Search, SETSS)

2.1.4.1 Tam arama, TA (full search, FS)

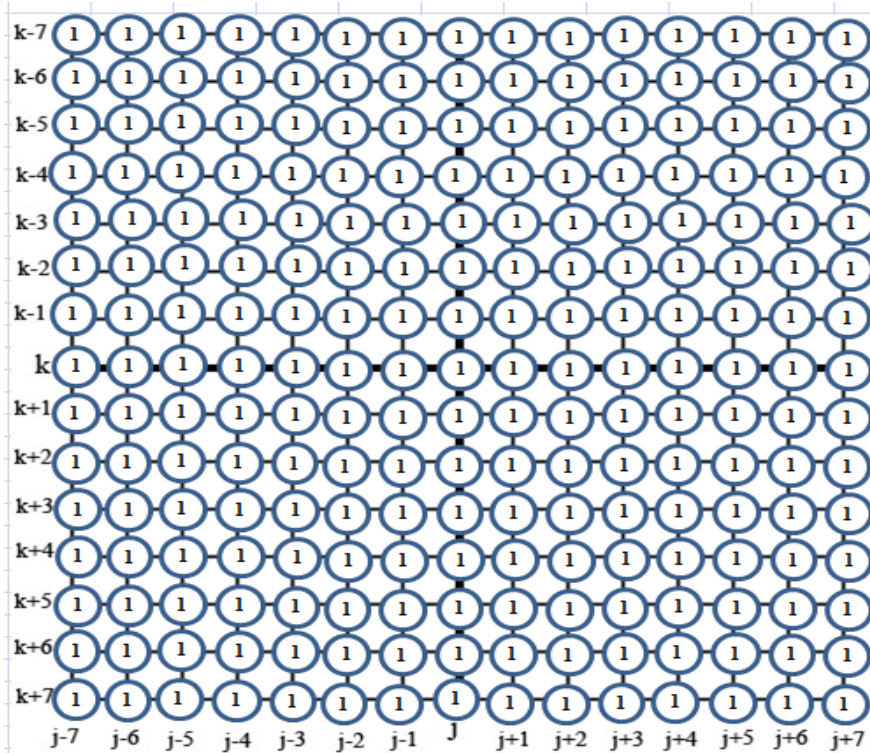
Tam arama, TA sade ve hesaplama yükü yüksek bir algoritmadır. Güncel çerçeveden seçilmiş bir öbek (çalışılan öbek), referans çerçevede arama penceresi içerisinde olası tüm öbeklerle eşleme ölçütlerinden biriyle (MAD, SSE, SAD gibi) karşılaştırılır. Eşleme ölçütü en düşük öbek bulunur. Bu öbek güncel çerçevedeki öbekle en iyi eşleşen öbeğdir. Referans çerçevede bulunan bu öbeğin konumu ile güncel çerçevedeki öbeğin konumu arasındaki fark hareket vektörünü verir. Bu şekilde güncel çerçevedeki her bir öbek için aynı işlemler tekrar edilerek tüm öbeklerin referans çerçevedeki en iyi karşılıkları algılanır.

Referans çerçevede arama penceresi içerisindeki ilinti öbeği, arama penceresi içerisindeki her bir durum için şekil 2.5'de görüldüğü gibi hareket eder. Bu nedenle en az hatayı veren algoritmadır. Fakat hızlı bir algoritma değildir. Bütün öbeklerde tarama yapıldığından toplam $(2p+1)(2p+1)$ durum için eşleme bulur. TA, p arama aralığı parametresidir ve $p=7$ için $15 \times 15 = 225$ tane eşleşen öbekten eşleme ölçütü değeri minimum olan bulur. Bu nedenle zaman gereksinimi fazla olmaktadır. Bu sorunu çözmek için yeni hızlı arama teknikleri geliştirilmiştir.

Diğer arama tekniklerine geçmeden önce bu algoritmalarda kullanılan parametrelerin tanımları aşağıda verilmiştir.

Arama aralığı parametresi (search space parameter) p : Merkez noktadan sağ, sol, üst veya alt kenara olan piksel uzaklığıdır. 15x15 piksel arama aralığı için p parametresi 7'dir.

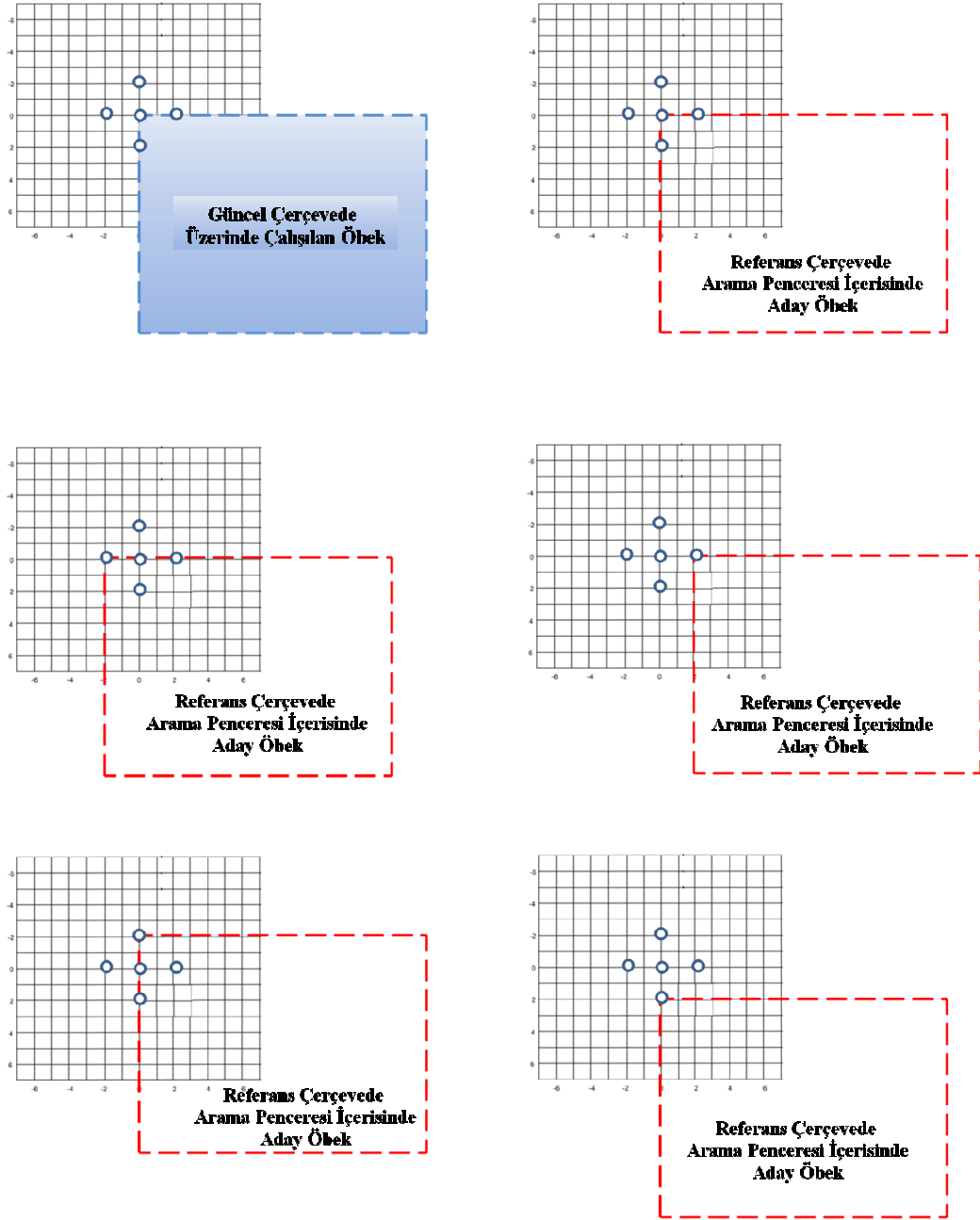
Adım büyüklüğü (step size) s : Arama algoritmalarına göre bu değer farklılık göstermektedir. Örneğin “Üç Adımda Arama” algoritmasında adım büyüklüğü ilk adımda $p/2$, ikinci adımda $p/4$, üçüncü adımda $p/8$ iken, “Dört Adımda Arama” algoritmasında adım büyüklüğü ilk üç adımda $p/4$, son adımda $p/8$ 'dir. Adım büyüklüğü hesaplanırken kesirli sonuçlar üst tamsayı değerine yuvarlanır. Örneğin $7/2=4$, $7/4=2$ ve $7/8=1$ olur.



Şekil 2-5 Tam arama

2.1.4.2 İki boyutlu logaritmik arama, İBLA (2-D logarithmic search, 2DLS)

Jain ve Jain (1981) tarafından bir boyutlu logaritmik arama (Knuth, 1973), algoritmasından yola çıkılarak iki boyutlu logaritmik arama, İBLA algoritması geliştirilmiştir (Jain vd. 1981).



Şekil 2-6 İBLA algoritmasının ilk adımı

Şekil 2.6'da İBLA algoritmasının ilk adımına ilişkin açık gösterim verilmiştir. Güncel çerçevede üzerinde çalışılan öbeğin sol üst köşesi merkez nokta olarak ele alınır. Bu merkez noktanın s adım büyüklüğü komşuluğunda (+) örüntü oluşturacak şekilde dört nokta seçilir.

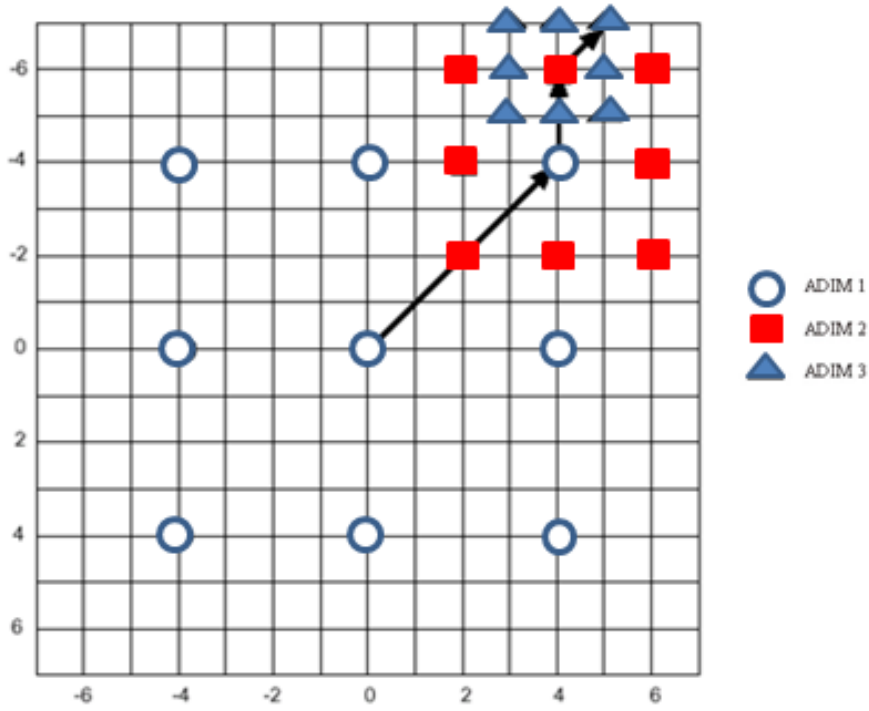
Referans çerçevedeki bu dört noktayı ve merkez noktasını sol üst pikseli olacak şekilde aday öbekler belirlenir. Aday öbekler ile üzerinde çalışılan çerçeve eşleme ölçütlerini kullanarak incelenir. Minimum hatayı veren öbekte sol üst köşesindeki konum birinci adım için yerel minimum noktayı veren konumdur. Bu nokta merkez ve (+) örüntülü s birim uzaklıktaki 4 komşu noktadan biridir.

Diğer adımlar ve bu çalışmada anlatılan diğer algoritmalar için “nokta (point)” kavramı; arama penceresi içerisindeki bir piksel değerini değil, referans çerçeve içerisinde üzerinde çalışılan öbek ile aynı en ve boyda olan en iyi eşleme için aday öbeğin sol üst köşe noktasını ifade eder. Eşleşme ölçütleri, sabit güncel çerçeve içerisindeki öbek ile referans çerçevede arama penceresi içerisindeki aday öbekler arasında yapılır ve bu öbeklerin sol üst köşe noktaları “nokta” ile ifade edilir.

İBLA'da her adımda (+) biçimli örüntü kullanılır. Başlangıç adım büyüklüğü $p/4$ 'dür ($p=7$ için bu değer 2 olur). Her adımda bulunan en iyi eşleşen öbeğin konumu, arama örüntüsünün merkezinde veya arama penceresinin sınır noktasında olursa adım büyüklüğü yarıya düşer. Diğer durumda adım büyüklüğü değişmez.

Adım büyüklüğü 1'e düştüğünde, bir önceki adımın en iyi eşleşen noktasına komşu 8 nokta için en iyi eşleşen bulunur. Şekil 2.7'de iki farklı arama yolu (search path) gösterilmiştir. Üstteki arama yolu $(5+3+3+8)$ 19, alttaki arama yolu $(5+3+2+3+2+8)$ 23 kontrol noktasını temel alarak aday öbek sayısını 19'a ve 25'e düşürür.

Son adımda bulunan nokta ile şekil 2.7'de (0,0) ile gösterilen merkez nokta arasındaki fark hareket vektörünü verir.



Şekil 2-8 Üç adımda arama

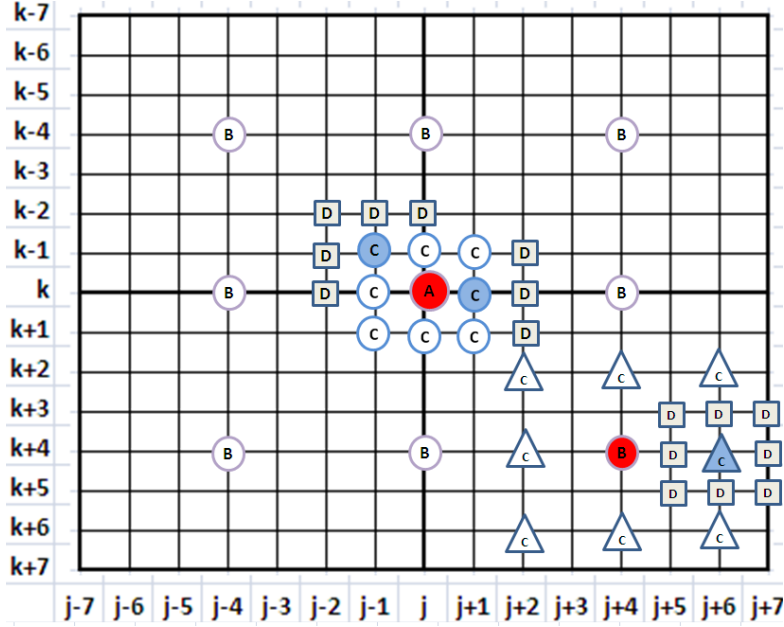
2.1.4.4 Yeni üç adımda arama, YÜAA (new three step search, NTSS)

Reoxiang, Bing ve Liou (1994) tarafından geliştirilmiştir. YÜAA, merkez odaklı bir yapı kullandığından ve ek hesaplama maliyetlerini yarıya indirme koşullarına sahip olduğundan ÜAA sonuçlarını iyileştirir. Bu algoritma, MPEG-1 ve H.261 gibi ilk kullanılan standartların uygulanmasında en çok kabul edilen ve sık kullanılan hızlı algoritmalarından biridir.

ÜAA algoritması önceden düzenli bir biçimde belirlenmiş eşleme pozisyonlarını incelediği için küçük hareketleri tespit etmek konusunda çok başarılı değildir. Bu algoritma incelemeye merkez pozisyonundan başlar.

Eğer minimum eşleme ölçütü değeri merkez noktaysa(A); sıfır hareket söz konusudur ve hareket vektörü (0,0) olarak atanır. Hareket vektörü 17 pozisyonda

bulunur (ÜAA'daki ilk adımdaki 9 nokta ve merkez A noktasına $s=1$ adım büyüklüğü uzaklığında komşu 8 nokta).



Şekil 2-9 Yeni üç adımda arama

Eğer minimum eşleme ölçütü değeri B noktalarından biriye; ÜAA'nın adımları uygulanır. Hareket vektörü 33 noktayı temel olarak bulunur (ÜAA'daki 25 nokta ve merkez A noktasına $s=1$ adım büyüklüğü uzaklığında komşu 8 nokta).

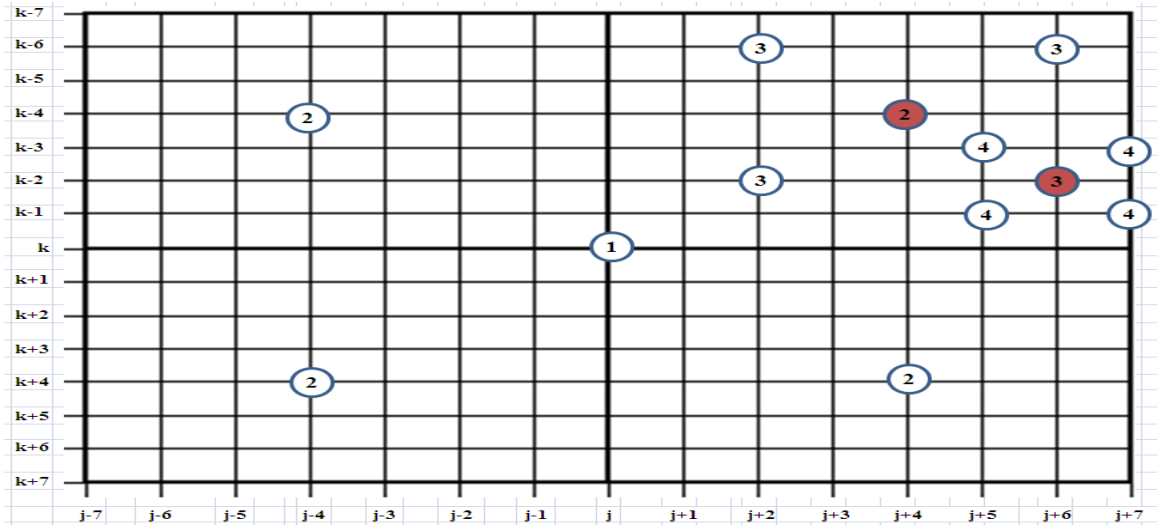
Eğer minimum eşleme ölçütü değeri kırmızı A noktasını saran 8 C noktasından biriye; bu nokta merkez nokta olarak atanır. Bitişik pozisyonların sayısı, merkezin ilk belirlenen pozisyonun konumuna bağlı olarak 5 yada 3 olmaktadır. C noktasının indislerinden biri k veya j ise (C ortada ise) bitişik 3 nokta, diğer durumda (C köşede ise) bitişik 5 nokta için en küçük MSE bulunur. Hareket vektörü 20 veya 22 pozisyonda bulunur.

En iyi durumda 17 pozisyon en kötü durumda ise 33 pozisyon incelenir. Ortalama olarak bakıldığında bu yeni algoritma Üç Adımda Arama tekniğine göre daha hızlıdır ve küçük hareketleri de tespit edebilme kabiliyetine sahiptir.

2.1.4.5 Çapraz arama, ÇA (cross search, CS)

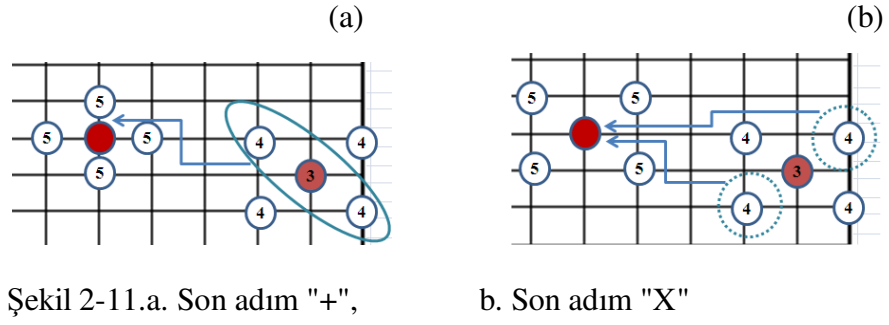
Po ve Ma (1996) tarafından geliştirilmiştir. ÜAA algoritması mantığına benzer şekilde çalışır. Ancak burada merkez nokta etrafında sıralanan yerel noktalar 4'tür. Son adıma kadar sadece merkez noktanın çaprazında sıralanır. Başlangıç adım büyüklüğü $p/2$ 'dir ($p=7$ için bu değer 4 olur). Her adımda bulunan en iyi eşleşen nokta sonraki adımın merkez noktası olur ve her adımda s adım büyüklüğü yarıya düşer.

Şekil 2.10'da görüldüğü gibi (j, k) merkez noktasının çaprazında $s=4$ uzaklığında sıralanan 4 nokta ve merkez noktadan eşleme ölçüt değeri minimum olan nokta bir sonraki adım için yeni merkez noktası ve $s=2$ olarak belirlenir. Bu örnekte $s=4$ iken kırmızı yuvarlak içersindeki 2, $s=2$ iken kırmızı yuvarlak içersindeki 3 noktası yeni merkez noktası olur. $s=1$ olduğunda da merkez noktanın çaprazlarındaki 4 nokta ve merkez nokta için eşleme ölçüt değeri minimum olan nokta bulunur.



Şekil 2-10 Çapraz arama

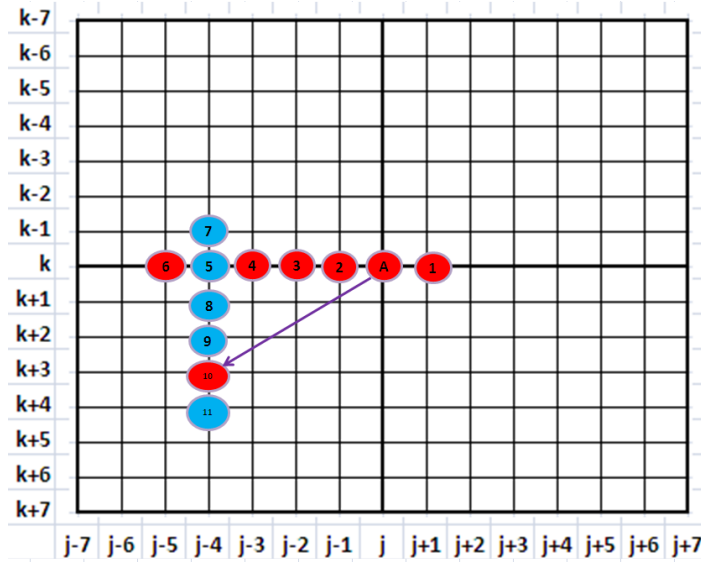
Son adım için ise, $s=1$ için bulunan yeni merkez noktanın konumuna göre şekil-2.11.a'daki gibi "+" veya şekil-2.11.b'deki gibi "X" örüntülü son 4 nokta eklenir ve bu 5 noktaya göre eşleme ölçüt değeri minimum olan noktaya doğru hareket vektörü tespit edilmiş olur.



2.1.4.6 Eşlenik arama, EA (conjugate direction search, CDS)

Srinivasan ve Rao (1985) tarafından geliştirilmiştir. Bu arama yöntemi dikey ve yatay yönlerde hareketin olduğu imgelerde en iyi sonucu verir. Daha karmaşık hareketlerin olduğu imgelerde kullanılması kötü bir sonuca yol açar. 2 kademeden oluşur:

- i) Aynı y ekseninde yatayda arama
- ii) Aynı x ekseninde düşeyde arama



Adım 1: İlk çerçeveden seçilen öbeğin 2. çerçevedeki karşılığı aranırken, ilk çerçeveden seçilen öbeğin bulunduğu düşey eksen 2. çerçevede sabit tutularak yatayda arama yapılır ve yataydaki her bir öbek için eşleme ölçütü değerleri hesaplanır.

Şekil 2.12'deki A noktasının 1 birim sol ve sağ tarafındaki noktalar için eşleme ölçütü değerlerinden minimum olan bulunur. Bu nokta $(j-1,k)$ ise sola 1 birim, $(j+1,k)$ ise sağa 1 birim yeni nokta belirler. Bu işlem belirlenen yeni nokta için eşleme ölçütü değerinin artması ile son bulur. Yukarıdaki örnekte 5 noktası eşleme ölçütü değeri minimum noktadır. Şekil 2.12'de eşleme ölçüt değerleri; $1 > A > 2 > 3 > 4 > 5 < 6 > 5$

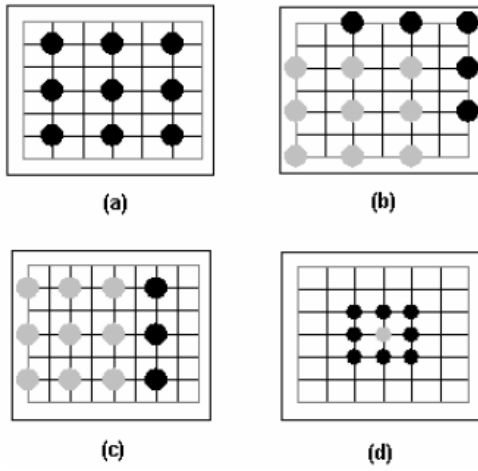
Adım 2: Yataydaki en küçük eşleme ölçütü değerini veren öbek bulunur. Şekil 2.12'de 5 noktasıdır. Bu kez öbeğin bulunduğu yatay eksen sabit tutularak düşeyde arama yapılır. Düşey ekseninde en düşük eşleme ölçütü değerine sahip öbek, aradığımız en iyi eş öbeğdir.

Şekil 2.12'deki 5 noktasının 1 birim üst ve aşağı tarafındaki noktalar için eşleme ölçütü değerlerinden minimum olan bulunur. Bu nokta $(j-4,k+1)$ ise üstte 1 birim, $(j-4,k-1)$ ise altta 1 birim yeni nokta seçilir. Bu işlem belirlenen yeni nokta için eşleme ölçütü değerinin artması ile son bulur. Yukarıdaki örnekte 10 noktası eşleme ölçütü değeri minimum noktadır. Şekil 2.12'de eşleme ölçüt değerleri; $7 > 5 > 8 > 9 > 10 < 11$

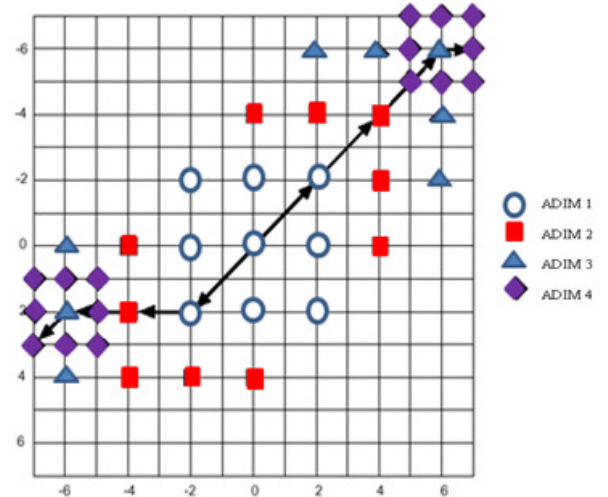
2.1.4.7 Dört adımda arama, DAA (four step search, 4SS)

L.M. Po ve W.C. Ma (1996) tarafından geliştirilmiştir. Şekil 2.13'de birinci adım için a , ikinci ve üçüncü adımlar için b veya c , dördüncü adım için d örüntüsü kullanılır. Arama dört adımda tamamlanır. İlk üç adımda, s adım büyüklüğü $p/4$ 'dür ($p=7$ için bu değer 2 olur). Son adımda s adım büyüklüğü $p/8$ 'dir ($p=7$ için bu değer 1 olur).

Şekil 2.14'de iki farklı arama yolu (search path) için DAA algoritması gösterilmiştir. Üstteki arama yolu $(9+5+5+8)$ 27, alttaki arama yolu $(9+5+3+8)$ 25 kontrol noktasını temel alarak aday öbek sayısını 27'ye ve 25'e düşürür.

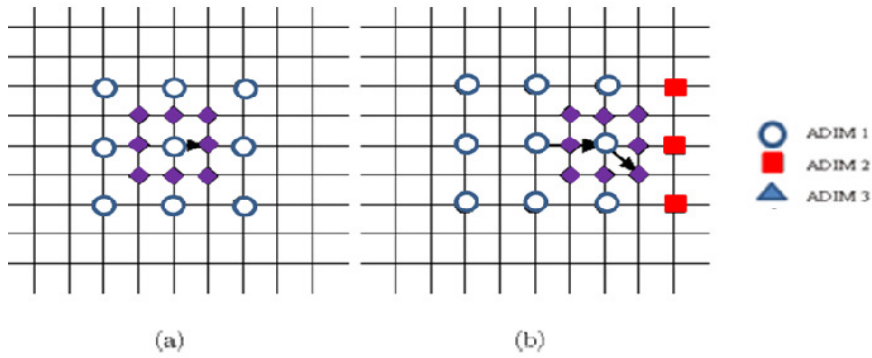


Şekil 2-13 DAA örüntüsü



Şekil 2-14 Dört adımda arama

Birinci aramada en iyi eşleşen nokta merkezde veya ikinci aramada en iyi eşleşen nokta başlangıç 9 noktadan biri ise şekil 2.13'deki d örüntüsü kullanır. Merkez noktasına yakın küçük hareketler için DAA, tıpkı YÜAA algoritmasında olduğu gibi yarıyol-dur tekniği (halfway-stop technique) ile 2 veya 3 adımda algoritma Şekil 2.15'deki gibi sonlanır.



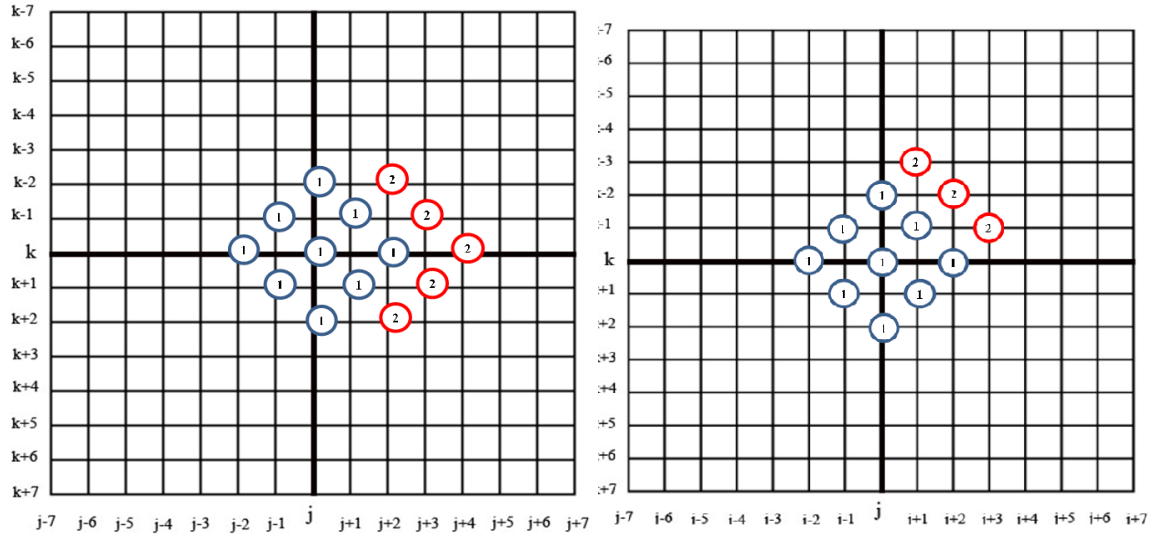
Şekil 2-15 Dört adımda arama –merkezdeki hareketler

2.1.4.8 Baklava desenli arama, BDA (diamond search, DS)

Zhu ve Ma (2000) tarafından geliştirilmiştir. ÜAA ve DAA algoritmalarına göre daha iyi bir performans gösterir. Algoritma mantığı DAA'ya çok benzemektedir.

Adım 1: $s=p/2=4$ 'dür. 8 nokta merkez nokta etrafında 2 piksel uzaklıkta baklava (diamond) deseninde sıralanır. Minimum eşleme ölçütü değeri merkez nokta ise **adım 4'e** geçer; değilse **adım 2'den** devam eder.

Adım 2: Algoritma minimum eşleme ölçütü değerinin konumuna göre şekillenir.



(a) Eşleme ölçütü köşede en küçük (b) Eşleme ölçütü ortada en küçük
Şekil 2-16 Baklava desenli arama

a) Eğer minimum eşleme ölçütü değeri köşe noktası ise (örnek $(k, j+2)$ noktası) bu noktanın komşuluğundaki 5 nokta Şekil 2.16.a'daki gibi seçilir.

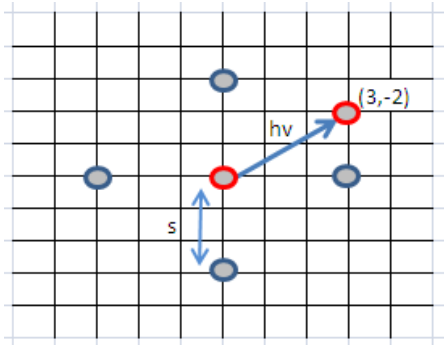
b) Eğer minimum eşleme ölçütü değeri orta noktalarından biri ise (örnek $(k-1, j+1)$ noktası) bu noktanın komşuluğundaki 3 nokta şekil 2.16.b'deki gibi seçilir. Bu seçilen yeni noktaların a veya b'ye göre (3 veya 5 nokta) eşleme ölçütü değerlerine bakılır ve eşleme ölçütü değeri minimum nokta merkez nokta ise **adım 4'e** geçilir; değilse **adım 3'ten** devam edilir.

Adım 3: Merkez nokta bulunana kadar **adım 2** tekrar edilir. Merkez nokta bulununca **adım 4'e** geçilir.

Adım 4: Adım büyüklüğü yarıya düşer ($s=1$) ve 4 nokta merkez noktanın düşeyinde ve yatayında yer alır. Böylece en iyi eş öbek bulunmuş olur.

2.1.4.9 Uyarlanırlık kök örüntü arama, UKÖA (adaptive root pattern search, ARPS)

Nie ve Ma (2002) tarafından geliştirilmiştir. Uyarlanırlık Kök Örüntü Arama tekniği bir bloğun çevre öbeklerdeki hareket yönünde harekete sahip olma ihtimalinin yüksek olmasını dikkate alan bir algoritmadır. Bu algoritma en yakın sol bloğun hareket vektörünü kendi vektörü olarak kabul ederek inceleme çerçevesini buna göre belirler.



Şekil 2-17 Uyarlanırlık kök örüntüsü

Şekil 2.17’de bir örnek gösterilmiştir. Tahmin edilen hareket vektörü (3,-2) noktasını işaret eder. Tahmin edilen hareket vektörünün gösterdiği noktayı kontrol etmesinin yanı sıra, şekil 2.17’de görüldüğü gibi, maksimum $S = \text{Max}(|X|, |Y|)$ olmak üzere kök örüntü dağılım noktalarını da kontrol eder. X ve Y tahmin edilen vektörün koordinatları olmak üzere algoritma bu vektörün işaret ettiği pozisyona ek olarak adım büyüklüğü ile belirlenen pozisyonları da inceler. Araştırmayı doğrudan yüksek olasılıkla iyi bir eşleşme bloğu bulma ihtimali olan bir bölgeye yöneltir. Bu yaklaşım algoritmanın ilk adımında gerçekleşir. Bu sayede ilk adımda iyi eşleme ölçütü değerine sahip olan pozisyonlar incelenmiş olur. Bu adımdan sonra en düşük eşleme ölçütü değerine sahip pozisyon diğer adımların merkezi olarak belirlenir ve işlemler Küçük Baklava Desenli Arama Örüntüsü ($S=\text{Max} |X|, |Y|$) çerçevesinde devam eder. İşlem adımları en iyi

eşleme ölçütü değerine sahip pozisyon Küçük Baklava Desenli Arama Örüntüsü çerçevesinin merkezine geldiği zaman sonlanır.

Algoritmada küçük bir iyileştirme de Sıfır Hareket Yargısının kontrol edilmesi ile yapılabilir. Buna göre, eğer en düşük değere sahip pozisyon diğer adımların merkezi ise araştırma yarıda bırakılır.

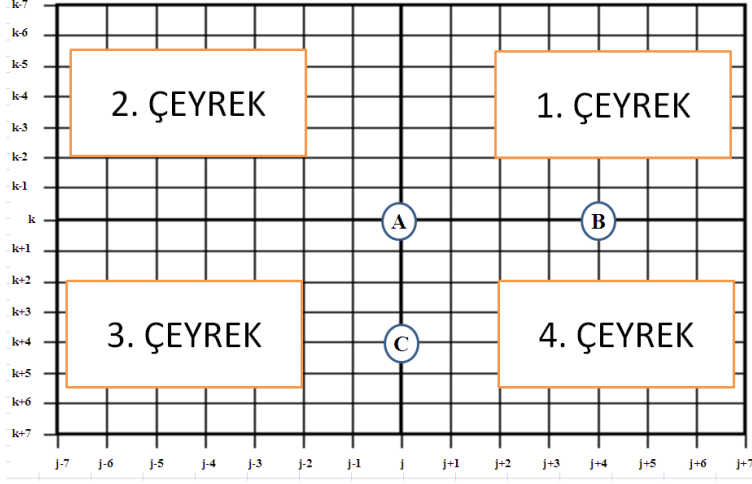
Bu algoritmanın en dikkat çeken avantajı sıfır-hareket durumunda (0, 0) Büyük Baklava Desenli Arama pozisyonlarını inceleyerek zaman kaybetmemesidir (Bunun yerine doğrudan Küçük Baklava Desenli Arama pozisyonları incelenir). Aynı şekilde Uyarlanı Kök Örüntü Arama ile tahmin edilen hareket vektörü çok uzakta ise direk olarak bu noktaya atlanarak Küçük Baklava Desenli Aramaları incelendiği için Baklava Desenli Aramada gerçekleştirilen Büyük Baklava Desenli Arama incelemeleri elimine edilmiş olur. Bu atlama işlemlerinden dolayı dikkat edilmesi gereken bazı durumlar şunlardır:

- Algoritmada daha önce incelenen pozisyonların tekrar incelenmesi ihtimali söz konusudur.
- Tahmin edilen hareket vektörü diğer incelenen pozisyonlardan biri ile çakışırsa bu noktada iki kez hesaplama yapma ihtimali söz konusudur. Çerçevenin ilk sütunundaki makro öbekler için, kök örüntüsü 2 pikselde sabitlenir.

2.1.4.10 Basit ve etkili üç adımda arama, BEÜAA (simple and efficient three step search, SETSS)

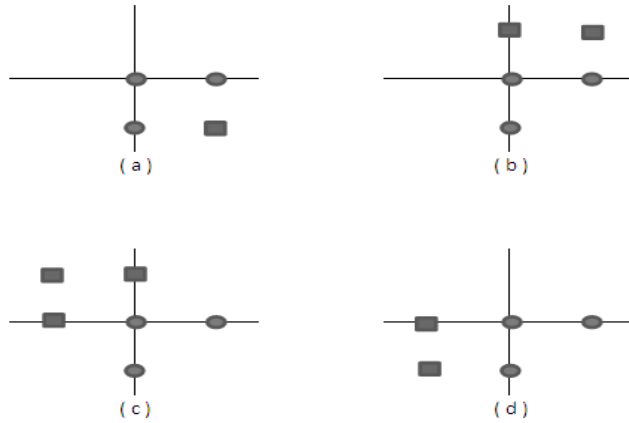
Jianhua ve Ming tarafından 1997'de geliştirilmiştir. Basit ve Etkili Üç Adımda Arama ÜAA algoritmasının hareket vektörlerinin unimodal olması özelliğinden yararlanır. Bu özellik dolayısıyla karşı yönlerde iki tane minimum eşleme ölçütü değeri olamayacağını göz önünde bulundurularak ÜAA algoritması hızlandırılmaya çalışılmıştır. Algoritma ÜAA'da olduğu gibi 3 adımda incelemeleri tamamlar, ama yeni olan şey her bir aşamanın iki yeni safhasının olmasıdır.

BEÜAA yaklaşımında öncelikle araştırma alanı 4 parçaya ayrılır ve bu parçalardan 3'ünde araştırma işlemi gerçekleşir. Araştırma alanının parçalanmış hali ve incelenen üç nokta olan A, B, C noktaları şekil 2.18'de gösterilmiştir.



Şekil 2-18 Basit ve etkili üç adımda arama çeyrekler

Bu şekilde A pozisyonu merkezde yer alır. B ve C pozisyonları ortogonal yönde A pozisyonuna 4 birim uzaklıktadır. BEÜAA algoritmasında her adımdan sonra ikinci bir aşamaya geçilir. Bu aşamada bir kaç tane ek inceleme pozisyonu eklenir (Şekil 2.19). Bu işlem yapılırken MAD hesaplamalarının sonucu dikkate alınır. A, B, C noktalarından herhangi bir noktayı X ile gösterirsek $MAD(X)$, X pozisyonu için Mutlak Farkların Ortalaması (MAD) değeridir.



Şekil 2-19 Basit ve etkili üç adımda arama MAD

Eğer $MAD(A) \geq MAD(B)$ ve $MAD(A) \geq MAD(C)$ ise 1. çeyrek seçilir. MAD değerlerinin küçük olduğu B ve C noktalarının keşimi olan 4. çeyreğe, hareket kestirim incelemesinin artırılması için şekil 2.19.a'daki gibi bir inceleme pozisyonu eklenir.

Eğer $MAD(C) > MAD(A) \geq MAD(B)$ ise 2. çeyrek seçilir. A noktası ile B noktası etrafında şekil 2.19.b'deki gibi iki inceleme pozisyonu eklenir. C noktası için MAD değeri büyük olduğundan bu iki inceleme pozisyonu C doğrultusunun tersi yönde alınır.

Eğer $MAD(A) < MAD(B)$ ve $MAD(A) < MAD(C)$ ise 3. çeyrek seçilir. Şekil 2.19.c'deki gibi sadece A pozisyonu etrafındaki 3 pozisyon incelemeye eklenir. Çünkü MAD(A) en küçüktür ve hareket kestirim incelemesinin artırılması için eklenecek bölge B ve C noktaları ile kesim noktası olmayan 2. çeyrek bölgesidir.

Eğer $MAD(B) > MAD(A) \geq MAD(C)$ ise 4. çeyrek seçilir. A noktası ile B noktası etrafında şekil 2.19.d'deki gibi iki inceleme pozisyonu eklenir. B noktası için MAD değeri büyük olduğundan bu iki inceleme pozisyonu B doğrultusunun tersi yönde alınır.

İkinci aşamada ek pozisyonlar belirlenir. Bu pozisyonların MAD hesaplamaları yapılır. Tüm pozisyonlar içinde hesaplanan en küçük MAD değerine sahip pozisyon diğer adımın merkezi olur. Bundan sonraki adımlar ÜAA algoritmasındaki gibi belirlenir, ancak farklı olarak her adımda ikinci aşama tekrarlanır. En son adımda elde edilen en küçük MAD değerine sahip bloğun yönü hareket vektörünü belirler.

2.1.5 Tepe sinyal gürültü oranı, TSGO (Peak Signal To Noise Ratio, PSNR)

Video dizilerinde yapılan çalışmaların başarısı, etkinliği, sonuçları tıpkı hayatta yaptığımız çalışmalarda olduğu gibi bir değerlendirmeyle anlamlandırılır. Bu değerlendirme öznel yani öznenen özneye değişebilen olabileceği gibi nesnel de olabilir. Öznel değerlendirmenin sonuçlarını çeşitli istatistikî çalışmalar ile

anlamlandırsak bile çalışma sonuçlarının değerlendirilmesinde karşılaştırma sorununun çıkması, kişiden kişiye değişmesi nedeni ile kişiden bağımsız nesnel değerlendirmeler uygulamada karşımıza çıkar.

Tepe Sinyal Gürültü Oranı, TSGO (Peak Signal To Noise Ratio, PSNR) da hareket kestirimi ile ilgili çalışmalardaki nesnel değerlendirmelerden biridir. TSGO logaritmik ölçekte ölçülmekle beraber, güncel çerçeve ile dengelenmiş çerçevenin arasındaki Hataların Karelerinin Ortalaması (HKO) ile ilintilidir. 2.3'deki denklemden farkı hesaplamada referans çerçevenin yerine dengelenmiş çerçevenin kullanılmasıdır. Görüntü dizisindeki yüksek TSGO genellikle yüksek niteliği, düşük TSGO düşük niteliği gösterir.

$$HKO(dx, dy) = \frac{1}{mn} \sum_{x=-m/2}^{m/2} \sum_{y=-n/2}^{n/2} (B_k(x, y) - D_k(x + dx, y + dy))^2 \quad (2.7)$$

$$TSGO = 10 \log_{10} (\text{maksP}^2 / HKO) \quad (2.8)$$

maksP: Çerçevede kullanılan maksimum parlaklık değeri.

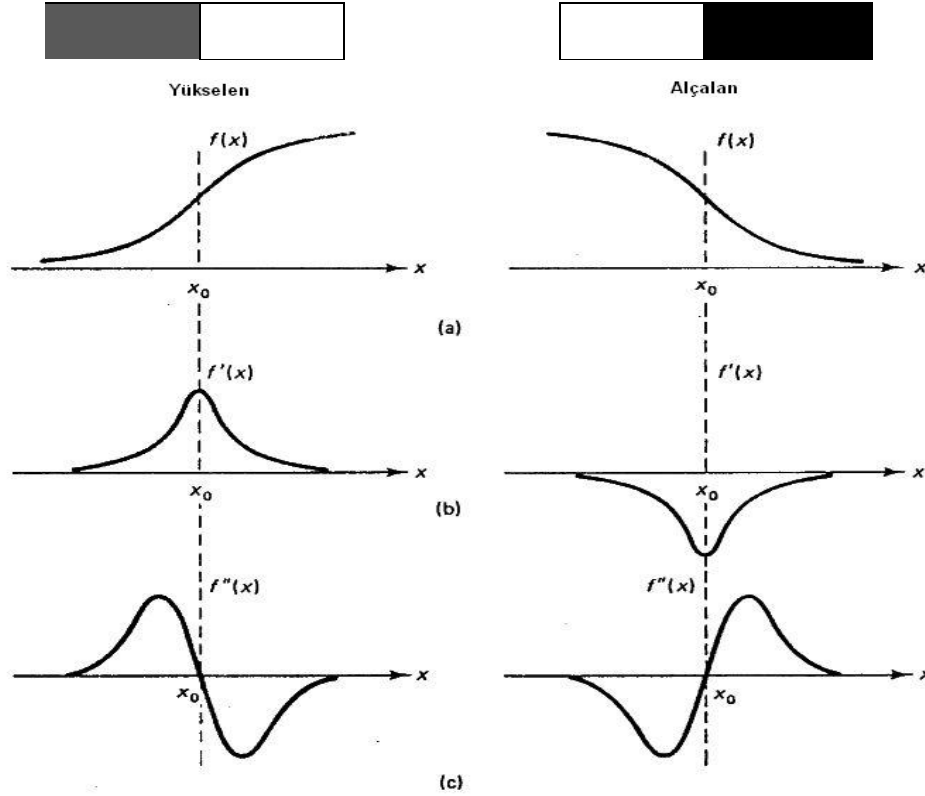
2.2 Kenar Algılaması (Edge Detection)

“Kenar”, görüntünün bir fiziksel özelliğinden, önemli bir değişikliğin ortaya çıktığı bir sınır olarak tanımlanır. Buradaki fiziksel özellikler; yüzey yansıması, ışımaya veya görünen yüzeylerin izleyiciden uzaklığı v.b.dir (Lim 1990).

Yukarıdaki “önemli değişiklik” uygulamadan uygulamaya farklıdır. Bir nesne tanıma sisteminde nesne sınırları tanıma için yeterli olabilir. Nesnedeki detaylar gözardı edilebilir.

2.2.1 Gradyana dayalı yöntemler

$f(x)$, bir satırdaki parlaklık değişimini gösteren fonksiyon olmak üzere (Lim 1990);

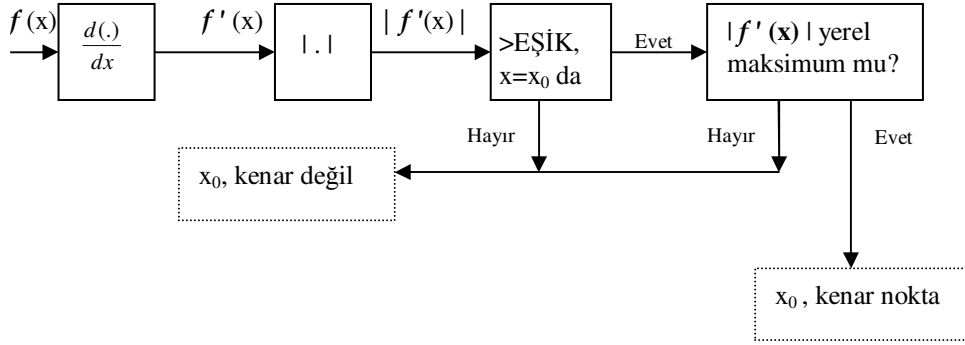


Şekil 2-20 Parlaklık değişim fonksiyonu

Şekil 2.20'deki parlaklık değişim fonksiyonundan görüldüğü üzere; $f''(x_0)=0$ 'dır. x_0 kenar noktasıdır.

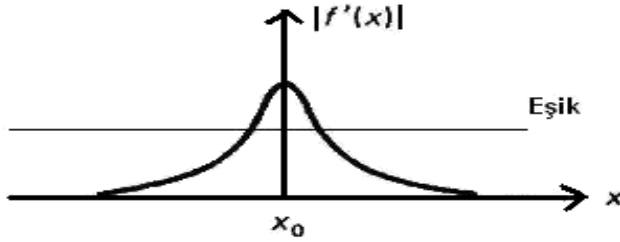
$f'(x)$ kenar noktası x_0 'ın belirlenmesinden başka, kenar yönünü ve kenar keskinliğinin belirlenmesinde kullanılır. Örneğin $|f'(x)|$ büyükse kenar keskin, $f'(x)$ pozitif ise yükselen vb. gibi.

Şekil 2.21'de bir boyutlu kenar algılamasının blok çizimi gösterilmiştir.



Şekil 2-21 Bir boyutlu (1-D) kenar algılaması blok çizimi

Eğer “ $|f'(x)| > \text{EŞİK}$ değer, $x=x_0$ ‘da “ doğru ise, sonuç bir nokta değil bir çizgidir. “ $|f'(x)|$ yerel maksimum“ ise x_0 kenar noktasıdır. Bu durum aşağıdaki şekil 2.22’de gösterilmiştir.



Şekil 2-22 Bir boyutlu (1-D) kenar algılaması eşik seçimi

Eşik değerinin seçimi uygulamaya bağlıdır.

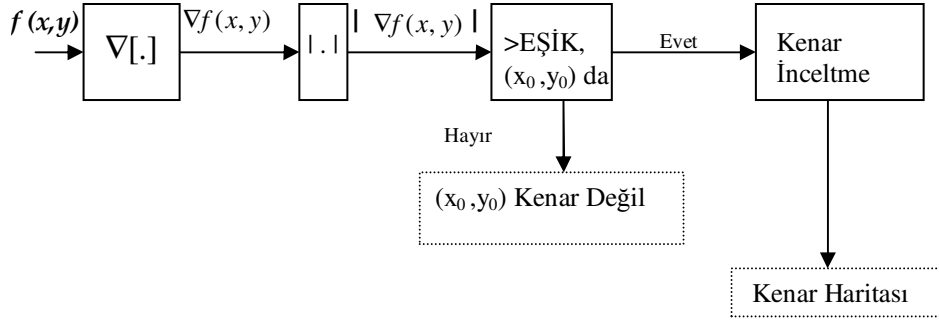
$f'(x)$ 'in iki boyutlu (2-D) bir fonksiyona uygulaması gradyan $\nabla f(x,y)$ 'dir.

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} i_x + \frac{\partial f(x,y)}{\partial y} i_y \quad (2.9)$$

i_x , x yönündeki birim vektör.

i_y , y yönündeki birim vektör.

Önceki şekil 2.21’deki blok çizimin $\nabla f(x,y)$ ‘ye dayalı biçimde genelleştirilmesi aşağıdadır.



Şekil 2-23 İki boyutlu (2-D) gradyana dayalı kenar algılaması blok şeması

Eşik değerden büyük olan noktalar “kenar adayı”dır. Tüm kenar adayları kenar kabul edilse, kenar haritasında çizgiler yerine şeritler ortaya çıkar. Basit bir kenar inceltme algoritmasında $|\nabla f(x,y)|$ 'nin en az bir yönden yerel maksimum olup olmadığına bakılır. $|\nabla f(x,y)|$ 'ye dayanan kenar algılaması sistemlerine “yönsüz kenar algılaması sistemleri” denir. Çünkü bu fonksiyon herhangi bir yönü vurgulamaz (Lim 1990).

“Yönlü kenar algılayıcıları” belirli bir yönü vurgular. Örneğin $|\nabla f(x,y)|$ yerine $\left| \frac{\partial f(x,y)}{\partial x} \right|$

kullanılırsa sistem düşey kenarları, $\left| \frac{\partial f(x,y)}{\partial y} \right|$ kullanılırsa sistem yatay kenarları algılar.

İki boyutlu (2-D) $f(n_1,n_2)$ ayrık uzay sinyali için $\left| \frac{\partial f(x,y)}{\partial x} \right|$ ve $\left| \frac{\partial f(x,y)}{\partial y} \right|$ kısmı türevleri, (2.10)'daki denklemlerdeki bazı farklar biçiminde gösterilebilir.

$$\left| \frac{\partial f(x,y)}{\partial x} \right| \rightarrow [f(n_1,n_2) - f(n_1-1,n_2)] / T \quad (\text{eğim}) \quad (2.10)$$

$$\text{veya } [f(n_1+1,n_2) - f(n_1,n_2)] / T$$

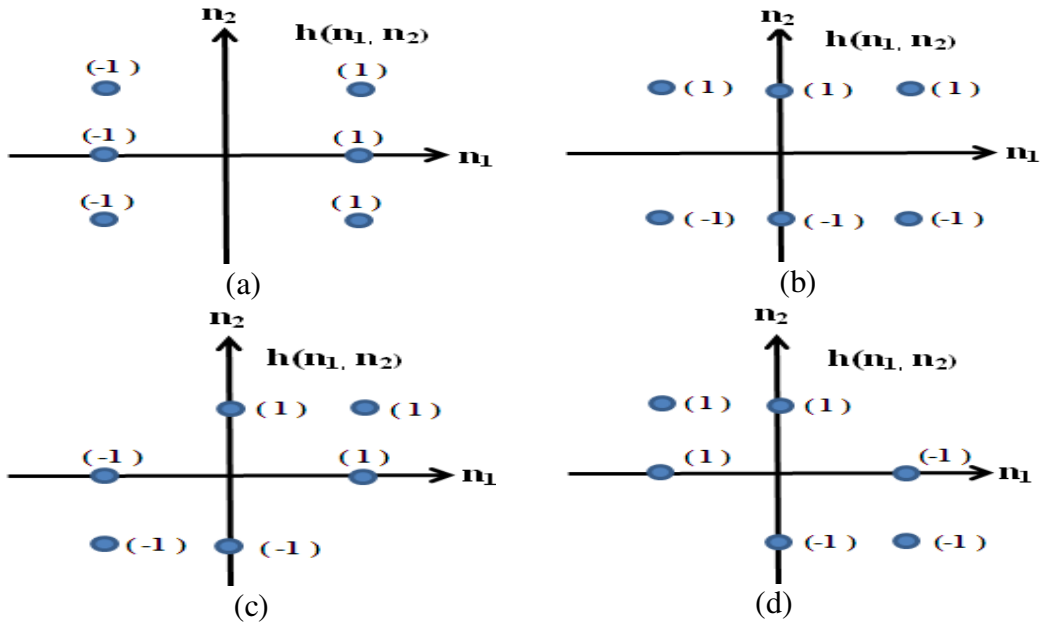
$$\text{veya } [f(n_1+1,n_2) - f(n_1-1,n_2)] / 2T$$

Hesaplanan türevler eşik değerleriyle karşılaştırılacağından $1/T$ veya $1/2T$ çarpanları atılabilir. Daha iyileştirilmiş bir kestirim;

$$\begin{aligned}
\left| \frac{\partial f(x, y)}{\partial x} \right| &\rightarrow [f(n_1+1, n_2+1) - f(n_1-1, n_2+1)] + [f(n_1+1, n_2) - f(n_1-1, n_2)] \\
&+ [f(n_1+1, n_2-1) - f(n_1-1, n_2-1)] \\
&\text{veya } [f(n_1+1, n_2+1) - f(n_1-1, n_2-1)] + 2[f(n_1+1, n_2) - f(n_1-1, n_2)] \\
&+ [f(n_1+1, n_2-1) - f(n_1-1, n_2-1)]
\end{aligned} \tag{2.11}$$

(2.10) ve (2.11) denklemlerindeki çıkarma işlemleri $f(n_1, n_2)$ ile süzgeç dürtü tepkisi $h(n_1, n_2)$ 'nin katlanması olarak düşünülebilir.

Yönlü kenar algılayıcıda kullanılan dürtü tepkelerine örnek şekil 2.18'de verilmektedir.



Şekil 2-24 Yönlü kenar algılayıcılarında kullanılan dürtü tepkeleri

Şekil 2.24.a) $\frac{\partial f(x, y)}{\partial x}$ ile düşey kenarlar kestirilmektedir.

Şekil 2.24.b) $\frac{\partial f(x, y)}{\partial y}$ ile yatay kenarlar kestirilmektedir.

Şekil 2.24.c) Diyagonal (/) kenarlar kestirilmektedir.

Şekil 2.24.d) Diyagonal (\) kenarlar kestirilmektedir.

Yönsüz kenar algılayıcılar $|\nabla f(x, y)|$ 'nin ayrık yaklaşıklığı ile elde edilir.

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad (2.12)$$

(2.12) denkleminin ayrık yaklaşığı,

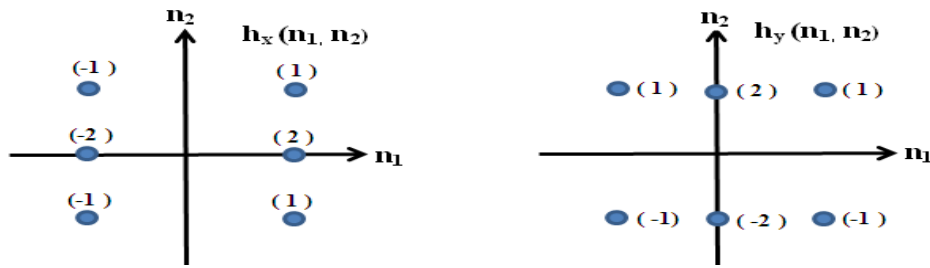
$$|\nabla f(x, y)| \rightarrow \sqrt{f_x^2(n_1, n_2) + f_y^2(n_1, n_2)} \quad (2.13)$$

Denklem (2.13)'de;

$$f_x(n_1, n_2) = f(n_1, n_2) * h_x(n_1, n_2) \quad \text{ve}$$

$$f_y(n_1, n_2) = f(n_1, n_2) * h_y(n_1, n_2) \quad \text{dir.}$$

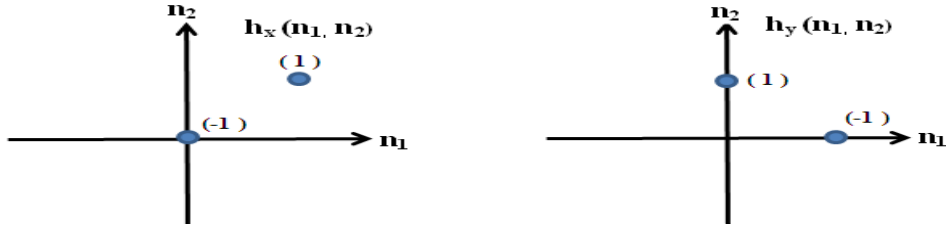
“Sobel”in geliştirdiği yöntem 2.13'deki denkleme dayanmaktadır ve şekil 2.25'deki $h_x(n_1, n_2)$ ve $h_y(n_1, n_2)$ dürtü tepkelerini kullanır (Lim 1990).



Şekil 2-25 Sobel kenar algılamasının dürtü tepkeleri

“Robert”in geliştirdiği yöntem 2.13'deki denkleme dayanmaktadır ve şekil 2.20'deki $h_x(n_1, n_2)$ ve $h_y(n_1, n_2)$ dürtü tepkelerini kullanır (Lim 1990).

Gradyana dayalı kenar algılaması yöntemlerinin algoritmaları basit yapıda olmasından dolayı uygulaması kolaydır. Aşağıdaki yöntemlere göre kenarlar daha kalındır.



Şekil 2-26 Robert kenar algılamasının dürtü tepkeleri

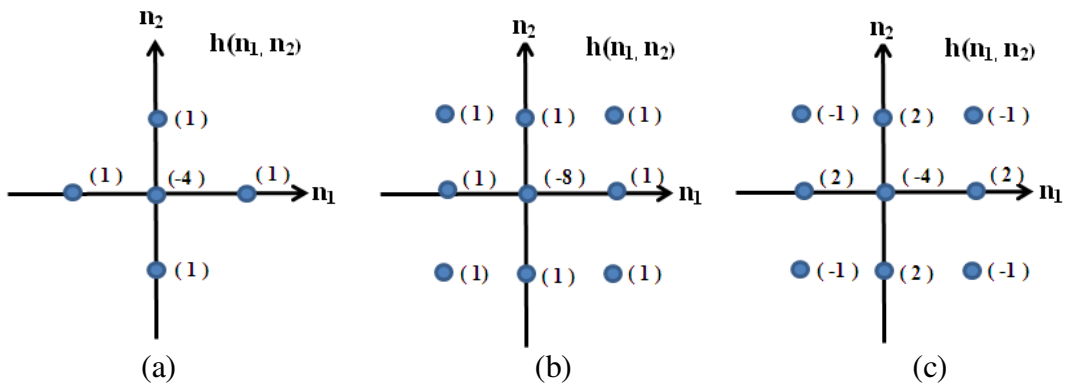
2.2.2 Laplace işlecine dayalı (Laplacian based) yöntemler

$f(x)$, bir satırdaki parlaklık değişimini gösteren fonksiyon olmak üzere; bu fonksiyonun birinci türevinin kenar noktasında maksimum değere ulaştığını, ikinci türevinin ise kenar noktasında sıfırdan geçtiğini şekil 2.20’de görmekteyiz.

$f''(x)$ 'in iki boyutlu (2-D) $f(x,y)$ 'ye güncellenmiş biçimi denklem (2.14)'de verilmiştir.

$$\nabla^2 f(x, y) = \nabla (\nabla f(x, y)) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (2.14)$$

$f(n_1, n_2)$ için 2. kısmi türevler, 2. dereceden farklar biçiminde ifade edilir. 2. dereceden farklar $f(n_1, n_2)$ 'nin uygun $h(n_1, n_2)$ süzgeç dürtü tepkisi ile katlanmasıyla elde edilebilir. Uygun süzgeçlere şekil 2.21’de örnek verilmiştir.



Şekil 2-27 Laplace işlecine dayalı kenar algılamasının dürtü tepkelerine örnekler

$f(n_1, n_2) * h(n_1, n_2)$ 'nin $\nabla^2 f(x, y)$ 'nin ayrık bir yaklaşığı olduğunu göstermek için 1. dürtü tepkisi ele alınsın.

$$\frac{\partial f(x, y)}{\partial x} \rightarrow f_x(n_1, n_2) = f(n_1 + 1, n_2) - f(n_1, n_2) \quad (2.15)$$

$$\frac{\partial^2 f(x, y)}{\partial x^2} \rightarrow f_{xx}(n_1, n_2) = f_x(n_1, n_2) - f_x(n_1 - 1, n_2) \quad (2.16)$$

(2.15) ve (2.16) denklemlerinden;

$$\frac{\partial^2 f(x, y)}{\partial x^2} \rightarrow f_{xx}(n_1, n_2) = f(n_1 + 1, n_2) - 2f(n_1, n_2) + f(n_1 - 1, n_2) \quad (2.17)$$

Aynı şekilde hesaplanırsa;

$$\frac{\partial^2 f(x, y)}{\partial y^2} \rightarrow f_{yy}(n_1, n_2) = f(n_1, n_2 + 1) - 2f(n_1, n_2) + f(n_1, n_2 - 1) \text{ olur.} \quad (2.18)$$

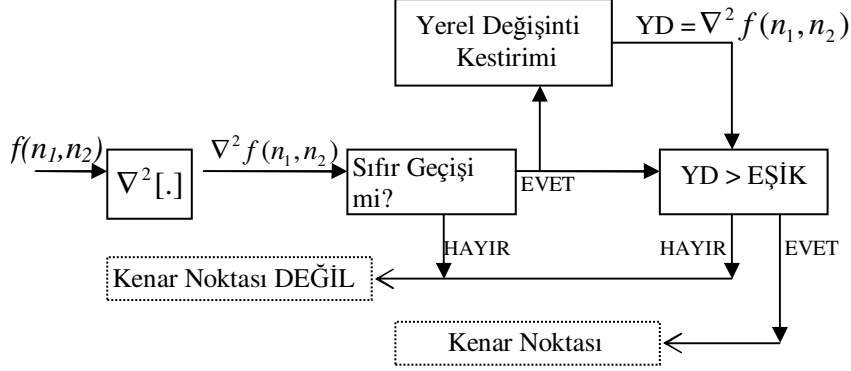
$$\begin{aligned} \nabla^2 f(x, y) &\rightarrow \nabla^2 f(n_1, n_2) = f_{xx}(n_1, n_2) + f_{yy}(n_1, n_2) \\ &= f(n_1 + 1, n_2) + f(n_1 - 1, n_2) \\ &\quad + f(n_1, n_2 + 1) + f(n_1, n_2 - 1) - 4f(n_1, n_2) \end{aligned} \quad (2.19)$$

Bu ise $f(n_1, n_2)$ 'nin şekil 2.27.a'daki $h(n_1, n_2)$ ile katlanmışdır. 2.dereceden türevlerin nasıl yaklaşıldığına göre birçok dürtü tepkisi elde edilir.

Sinyaldeki gürültü, 'yanlış' kenar üretebilir. Öncelikle gürültü azaltma yapılabilir. Başka bir yöntem ise, şekil 2.28'de görüldüğü gibi yerel değişiminin (YD) kenarda yeterli büyüklükte olması koşuluna dayanır (Lim 1990).

Yerel değişimi kestirmek için; görüntünün M piksel kadar sağına, soluna, yukarısına, aşağısına gidilir. Toplam alan $(2M+1)^2$ piksel olur. M çok küçük seçilirse gürültü de

değişinti de büyük çıkar. σ_f , f fonksiyonunun değişintisini göstermek üzere denklem (2.20)'deki gibi hesaplanır.



Şekil 2-28 İki boyutlu (2-D) Laplace işlecine dayalı kenar algılaması blok şeması

$$\sigma_f(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} [f(k_1, k_2) - m_f(k_1, k_2)]^2 \quad (2.20)$$

Burada $m_f(n_1, n_2)$ yerel ortalama kestirim değeri olup denklem (2.21)'deki gibi hesaplanır.

$$m_f(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} f(k_1, k_2) \quad (2.21)$$

Nokta kenar olduğunda değişinti çok büyük çıkar. Eşik büyük seçilirse, kenarlar gürültü gibi algılanabilir.

Laplace işlecine dayalı yöntemler gürültüler için hassas olduğu için kenar belirleme işlemi gerçekleşmeden önce süzgeç ile temizlenmelidir. Bu amaçla Gauss süzgeci kullanılırsa LoG (Laplacian of Gauss) kenar belirleme detektörü elde edilir. Bu detektör kullanılarak,

- Gauss süzgeç $g(x,y)$ ile görüntü $f(x,y)$ katlanarak yumuşatılır. Böylece gürültülü noktalar ve küçük yapıcıklar izole edilir. Ancak yumuşatma ile birlikte kenarlarda yayılma gerçekleşir.
- Görüntüyü iyileştirmek (enhancement) için Gauss süzgeç, ikinci türev elde etme bölümünde de kullanılır. Böylece gürültülü noktalar ve küçük yapıcıklar izole edilir.

$h(x,y)$:LoG operatörü

$$\begin{aligned} h(x, y) &= \Delta^2 [g(x, y) * f(x, y)] \\ &= [\Delta^2 g(x, y)] * f(x, y) \end{aligned} \quad (2.22)$$

$$\Delta^2 g(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right)^{-(x^2 + y^2) / 2\sigma^2} \quad (2.23)$$

$\Delta^2 g(x, y)$: Meksika şapkası operatörü

x ve y : Görüntü koordinatları,

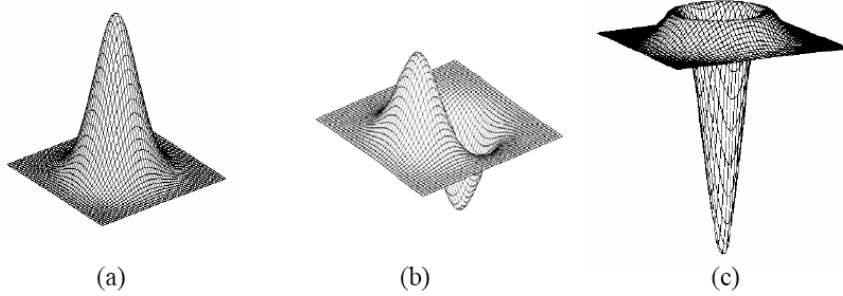
σ : Standart sapma

Standart sapma Gauss süzgecinin tek parametresidir. Süzgeç operatörlerindeki komşuluğun büyüklüğünü belirleyen özelliktir. Operatörün merkezinden uzaktaki pikseller daha küçük etkilere sahiptir ve merkezden $\sigma = 3$ 'den daha uzaktaki piksellerin etkileri önemsiz sayılır.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	+16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Şekil 2-29 LoG operatöründe kullanılan maskenin ağırlık gösterimi

Şekil 2.29’da verilen örnek 5x5 piksellik LoG maskesi şekil 2.30.(a)’da verilen Gauss fonksiyonunu, şekil 2.30.(c)’deki biçime sokar.



Şekil 2-30.a. Gauss fonksiyonu, b. Birinci türevi, c. Gauss fonksiyonun Laplace’ı.

LoG operatörü ile elde edilen kenarlar, gradyana dayalı kenar bulma yöntemlerine göre daha ince yapıya sahiptir. Kenar bulma özelliğinin güçlü olmasının yanında, büyük yerel alanları hesaba katma özelliği de mevcuttur. Gauss süzgeci kullanıldığı için gürültü düşük seviyededir. Düzeltme işlemi yapıldığında keskin köşeler kaybolmasının yanında ikinci türev alma işlemi kenar yönlerini de ortadan kaldırmıştır

2.2.3 Canny kenar algılayıcılar

Canny kenar algoritmasında öncelikle görüntünün fonksiyonu olan $f(x, y)$ ’ye Gauss $G(x, y)$ süzgeci uygulanır. Burada Gauss süzgeci kenarların belirlenmesinde ve gürültünün azaltılışında etkindir.

$$S(x, y) = G(x, y; \sigma) * f(x, y) \quad (2.24)$$

Burada $S(x, y)$ yumuşatılmış (smoothed) verilerdir.

σ : Gauss süzgecinin standart sapmasıdır. Düzeltme işleminin derecesini belirler.

Sobel, Robert veya Prewit dürtü tepkilerini kullanılarak $S(x, y)$ verilerinin birinci türevleri elde edilir.

$$P[i, j] \approx (S[i, j+1]- S[i, j]+ S[i+1, j+1]- S[i+1, j]) / 2 \quad (2.25)$$

$$Q[i, j] \approx (S[i, j]-S[i+1, j] + S[i, j+1]- S[i+1, j+1]) / 2 \quad (2.26)$$

Eğim büyüklüğü $M(x, y)$ ve yönü θ yukarıdaki denklemlerden bulunur.

$$M [i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (2.27)$$

$$\theta[i, j] = \arctan(Q[i, j] / P[i, j]) \quad (2.28)$$

Eğimin büyük olduğu yerlerde $M(x, y)$ değeri büyüktür. Kenarları bulmak için maksimum olmayan $M(x,y)$ değerlerini yok etme işlemi uygulanır.

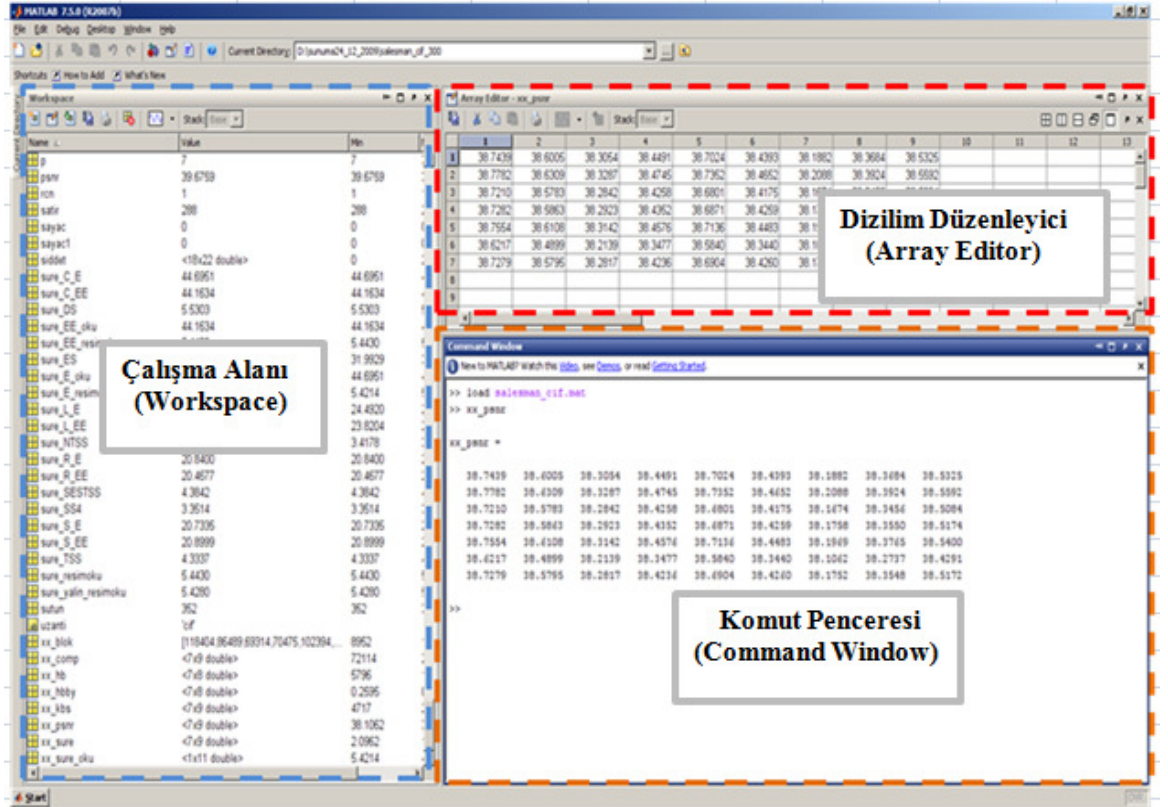
$M(x, y)$ değerleri 3x3 komşuluk ilişkisini veren bir maskeden geçirilir. Her (x, y) noktasında, merkez değer olan $M(x, y)$, her sektör tarafından iki komşusuyla karşılaştırılır. Eğer $M(x, y)$, komşu piksellerin değerlerinden büyük değilse $M(x, y)$ sıfır olarak kümelenir (Green 2002). Yanlış kenar parçacıklarının sayısı bir eşik değer kullanarak azaltılır. Eğer eşik değeri T , çok düşük seçilirse çıkış görüntüsünde yanlış kenar fazlalığı olur. T çok büyük seçildiğinde ise gerçek kenarlar da kaybolur.

Canny metodunda diğer yöntemlere göre daha ince bir kenar çıkarır. Ancak geniş piksel hatlardaki geçişlerde sıfır yoğunluk elde edilir. Yanlış kenar anlamına gelen doldurulmuş kenarlar oluşur. Bu yüzden yoğunluğu büyük olan uygulamalarda tercih edilir (Kang vd. 2006).

3. MATERYAL ve YÖNTEM

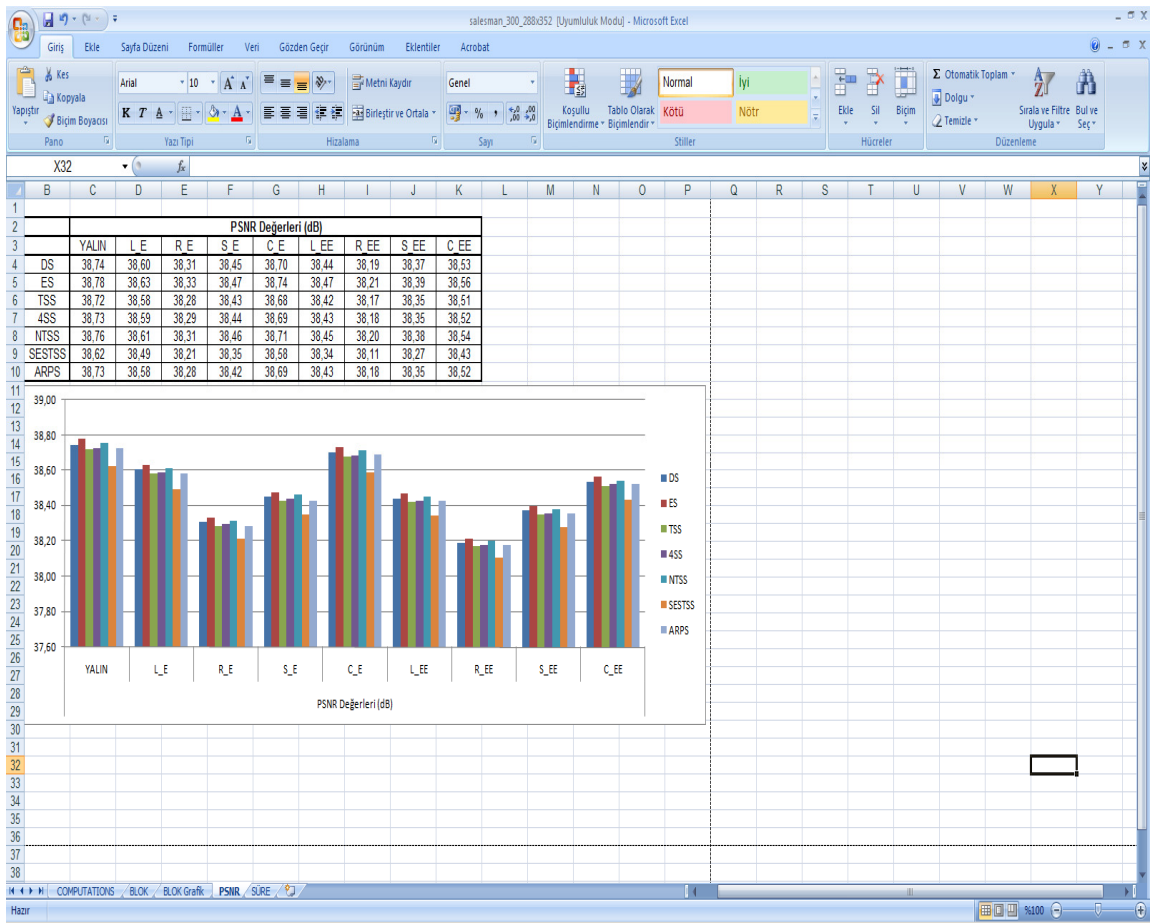
3.1 Materyal

Görüntü dizilerinde kenar bulma ile öbek eşleme algoritmalarını birbiriyle bütünleşerek hareket kestirimi yapılan bu çalışma şekil 3.1'de arayüzü gösterilen MATLAB (sürüm 7.5, R2007b) kullanılmıştır. MATLAB, teknik hesaplamalar, matematiksel problemlerin çözümü ve analizi için tasarlanmış bir yazılım geliştirme aracıdır. MATrix LABoratory kelimesinin kısaltması olan MATLAB, adında da anlaşılacağı üzere matrisler yani diğer bir deyişle diziler ile çalışır. Özellikle mühendislik alanındaki sistemlerin analizinde kullanılan görüntü işleme, sayısal işaret işleme, yapay sinir ağları, bulanık mantık, dalgacıklar (wavelets), sistem tanılama (system identification), optimizasyon (optimization), veri yakalama (data acquisition), veritabanı (database), süzgeç tasarımı (filter design) gibi araçları ile mükemmel bir ortam sunar.



Şekil 3-1 Matlab arayüzü

Kodlanan algoritmalarının sonuçlarının analizinde kullanılmak üzere, MATLAB ortamında “save” (kaydet) komutu kullanılarak “çalışma alanındaki” (workspace) değişkenler “.mat” uzantılı olarak diske kayıt edilir. Sonuçları inceleme aşamasında ”load” (aç) komutu ile kaydedilen “.mat” uzantılı dosyadaki değişkenler çalışma alanına getirilir. Çalışma alanındaki değişken çift tıklanarak bu değişkenin içerisindeki veri MATLAB ortamında “dizilim düzenleyiciden” (Array Editor) kopyalanarak şekil 3.2’de gösterilen “Microsoft Office Excel’de” (sürüm Enterprise 2007) grafiklere dönüştürülmüştür.



Şekil 3-2 Microsoft Office Excel arayüzü

Görüntü dizileri (Şekil 3.3) hareket kestirimi ile ilgili literatürde yaygın olarak kullanılan “pazarlamacı (salesman)”, “haberler (news)”, “mobil (mobile)” test dizileridir (<http://media.xiph.org/video/derf/>). Bu görüntü dizileri 300 adet ardışık CIF (Common

Intermediate Format) çerçeve formatında görüntülerden oluşmaktadır. Çerçevelerin çözünürlüğü (yatay x dikey) 352x288 pikseldir.



(a) Pazarlamacı

(b) Haberler

(c) Mobil

Şekil 3-3 Test görüntüleri

Yaygın olarak kullanılan CIF formatından elde edilmiş bazı çerçeve formatları Çizelge 3.1’de verilmiştir. Ekran çözünürlüğünün ve bit hızının düşük olduğu (görüntülü konuşma gibi) mobil çoklu ortamlarda QCIF, SQCIF; video konferans uygulamalarında CIF ve QCIF, standart tanımlı televizyon uygulamalarında ise 4CIF yaygın olarak çerçeve çözünürlüğünü belirtir.

Çizelge 3.1 Video çerçeve formatları

Format	Çerçeve Çözünürlüğü (yatayxdüsey)
Sub-QCIF (SQCIF)	128x96
Quarter CIF (QCIF)	176x144
CIF	352x288
4CIF	704x576

MATLAB uygulamasının çalıştırıldığı bilgisayarın işlemcisi Intel Xeon E5520 2.26 Ghz 8MB/1066MHz Quad Core, RAM 3x2GB DDR3-1333 ECC ‘dir.

3.2 Yöntem

Hareket kestiriminin birçok uygulama alanı olduğu gibi uygulama alanlarına göre tercih edilebilecek genel tabanlı, piksel tabanlı, öbek tabanlı ve bölge tabanlı olmak üzere temelde 4 yöntem ile hareket kestiriminde bulunulur.

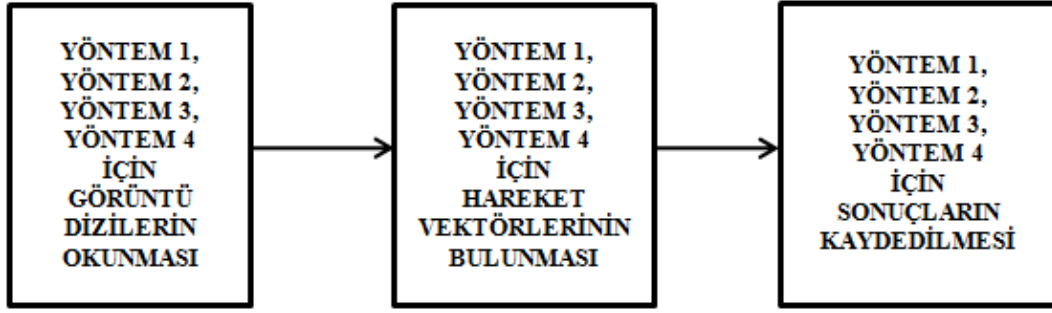
Herhangi bir uygulama (hedef takibi, veri sıkıştırma, görüntü iyileştirme, güvenlik kamerası uygulamaları gibi) için sadece bir yöntem ağırlık kazanabileceği gibi, birkaç farklı yöntem de o uygulamaya yönelik kullanılabilir. Her bir yöntem kendi içerisinde alt yöntemlere ayrılır. Örneğin öbek tabanlı hareket kestirimi “zaman bölgesi” algoritmalar ve “frekans bölgesi” algoritmalar olarak iki temel alt başlıkta ele alınır. Zaman bölgesi algoritmalar “eşleme algoritmaları” ve “gradyana dayalı algoritmalar” olarak iki alt kola ayrılır. Eşleme algoritmaları ise “öbek eşleme” ve “öznitelik eşleme” algoritmaları olarak ikiye ayrılır. Hareket kestirimi için çok sayıda yöntemin mevcut olduğu görülmektedir.

Gerek H.261, H.263, MPEG-1, MPEG-2, MPEG-4 gibi uluslararası video kodlama standartlarınca benimsenmiş olması, gerekse sadeliği ve donanım karmaşıklığının az olması açısından bu çalışmada öbek tabanlı hareket kestirim algoritması kullanılmıştır.

Bu çalışmada “pazarlamacı (salesman)”, “haberler (news)”, “mobil (mobile)” görüntü dizilerindeki çerçeve sayısı 300, çerçevelerin çözünürlüğü (yatay x dikey) 352x288 piksel, üzerinde çalışılan öbek çözünürlüğü 16x16 pikseldir.

Arama aralığı parametresi $p=7$ 'dir. Eşleme ölçütü olarak sadeliği, donanımsal olarak doğrudan uygulaması olması ve literatürde yaygın kullanılmasından dolayı mutlak farkların ortalaması (MAD) seçilmiştir.

Bu çalışmada 4 yöntem üzerinde durulmuştur.



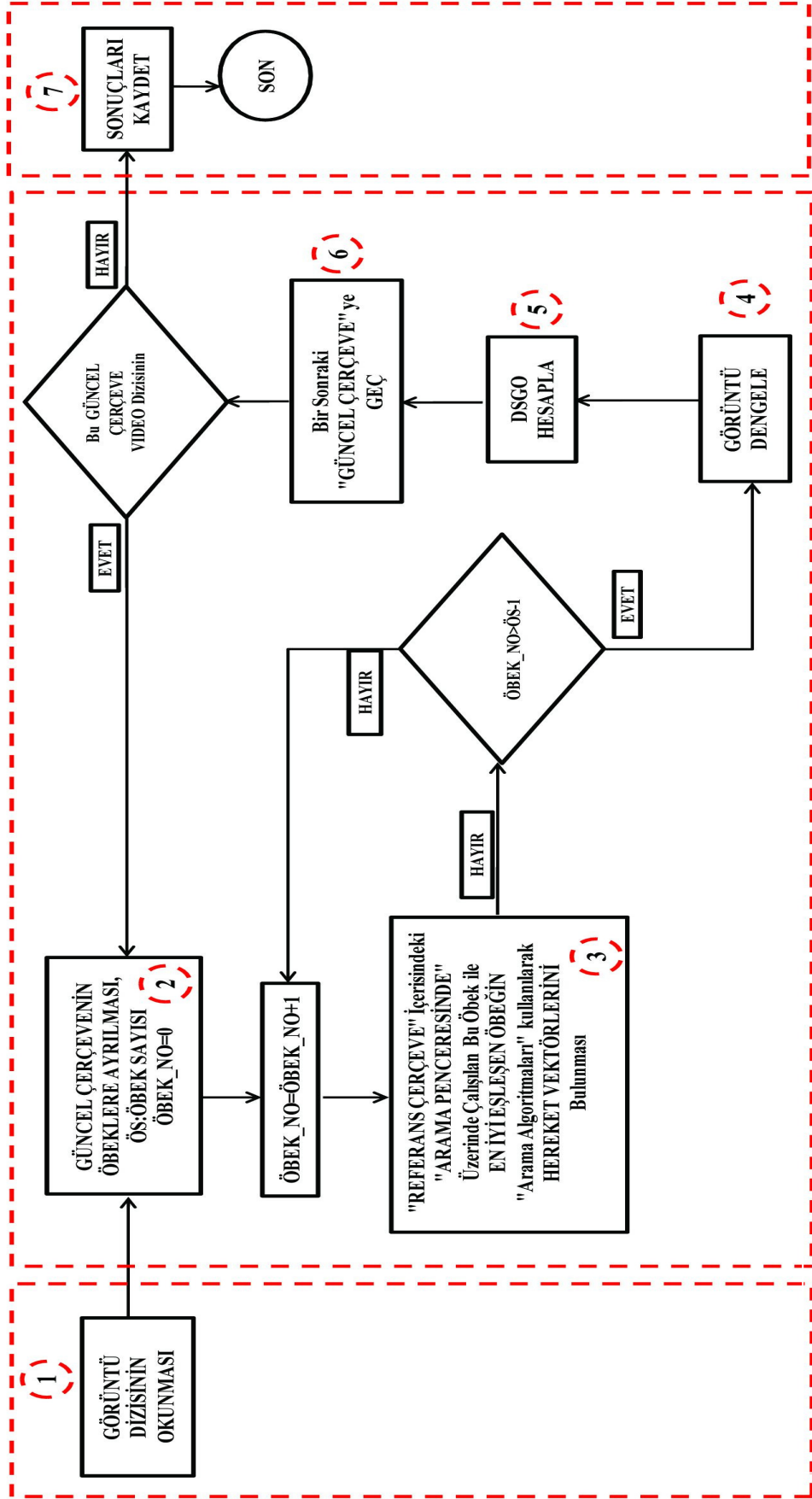
Şekil 3-4 Çalışmaların ana blok şeması

3.2.1 Yöntem-1

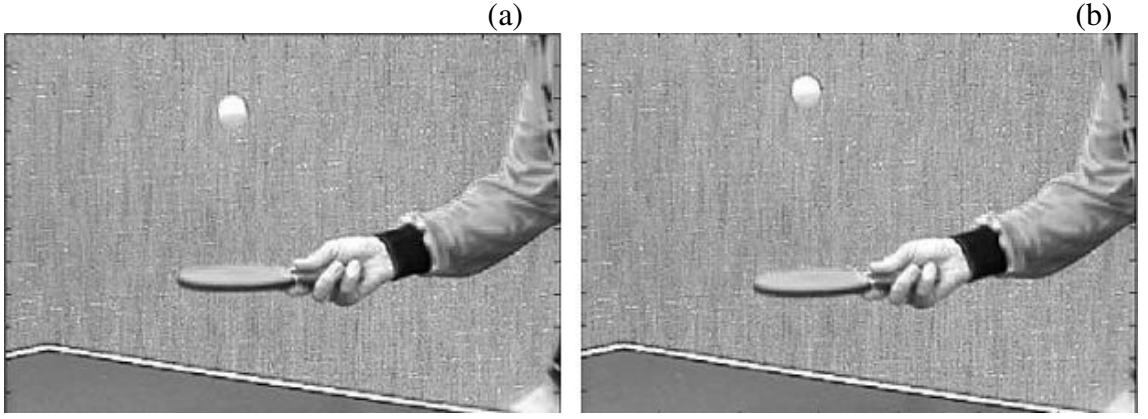
Birinci yöntem; öbek eşlemeyi temel alan, öbek büyüklüğünün sabit ve öbeklerin karesel olduğu literatürde sıklıkla karşımıza çıkan hareket kestirimidir. Şekil 3.4'deki ana blok şemanın birinci yöntem için detaylandırılmış blok şeması şekil 3.5'de gösterilmiştir.

Şekil 3.5 içerisinde **“1” nolu blokta** 352x288 piksel çözünürlükteki 300 adet çerçeve okunur. Bu çerçevelerden birincisi referans (Şekil 3.6.a), ikincisi güncel çerçeve (Şekil 3.6.b) olarak kabul edilmiştir. İleri yönlü hareket kestirimi yapıldıktan sonra ikinci çerçeve referans, üçüncü çerçeve güncel çerçeve olarak **“6” nolu blokta** belirlenmiştir. Özyinelemeler (iterations) sonucu en son referans çerçeve 299. çerçeve ve en son güncel çerçeve 300. çerçeve olunca son kez ileri yönlü hareket kestiriminde bulunulur.

İleri yönlü hareket kestirimi yapılmasının, başka bir deyişle referans çerçevenin güncel çerçeveden önce olmasının bir avantajı gerçek zamanlı uygulamalarda referans çerçevenin daha önce gelmesi yani mevcut olmasıdır.



Şekil 3-5 Birinci yöntem blok şeması

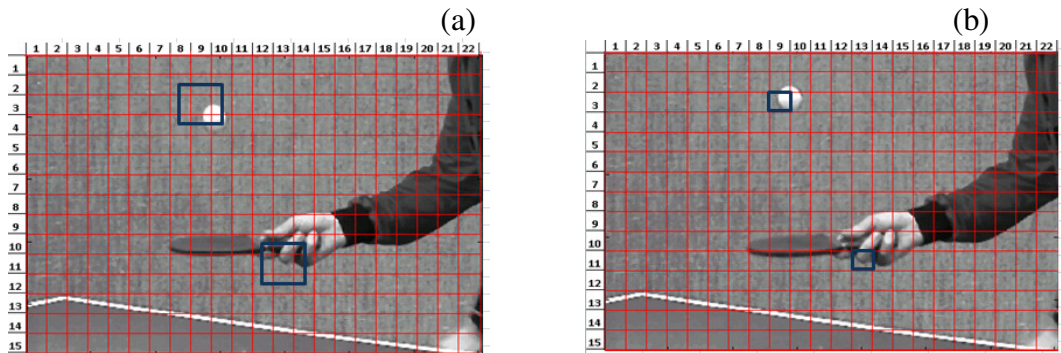


Şekil 3-6.a. Referans çerçeve,

b. Güncel çerçeve

Şekil 3.5 içerisinde “2” nolu blokta, üzerinde çalışılan **güncel çerçeve** (current frame) şekil 3.7.b’de görüldüğü gibi birbiri ile örtüşmeyen 16x16 piksellik karesel öbeklere ayrılır.

Şekil 3.5 içerisinde “3” nolu blokta her bir öbek **referans çerçevede** (reference frame) oluşturulan **arama penceresinde** (search window) (Şekil 3.7.a) belirli bir eşleme ölçütü değerine göre öbek eşleme algoritmalarından biri (Tam Arama, Üç Adımda Arama, Yeni Üç Adımda Arama, Dört Adımda Arama, Baklava Desenli Arama, Uyarlanırlı Kök Örüntü Arama, Basit ve Etkili Üç Adımda Arama) kullanılarak en iyi eşleşen öbek bulunur. En iyi eşleşen öbek ile güncel çerçevede çalışılan öbek arasındaki konumsal fark **hareket vektörünü** (motion vector) belirtir.



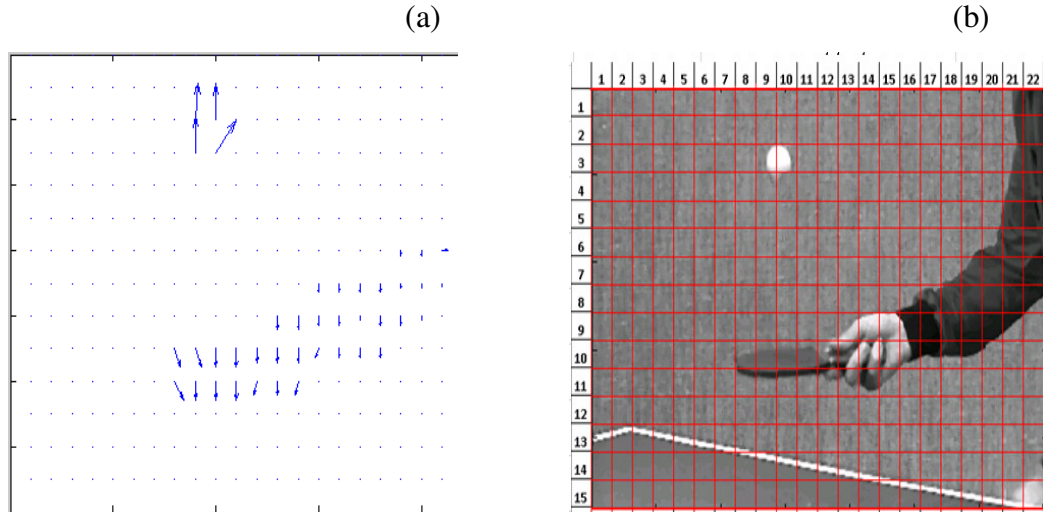
Şekil 3-7.a. Referans çerçeve-öbeklere ayrılmış,

b. Güncel çerçeve- öbeklere ayrılmış

Örneğin şekil 3.7.b'deki güncel çerçevedeki 3. satır, 9. sütun bloğu için şekil 3.7.a'daki Referans çerçevede 3. satır ve 9. sütunu +7,-7 piksel sol, sağ, üst ve altından içine alacak şekilde arama penceresi seçilir.

Bu pencere içerisinde Bölüm 2.1.4'de bahsedilen öbek eşleme arama teknikleri kullanılarak eşleme ölçütü değerini minimum veren öbek bulunur. Şekil 3.8.a'daki hareket vektörü belirlenir. Güncel çerçevedeki her bir öbek için şekil 3.5'deki "3" nolu blok gerçekleştirilir.

Şekil 3.5'deki "4" nolu blok sonucu hareket vektörleri ve referans çerçeveden yola çıkarak şekil 3.8.b'deki dengelenmiş çerçeve elde edilir.



Şekil 3-8.a. Hareket vektörleri,

b. Dengelenmiş çerçeve

Şekil 3.5'deki "5" nolu blokta güncel çerçeve ile dengelenmiş çerçeve arasındaki HKO'dan (hataların karesinin ortalamasından) yararlanarak TSGO değeri hesaplanır.

Şekil 3.5'deki "7" nolu blokta birinci yöntem için elde ettiğimiz TSGO, eşleme süreleri, hareket vektörleri gibi değerler kaydedilir.

3.2.2 Yöntem-2

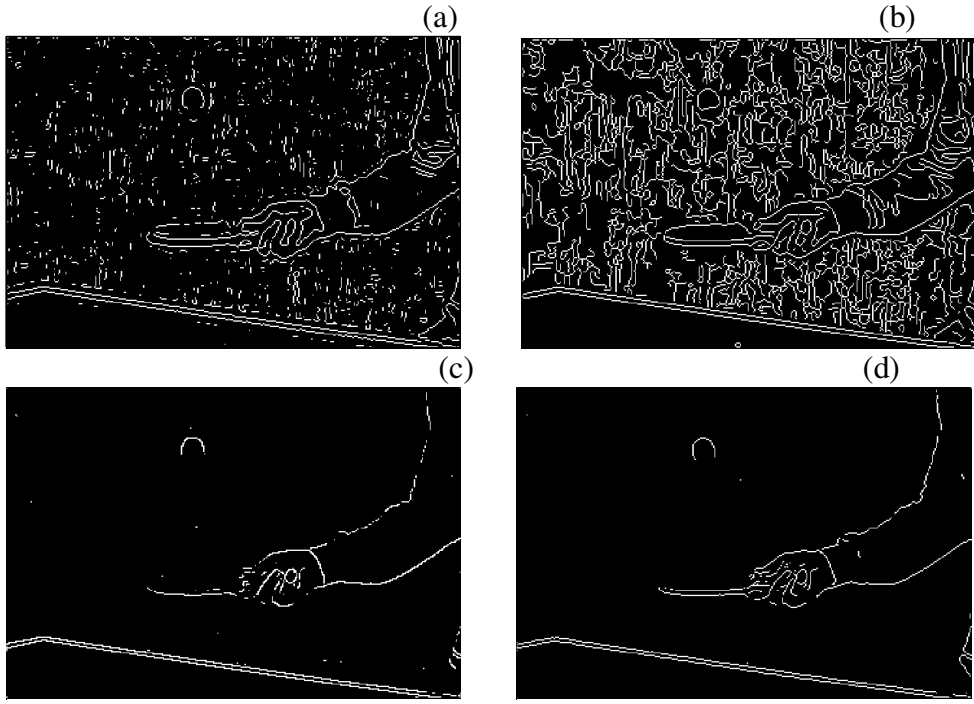
İkinci yöntem; bölüm 3.2.1’de anlatılan birinci yönteme kenar algılama algoritmaları ile değişiklik (modification) yapılması ile oluşturulur (Boyacı 2004). Kullanılan kenar algılama algoritmaları LoG kenar algılaması, Sobel kenar algılaması, Robert kenar algılaması ve Canny kenar algılaması algoritmalarıdır.

Bu yöntem kameranın sabit olduğu durumlar için görüntü dizilerinde hareket varsa bu hareketin nesneden kaynaklandığı ve her bir nesnenin kenarları ile geri plandan (tek düze, kenar olarak algılanacak kadar renk değişiminin olmadığı, fon diyebileceğimiz bölgeden) ayrıldığı tezine dayanır.

Bu yöntemde birinci yöntemde olduğu gibi güncel çerçevedeki **her bir öbek** için referans çerçevede oluşturulan arama penceresinde belirli bir eşleme ölçütü değerine göre öbek eşleme algoritmalarından biri kullanılarak en iyi eşleşen öbek bulunmaz. Kenar haritalarını inceleyerek çerçevelerde nesnelere olduğu öbekleri buluruz.

Şekil 3.9’da referans çerçevesi için kenar haritaları çeşitli kenar algılama yöntemlerine göre gösterilmiştir. Siyah renkte görülen yerlerin piksel değeri ‘0’, beyaz ile gösterilen yerlerin piksel değeri ‘1’dir. LoG ve Canny yöntemlerinde nesnelere daha belirgin çıkmakla birlikte (raketin tam olarak ortaya çıkması gibi) geri plan bölgesinde de “1” ile gösterilen ama nesne olmayan yerler kenar olarak algılanmıştır.

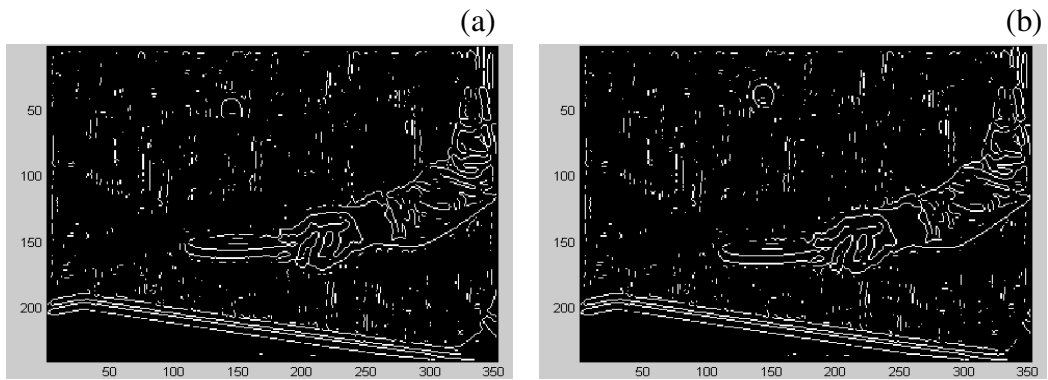
Bu durum Robert ve Sobel için geçerli değildir. Bu nedenle kenar algılama algoritması olarak LoG veya Canny yöntemleri kullanılıyorsa şekil 3.10’daki blok şemasında **“Hareketli Nesne Var mı?” (B bloğu)** sorusunun yanıtı için eşik değeri 10, Robert veya Sobel yöntemleri kullanılıyorsa eşik değeri 2 seçilmiştir. Kenar haritasında öbek içerisindeki “1” sayılarının toplamı eşik değerinden büyükse nesne var, değilse nesne yok anlamına gelir.



Şekil 3-9.a. LoG kenar haritası,
c. Robert kenar haritası,

b. Canny kenar haritası,
d. Sobel kenar haritası

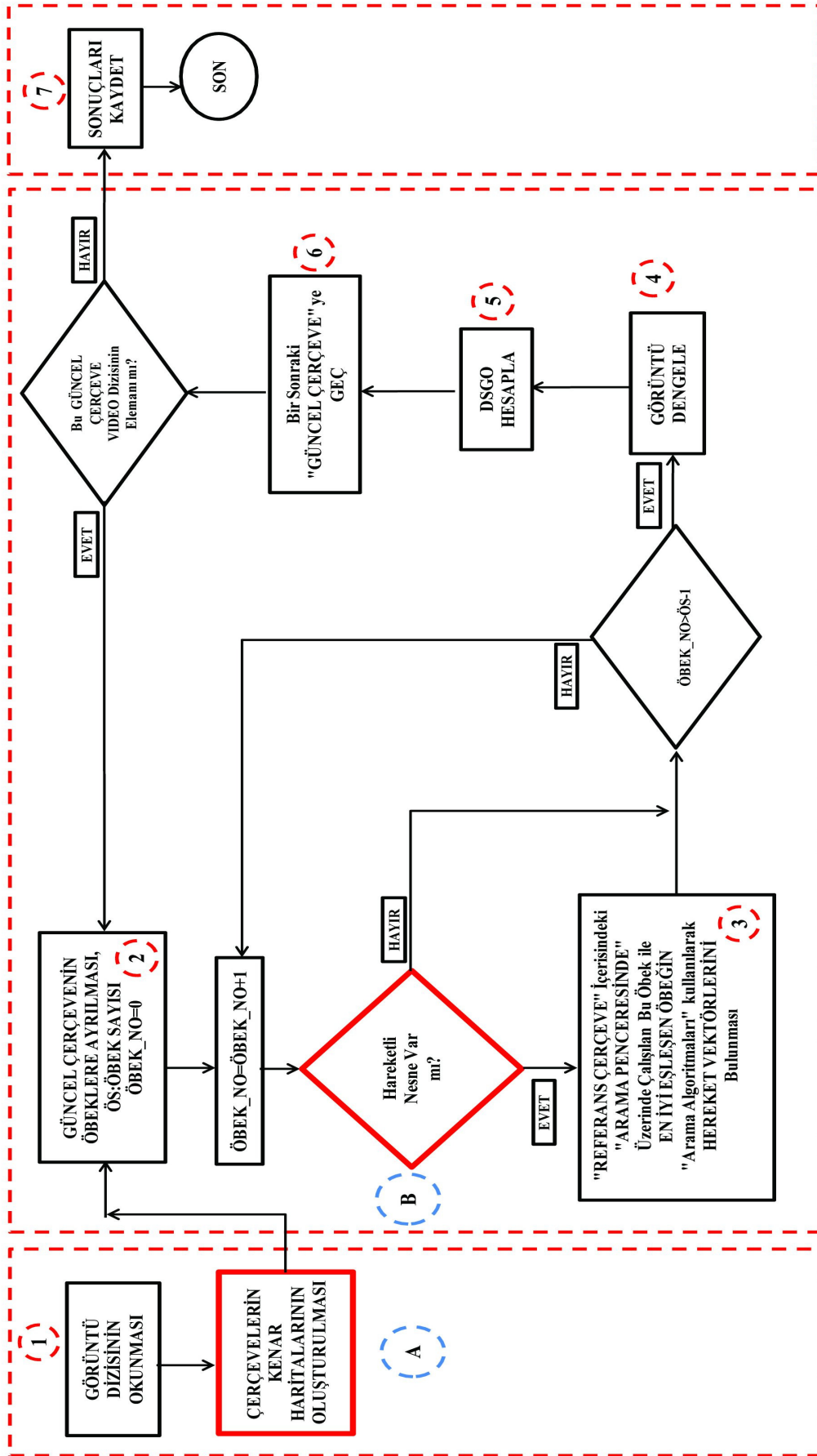
Bu yöntemde görüntü dizisi okunduktan sonra her bir çerçevenin kenar haritası bulunur. RÇKH, referans çerçeve kenar haritasını; GÇKH ise güncel çerçeve kenar haritasını ifade etmek üzere Şekil 3.10.a ve şekil 3.10.b’de Güncel ve Referans çerçevelerin kenar haritaları LoG kenar algılama algoritmasına göre gösterilmiştir.



Şekil 3-10.a. İkinci yöntem RÇKH,

b. İkinci yöntem GÇKH

Şekil 3.4’deki ana blok şemanın ikinci yöntem için detaylandırılmış blok şeması şekil 3.11’de verilmiştir. Birinci yöntemde şekil 3.11’de görülen “A” ve “B” blokları eklenmiştir.



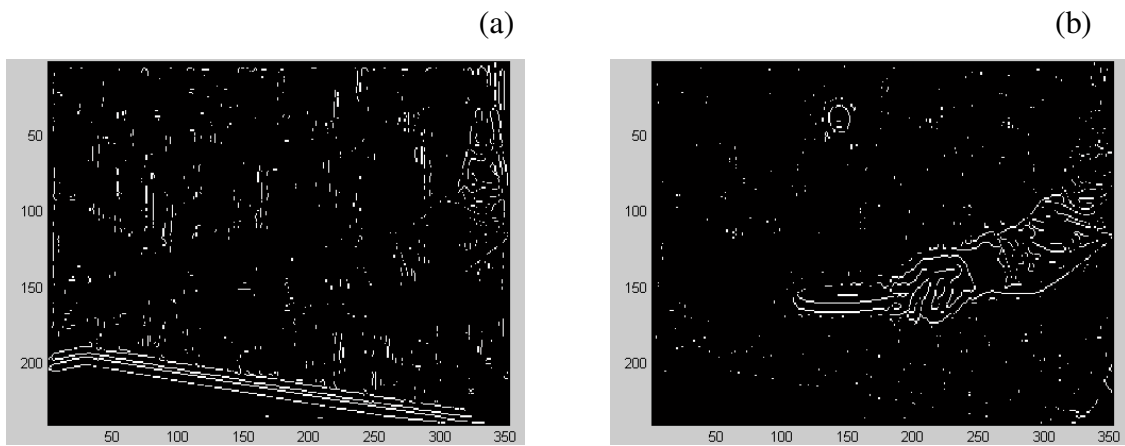
Şekil 3-11 İkinci yöntem blok şeması

Kenar haritasına dayalı bu yöntemde **referans çerçevenin kenar haritasındaki öbekler** eşik değerinden büyükse o öbek için hareket vektörleri bulunur; değilse hareket vektörü sıfır (0,0) olarak atanır. Güncel çerçevenin kenar haritasındaki öbekleri, eşik değerini geçip geçmemelerine göre hareketli nesne var veya yok şeklinde anlamlandırdığımızda TSGO değerleri daha düşük çıkmaktadır. Bu yöntemde referans çerçevenin kenar haritasına şekil 3.11'deki **“Hareketli nesne var mı?” (B bloğu)** sorusu sorulur.

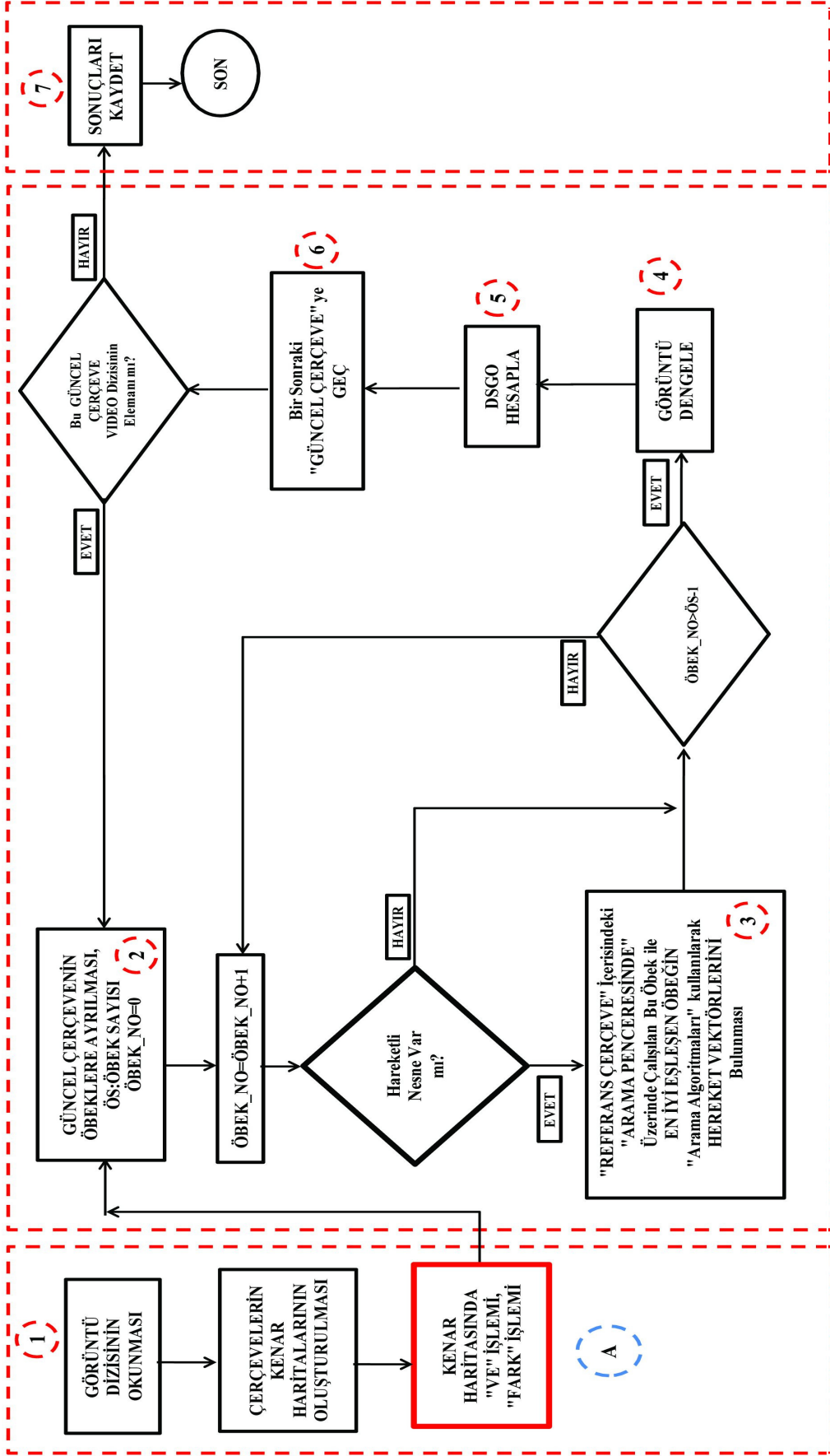
3.2.3 Yöntem-3

İkinci yöntemde kenar haritaları incelediğimizde sadece hareket eden nesneye ait kenar bilgileri değil, görüntüde hareket etmeyen nesnelere ait kenar bilgileri de gözlenmektedir.

RÇKH ile GÇKH, VE kapısından (and gate) geçirilirse şekil 3.12.a ortaya çıkar. RÇKH'den VE kapısının çıkışının (Şekil 3.12.a) farkını alırsak şekil 3.12.b elde edilmiş olur (Boyacı 2004). Şekil 3.4'deki ana blok şemanın üçüncü yöntem için detaylandırılmış blok şeması şekil 3.13'de verilmiştir.



Şekil 3-12.a. VE kapısı çıkışı kenar haritası, b. Üçüncü yöntem kenar haritası



Şekil 3-13 Üçüncü yöntem blok şeması

Üçüncü yöntemdeki eşik değerleri ikinci yöntemle aynıdır. Bu yöntemin ikinci yöntemden tek farkı, referans çerçeve kenar haritasının yerine “VE kapısı” ile “fark” işlemleri sonucu elde edilen kenar haritasını (Şekil 3.12.b) temel aldığı şekil 3.13’te ki “A” bloğudur.

“A” bloğunda elde edilen kenar haritasına işlenmiş kenar haritası, İKH aşağıdaki gibi tanımlanır:

$$\text{İKH} = \text{RÇKH} - (\text{GÇKH} * \text{RÇKH}) \quad (3.1)$$

İKH’nin alabileceği olası değerler çizelge 3.2’de verilmiştir.

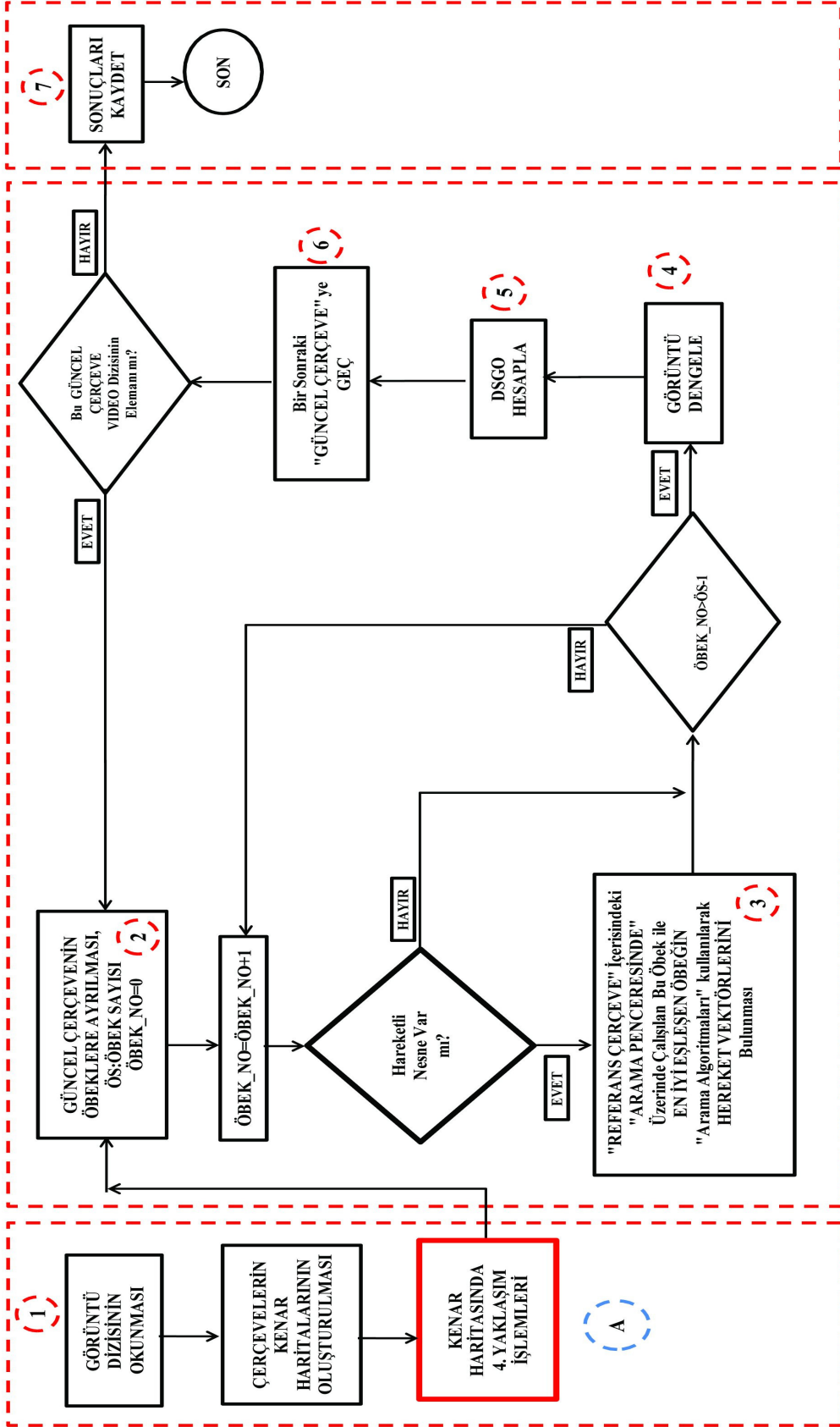
Çizelge 3.2 Üçüncü yöntem İKH değerler tablosu

İKH	RÇKH	GÇKH
0	0	0
0	0	1
1	1	0
0	1	1

İKH oluşturulurken, RÇKH’nin “1” ve GÇKH’nin “0” olduğu pikseller “1”, diğer pikseller “0” olarak hesaplanır. RÇKH’nin GÇKH’yi temel alarak hesaplanan TSGO değerlerinin daha yüksek olması nedeni ile İKH oluşturulurken RÇKH’den VE kapısından geçirilmiş harita çıkarılmıştır.

3.2.4 Yöntem-4

Bu yöntem ikinci yöntemi arama süresi, üçüncü yöntemi ise TSGO değeri açısından iyileştirme amacı ile tasarlanmıştır. Şekil 3.4’deki ana blok şemanın dördüncü yöntem için detaylandırılmış blok şeması şekil 3.14’de verilmiştir.



Şekil 3.3-144 Dördüncü yöntem blok şeması

Dördüncü yöntemdeki eşik değerleri ikinci yöntemle aynıdır. Bu yöntemin blok şemasının üçüncü yöntemden tek farkı “A” bloğundaki işlemin farklı oluşudur. Bu yöntemdeki İKH aşağıdaki gibi tanımlanır:

$$\text{İKH} = \text{Mutlak Değer}(\text{GÇKH} - 2 * \text{RÇKH} + (\text{GÇKH} * \text{RÇKH})) \quad (3.2)$$

Çizelge 3.3 Dördüncü yöntem İKH değerler tablosu-RÇKH ağırlıklı

İKH	RÇKH	GÇKH
0	0	0
1	0	1
2	1	0
0	1	1

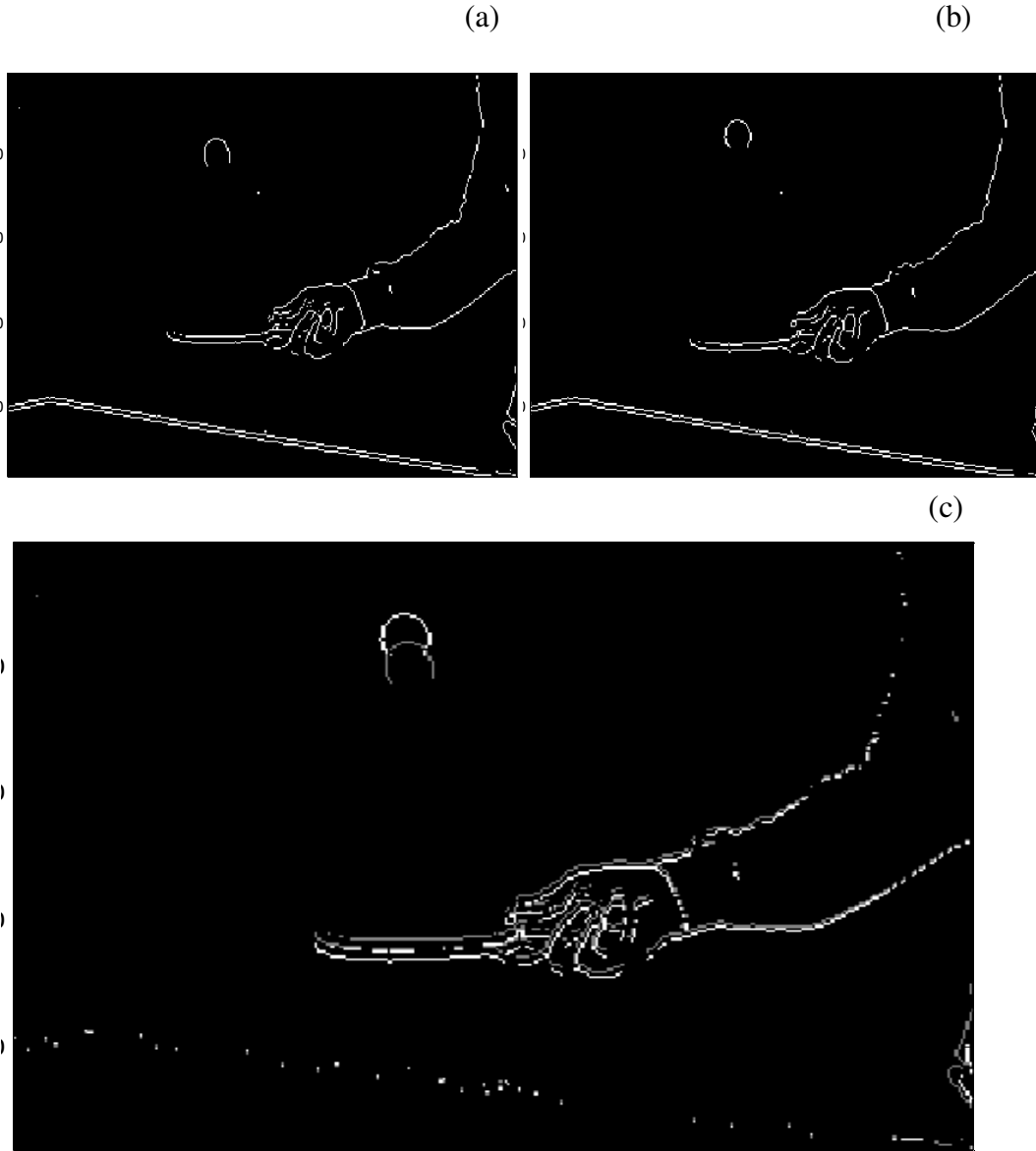
Çizelge 3.3’de RÇKH’nin ve GÇKH’nin aynı olduğu yerler için, üçüncü yöntemde olduğu gibi eleme yapıldığı, aynı olmadığı yerlerde “0” değerinden büyük olduğu görülmektedir. Sadece RÇKH değil GÇKH de İKH içerisinde her ikisinde aynı olmadığı yerler için temsil edilmiş olur. Çizelge 3.3’de RÇKH’nin “1” olduğu yerde İKH “2” olabilmektedir. RÇKH’nin İKH içerisinde ağırlıklı olmasını sağlamak için 3.2’deki denklem 3.3’deki denkleme göre tercih edilmiştir.

$$\text{İKH} = \text{Mutlak Değer}(\text{RÇKH} - 2 * \text{GÇKH} + (\text{GÇKH} * \text{RÇKH})) \quad (3.3)$$

Çizelge 3.4’te görüldüğü üzere GÇKH’nin “1” olduğu yerde İKH “2” olabilmektedir.

Çizelge 3.4 Dördüncü yöntem İKH değerler tablosu-GÇKH ağırlıklı

İKH	RÇKH	GÇKH
0	0	0
2	0	1
1	1	0
0	1	1



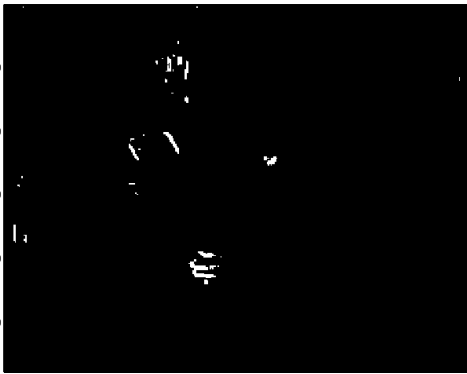
Şekil 3.15.a. Dördüncü yöntem RÇKH, b. Dördüncü yöntem GÇKH,
c. Dördüncü yöntem İKH

Şekil 3.15.c’de, 3.2’deki denklemin ve çizelge 3.2’deki tablonun uygulama içerisindeki ifadesini görmekteyiz. Beyaz renk ‘2’, gri renk ‘1’, siyah renk ‘0’ olarak gözlemlenir. Bu yöntem ile eşik seviyesi aynı tutulmakla beraber hareketin olduğu öbeklerdeki ‘1’lerin toplamı eşik seviyesini daha kolay geçeceğinden ve GÇKH de İKH içerisinde yer bulduğu için ve sabit bölgeler üçüncü yöntemde olduğu gibi elendiğinden hem TSGO , hem de arama süresi açısından iyileşme gözlenmesi beklenilmektedir.

3.2.5 Literatür karşılaştırması

Nasır (2007) tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü'nde hazırlanan“ 2-Boyutlu Hareket Kestirimi ve Video Çerçeve Aradeğerlendirmesinde Uygulanması” konulu yüksek lisans tezinin bölüm 4.3 maddesinde “Tezde Önerilen Yöntem” başlığı altındaki çalışma yeniden gerçekleştirilmiştir. Nasır'ın çalışmasının “1” ve “0” değerlerini içeren çerçevelere ve belirli bir eşik seviyesine göre arama algoritmalarını adapte etmesi nedeni ile genel düşünce olarak bu çalışma ile benzerlik göstermektedir. Tezimdeki yöntemlerle karşılaştırırken “arama sürelerini” aynı bilgisayar ortamında elde etmek, aynı çerçeve büyüklüğü için aynı görüntü dizilerini kullanarak karşılaştırma için uygun hale getirmek amaçlanmıştır.

Yeniden gerçekleştirme aşamasında; güncel çerçeve ile referans çerçeve arasında “fark çerçevelerinin” mutlak değeri 3x3 piksel “ortalama süzgecinden” geçirilir. Elde edilen sonuç için, Matlab R2007b'de “graythresh” komutundan yararlanarak Otsu (1979) metodundaki eşik değeri hesaplanır. Pazarlamacı görüntü dizisi için bu eşik değeri 0.4980 çıkmıştır. Bu eşik değeri, Nasır'ın elde ettiği eşik değerinden farklı olduğundan (bu değer 2007 yılındaki çalışmada 0.0196 çıkmış ve Matlab sürümündeki farklılıklar, referans alınan skalalar gibi nedenlerle farklılık oluşmuş olabilir) Nasır (2007) yüksek lisans tezi 56. sayfasındaki pazarlamacı hareket değişimini gösteren ikili görüntüsüne yakın çerçeve elde etmek için eşik değerini 10 ile genişletilmiştir. Eşik seviyesinin 10 kat genişletildiği durum için elde edilen harita şekil 3.16'da görülmektedir.



Şekil 3.16 Hareket değişimini gösteren ikili görüntü

Bu karşılaştırma, ikinci ve üçüncü yöntemlerin TSGO ve “arama süresi” değerleri açısından dengelendiği dördüncü yöntem için yapılmıştır.

Karşılaştırma “pazarlamacı (salesman)”, “haberler (news)” görüntü dizileri için 100 çerçevede (çerçevelerin çözünürlüğü (yatay x dikey) 352x288 piksel, üzerinde çalışılan öbek çözünürlüğü 8x8 piksel, alt öbek çözünürlüğü 4x4 piksel), maksimum arama aralığı 7, minimum arama aralığı 2 parametre değerleri kullanılarak yapılmıştır.

4. BULGULAR

Bu bölümde “pazarlamacı (salesman)”, “haberler (news)”, “mobil (mobile)” görüntü dizilerinde hareket kestirimi bulguları ve kullanılan kısaltmalar aşağıda verilmiştir.

- **Hesaplama:** Üzerinde çalışılan öbek ile en iyi eşleşen öbeği bulmak için kontrol edilen aday öbek sayısıdır. Bu değer milyon mertebesinde olduğu için “Hesaplama” değerleri 1.000’e bölünmüştür. Örneğin 2.748 değeri $2.748 \times 1000 = 2.748.000$ ’i göstermektedir.
- **Öbek sayısı (ÖS):** Hareket kestiriminde bulunulan öbek sayısıdır. Bu sayı birinci yöntem için tüm çerçevelerdeki (300 çerçeve) 16x16 piksellik toplam öbek sayısıdır.

$$\text{Öbek Sayısı} = (\text{Çerçeve sayısı} - 1) \times (\text{Çerçeve çözünürlüğü}) / (\text{Öbek çözünürlüğü})$$
$$\text{Öbek Sayısı} = 299 \times (352 \times 288) / (16 \times 16) = 118.404$$

İkinci, üçüncü ve dördüncü yöntemler için öbek sayısı, eşik değeri geçen öbeklerin sayısını belirtmektedir.

- **Hareket Vektörünün (0,0) Olmadığı Öbek Sayısı:** Yatay veya düşey konumda hareketin olduğu öbeklerin sayısıdır. **HVOÖ** (Hareket Vektörünün Olduğu Öbek) sayısı şeklinde kısaltılmıştır.

İkinci, üçüncü ve dördüncü yöntemlerde eşik değeri geçemeyen öbekler hareket vektörünü belirlemede elendiği için, eşik seviyeyi geçen öbeklerden kaçının birinci yöntemde hareket vektörü algılanan öbek ile aynı olduğunun ölçüsüdür.

- **Hareket Vektörünü Bulma Yüzdesi (HVBY):** İkinci, üçüncü ve dördüncü yöntem için hareket vektörünün (0,0)'dan farklı olduğu öbek sayısının, birinci yöntemde bulunan HVÖÖ (hareket vektörünün olduğu öbek) sayısına oranıdır.
- **Tepe Sinyal Gürültü Oranı, TSGO (peak to signal noise ratio, PSNR):** 300 çerçevelik görüntü dizisi için 299 tane hareket vektörü ve 299 tane TSGO değeri bulunur. Bulgularda bahsi geçen TSGO değeri, tüm görüntü dizisindeki TSGO değerlerinin aritmetik ortalamasını ifade eder.
- **Arama Süresi:** Referans çerçevede arama penceresi içerisinde en iyi eşleşen öbeği bulmak için geçen süredir. Birimi saniyedir.

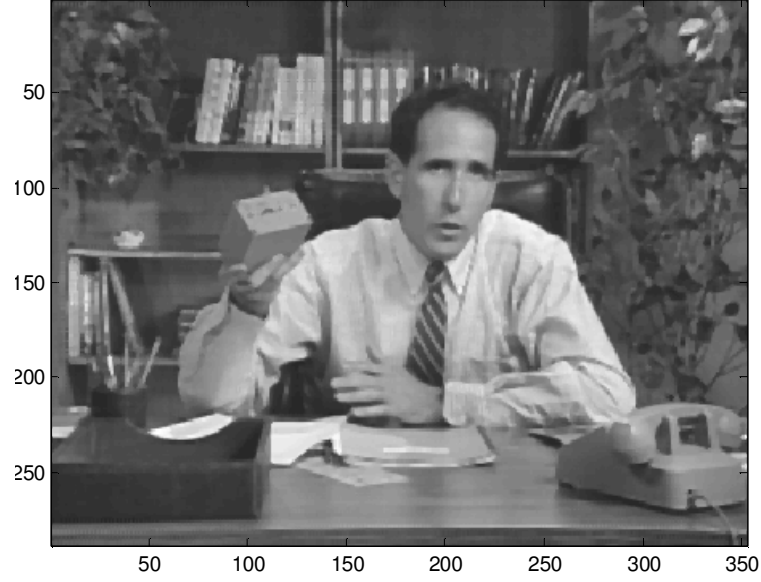
Çizelge 4.1 Bulgular kısaltma-arama teknikleri

BDA:	Baklava Desenli Arama (Diamond Search, DS)
TA:	Tam Arama (Full Search,FS)
ÜAA:	Üç Adımda Arama (Three Step Search, TSS)
DAA:	Dört Adımda Arama (Four Step Search, 4SS)
YÜAA:	Yeni Üç Adımda Arama (New Three Step Search, NTSS)
BEÜAA:	Basit ve Etkili Üç Adımda Arama (Simple and Efficient Search, SETSS)
ÜKÖA:	Uyarlanırlı Kök Örüntü Arama (Adaptive Root Pattern Search, ARPS)

Çizelge 4.2 Bulgular kısaltma-yöntemler

YK1:	Birinci Yöntem
YK2:	İkinci Yöntem
YK3:	Üçüncü Yöntem
YK4:	Dördüncü Yöntem

4.1 “Pazarlamacı” Görüntü Dizisi İçin Bulgular

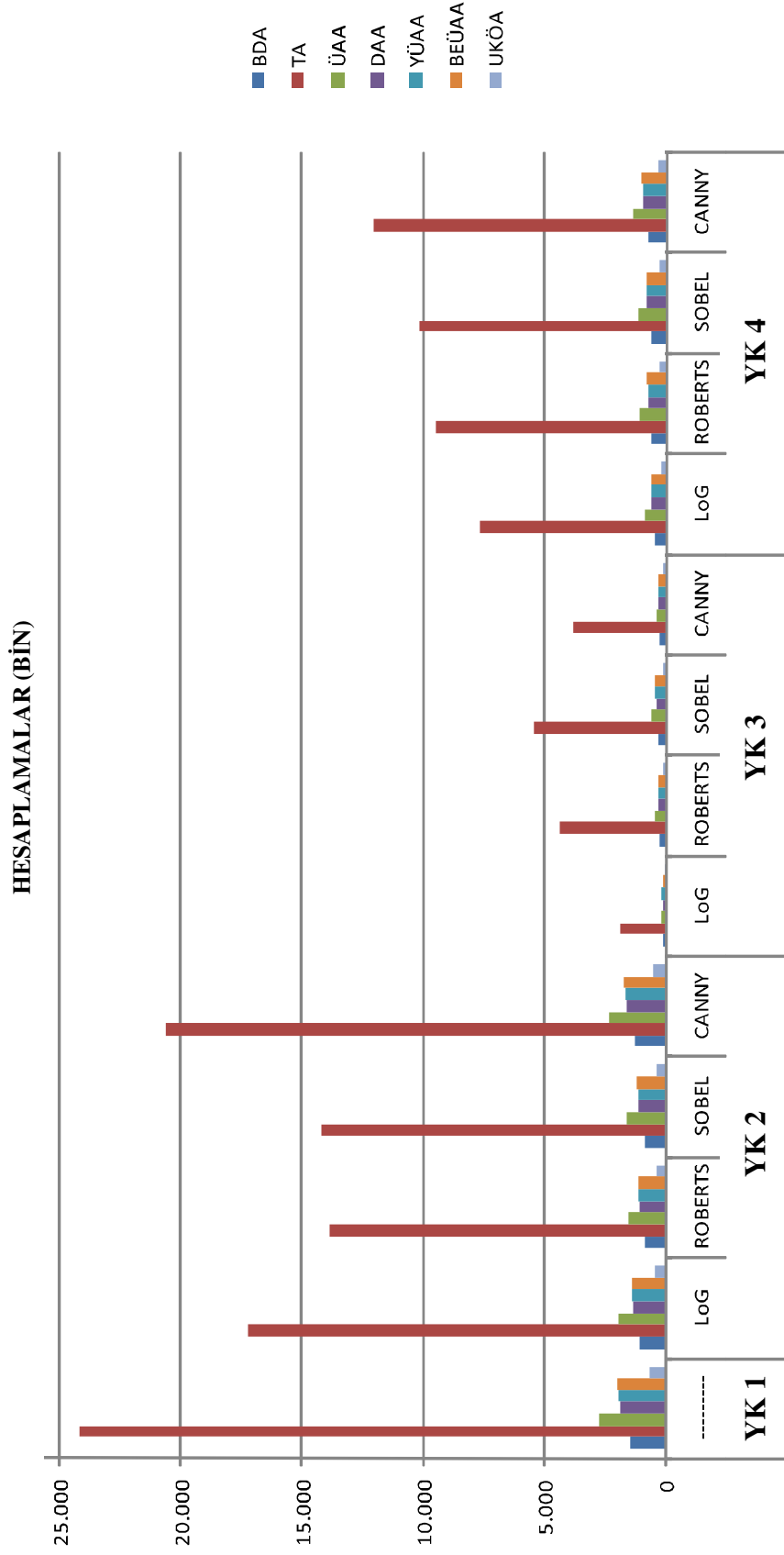


Şekil 4-1 Pazarlamacı

4.1.1 “Pazarlamacı” görüntü dizisi için hesaplamalar (aday öbek sayısı)

Çizelge 4.3 Pazarlamacı-hesaplamalar (aday öbek sayısı)

	HESAPLAMALAR (BIN)												
	YK1	YK2				YK3				YK4			
	-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	1.490	1.067	857	876	1.272	132	273	344	252	489	585	626	746
TA	24.188	17.263	13.864	14.191	20.621	1.897	4.358	5.488	3.849	7.710	9.476	10.146	12.064
ÜAA	2.748	1.973	1.583	1.618	2.351	214	491	618	434	877	1.070	1.147	1.364
DAA	1.896	1.360	1.092	1.117	1.622	158	344	433	310	614	743	796	947
YÜAA	1.947	1.393	1.119	1.145	1.660	173	358	449	328	639	764	818	974
BEÜAA	2.004	1.448	1.158	1.180	1.722	151	353	445	310	640	777	833	991
UKÖA	660	472	379	388	562	72	126	158	125	229	260	279	335

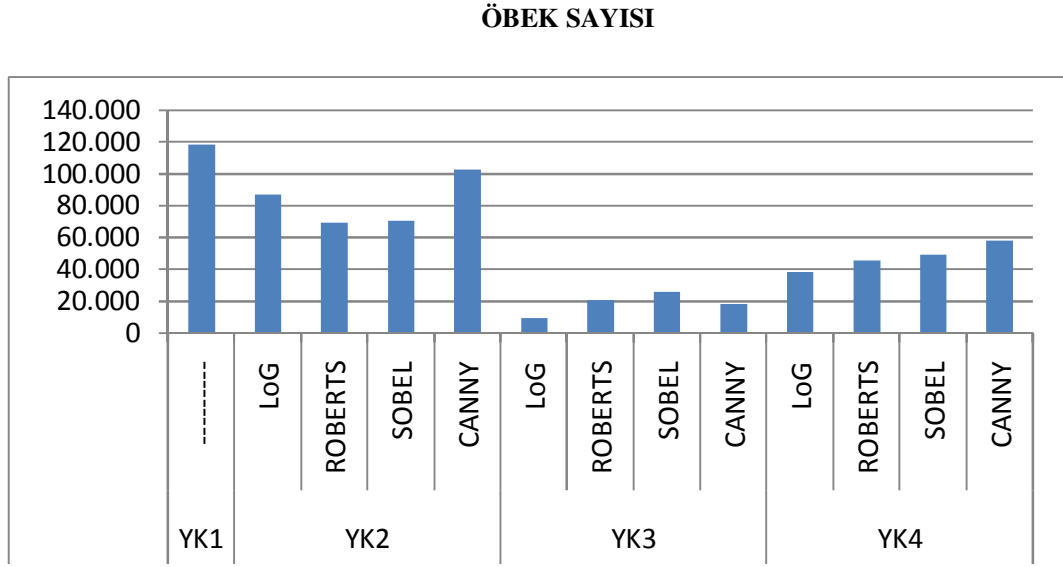


Şekil 4-2 Pazarlamacı-hesaplamalar (aday öbek sayısı)

4.1.2 “Pazarlamacı” görüntü dizisi için öbek sayısı (ÖS)

Çizelge 4.4 Pazarlamacı-öbek sayısı

ÖBEK SAYISI												
YK1	YK2				YK3				YK4			
-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
118.404	86.489	69.314	70.475	102.394	8.952	20.520	25.901	18.208	37.910	45.214	48.739	57.892



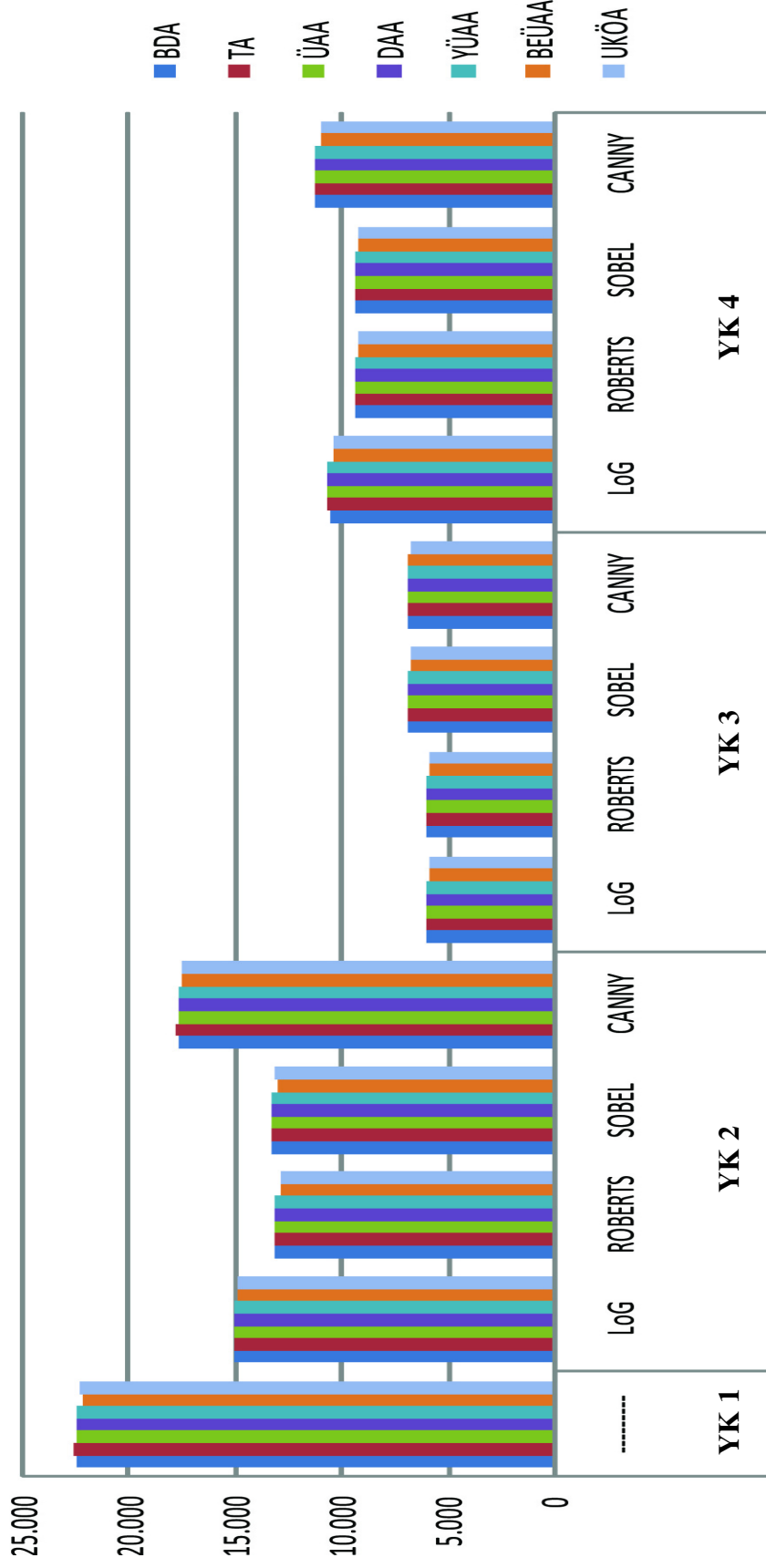
Şekil 4-3 Pazarlamacı-öbek sayısı

4.1.3 “Pazarlamacı” görüntü dizisi için HVOÖ

Çizelge 4.5 Pazarlamacı-HVOÖ

	HVOÖ SAYISI												
	YK1	YK2				YK3				YK4			
		LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	22.565	15.054	13.090	13.268	17.733	5.971	5.953	6.831	6.915	10.599	9.392	9.352	11.198
TA	22.610	15.087	13.120	13.299	17.769	5.993	5.975	6.854	6.941	10.633	9.422	9.383	11.234
ÜAA	22.576	15.063	13.096	13.276	17.743	5.979	5.958	6.838	6.924	10.608	9.400	9.361	11.208
DAA	22.568	15.057	13.091	13.270	17.736	5.974	5.953	6.833	6.918	10.602	9.394	9.354	11.201
YÜAA	22.574	15.061	13.095	13.274	17.741	5.978	5.958	6.836	6.923	10.606	9.398	9.359	11.206
BEÜAA	22.196	14.845	12.912	13.077	17.479	5.902	5.874	6.722	6.820	10.430	9.266	9.207	11.016
UKÖA	22.333	14.873	12.894	13.085	17.549	5.844	5.796	6.665	6.781	10.421	9.204	9.173	11.021

HVOÖ SAYISI



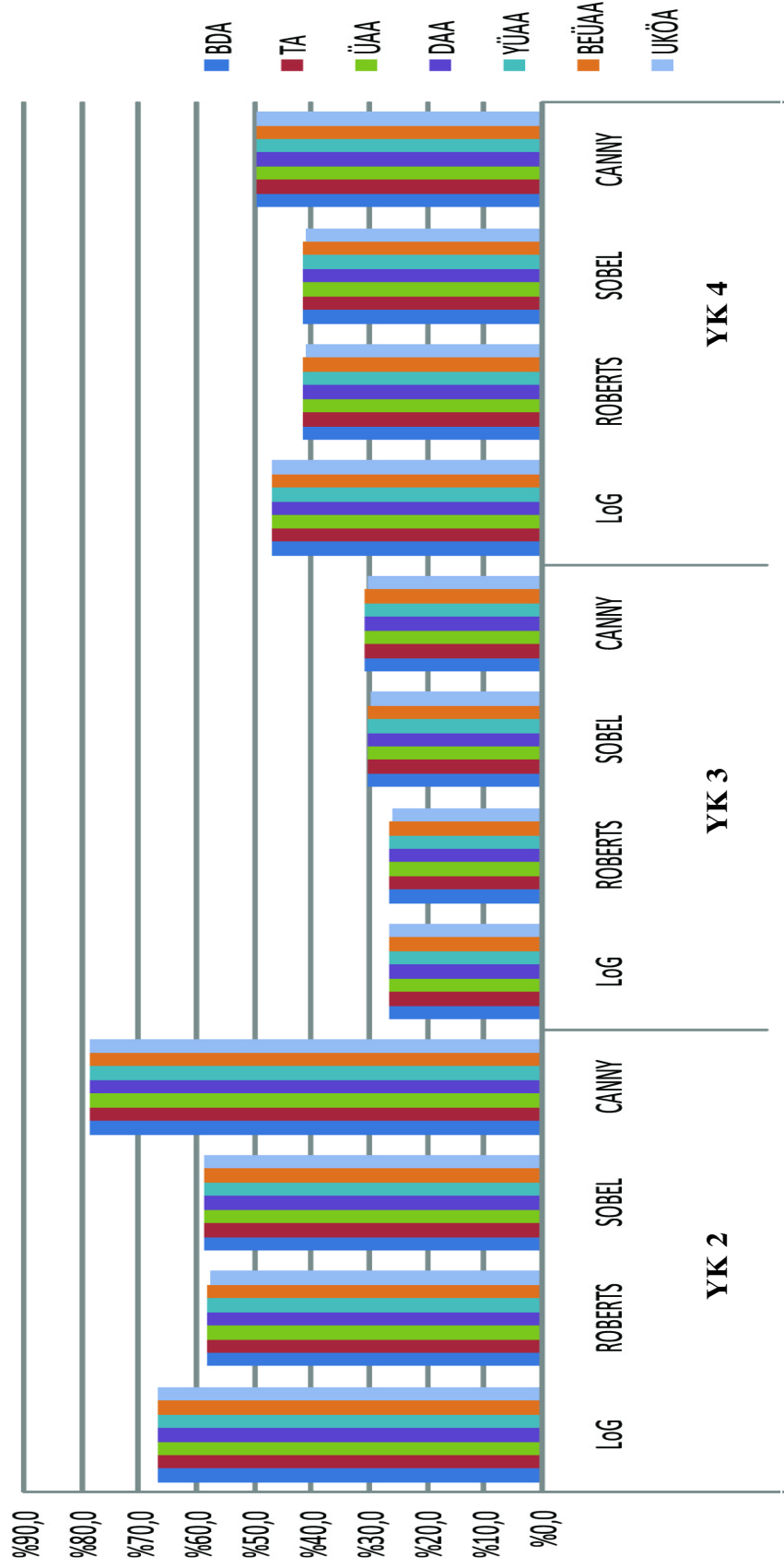
Sekil 4-4 Pazarlamacı-HVOÖ

4.1.4 “Pazarlamacı” görüntü dizisi için HVBY

Çizelge 4.6 Pazarlamacı-HVBY

	HVBY											
	YK2				YK3				YK4			
	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	%66,7	%58,0	%58,8	%78,6	%26,5	%26,4	%30,3	%30,6	%47,0	%41,6	%41,4	%49,6
TA	%66,7	%58,0	%58,8	%78,6	%26,5	%26,4	%30,3	%30,7	%47,0	%41,7	%41,5	%49,7
ÜAA	%66,7	%58,0	%58,8	%78,6	%26,5	%26,4	%30,3	%30,7	%47,0	%41,6	%41,5	%49,6
DAA	%66,7	%58,0	%58,8	%78,6	%26,5	%26,4	%30,3	%30,7	%47,0	%41,6	%41,4	%49,6
YÜAA	%66,7	%58,0	%58,8	%78,6	%26,5	%26,4	%30,3	%30,7	%47,0	%41,6	%41,5	%49,6
BEÜAA	%66,9	%58,2	%58,9	%78,7	%26,6	%26,5	%30,3	%30,7	%47,0	%41,7	%41,5	%49,6
UKÖA	%66,6	%57,7	%58,6	%78,6	%26,2	%26,0	%29,8	%30,4	%46,7	%41,2	%41,1	%49,3

HVBY

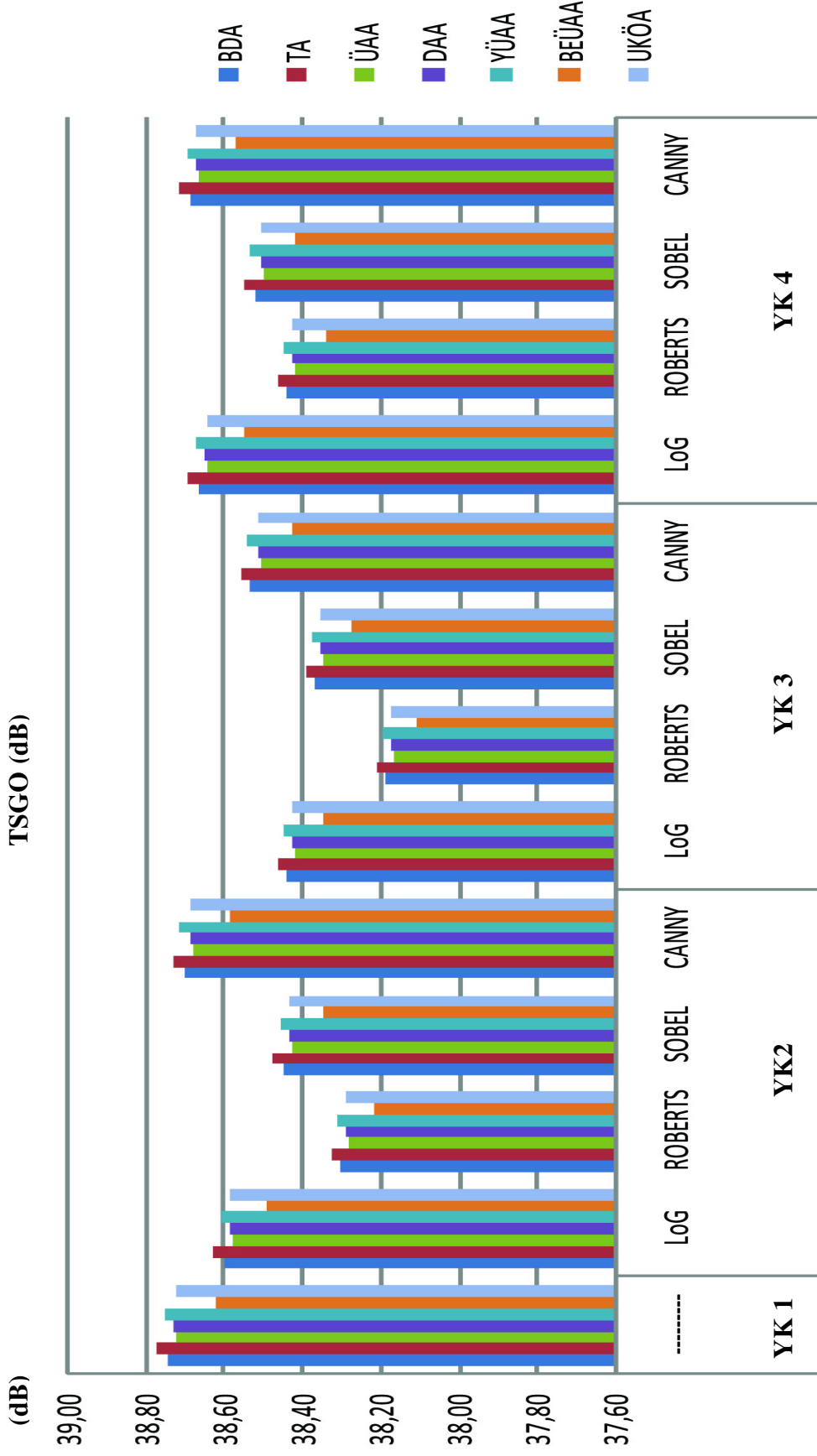


Şekil 4-5 Pazarlamacı-HVBY

4.1.5 ‘‘Pazarlamacı’’ görüntü dizisi için TSGO (PSNR)

Çizelge 4.7 Pazarlamacı-TSGO

		TSGO (dB)											
YK1		YK2				YK3				YK4			
-----		LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	38,74	38,60	38,31	38,45	38,70	38,44	38,19	38,37	38,53	38,66	38,44	38,52	38,69
TA	38,78	38,63	38,33	38,47	38,74	38,47	38,21	38,39	38,56	38,70	38,47	38,55	38,72
ÜAA	38,72	38,58	38,28	38,43	38,68	38,42	38,17	38,35	38,51	38,64	38,42	38,50	38,66
DAA	38,73	38,59	38,29	38,44	38,69	38,43	38,18	38,35	38,52	38,65	38,42	38,51	38,67
YÜAA	38,76	38,61	38,31	38,46	38,71	38,45	38,20	38,38	38,54	38,67	38,45	38,53	38,70
BEÜAA	38,62	38,49	38,21	38,35	38,58	38,34	38,11	38,27	38,43	38,55	38,34	38,42	38,57
UKÖA	38,73	38,58	38,29	38,43	38,69	38,43	38,18	38,35	38,52	38,65	38,43	38,51	38,67



Sekil 4-6 Pazarlamacı-TSGO

4.1.6 “Pazarlamacı” görüntü dizisi için arama süresi

Çizelge 4.8 Pazarlamacı-arama süreleri

	ARAMA SÜRELERİ (SN)												
	YK1	YK2			YK3			YK4					
		LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	26,90	21,07	17,02	17,46	24,72	3,43	6,02	7,43	5,64	10,33	11,78	12,69	14,91
TA	200,33	143,92	114,11	118,21	171,44	16,39	36,64	46,66	32,60	63,71	76,86	83,62	99,34
ÜAA	24,30	18,26	14,64	15,14	21,53	2,50	4,90	6,14	4,42	8,41	9,95	10,78	12,76
DAA	17,16	13,10	10,51	10,81	15,26	2,14	3,73	4,61	3,46	6,25	7,22	7,78	9,13
YÜAA	17,49	13,26	10,66	11,00	15,49	2,20	3,78	4,66	3,55	6,41	7,36	7,94	9,33
BEÜAA	22,41	17,42	14,13	14,44	20,28	2,69	5,03	6,10	4,51	8,33	9,67	10,36	12,11
UKÖA	18,32	14,36	11,72	12,22	16,77	2,98	4,57	5,63	4,52	7,57	8,32	9,07	10,49



Sekil 4-7 Pazarlamacı-arama süreleri

4.2 “Haberler” Görüntü Dizisi İçin Bulgular

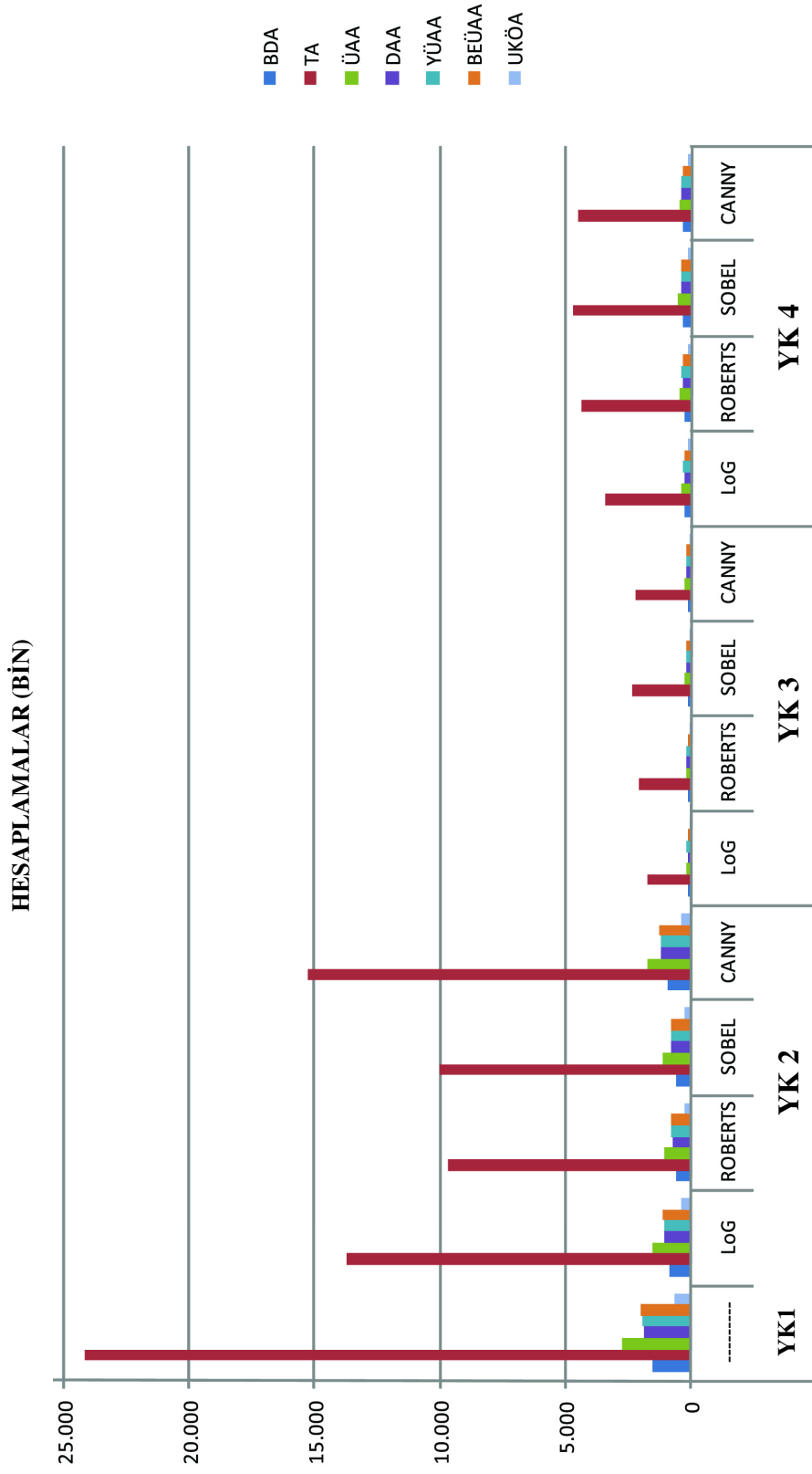


Şekil 4-8 Haberler

4.2.1 “Haberler” görüntü dizisi için hesaplamalar (aday öbek sayısı)

Çizelge 4.9 Haberler-hesaplamalar (aday öbek sayısı)

	HESAPLAMALAR (BIN)												
	YK1	YK2				YK3				YK4			
	-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	1.520	851	605	628	950	136	151	168	167	242	293	314	310
TA	24.188	13.727	9.697	10.034	15.307	1.754	2.097	2.341	2.230	3.425	4.397	4.719	4.528
ÜAA	2.749	1.550	1.094	1.132	1.730	195	234	262	248	382	493	529	505
DAA	1.917	1.078	764	792	1.203	153	177	197	192	285	357	382	371
YÜAA	1.965	1.103	785	813	1.230	171	194	216	212	311	379	406	399
BEÜAA	2.007	1.131	798	825	1.264	133	163	182	170	267	351	378	356
UKÖA	680	375	270	281	418	81	84	94	97	131	146	156	160

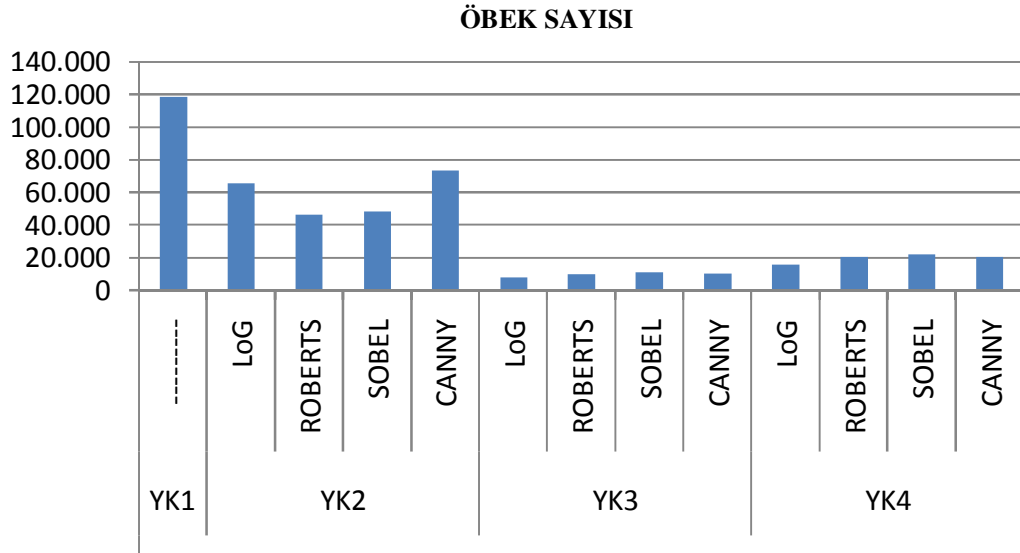


Şekil 4-9 Haberler – hesaplamalar (aday öbek sayısı)

4.2.2 “Haberler” görüntü dizisi için öbek sayısı (ÖS)

Çizelge 4.10 Haberler-öbek sayısı

ÖBEK SAYISI												
YK1	YK2				YK3				YK4			
-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
118.404	65.482	46.160	47.768	73.196	7.843	9.541	10.658	9.966	15.533	20.270	21.834	20.496



Şekil 4-10 Haberler- öbek sayısı

4.2.3 “Haberler” görüntü dizisi için HVOÖ

Çizelge 4.11 Haberler - HVOÖ

	HVOÖ SAYISI															
	YK1				YK2				YK3				YK4			
	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	16.054	8.576	6.661	7.308	9.358	6.400	5.555	6.285	7.297	8.533	7.141	7.604	9.340			
TA	16.238	8.667	6.742	7.388	9.446	6.443	5.605	6.340	7.337	8.614	7.205	7.679	9.413			
ÜAA	16.101	8.611	6.690	7.335	9.392	6.417	5.567	6.300	7.311	8.563	7.158	7.629	9.364			
DAA	16.066	8.582	6.665	7.312	9.364	6.404	5.559	6.289	7.301	8.539	7.145	7.608	9.346			
YÜAA	16.078	8.600	6.684	7.329	9.383	6.414	5.563	6.296	7.308	8.558	7.154	7.625	9.359			
BEÜAA	15.828	8.527	6.621	7.266	9.302	6.356	5.514	6.247	7.244	8.475	7.088	7.558	9.271			
UKÖA	15.802	8.383	6.463	7.116	9.178	6.230	5.380	6.118	7.145	8.355	6.969	7.428	9.161			

HVOÖ SAYISI



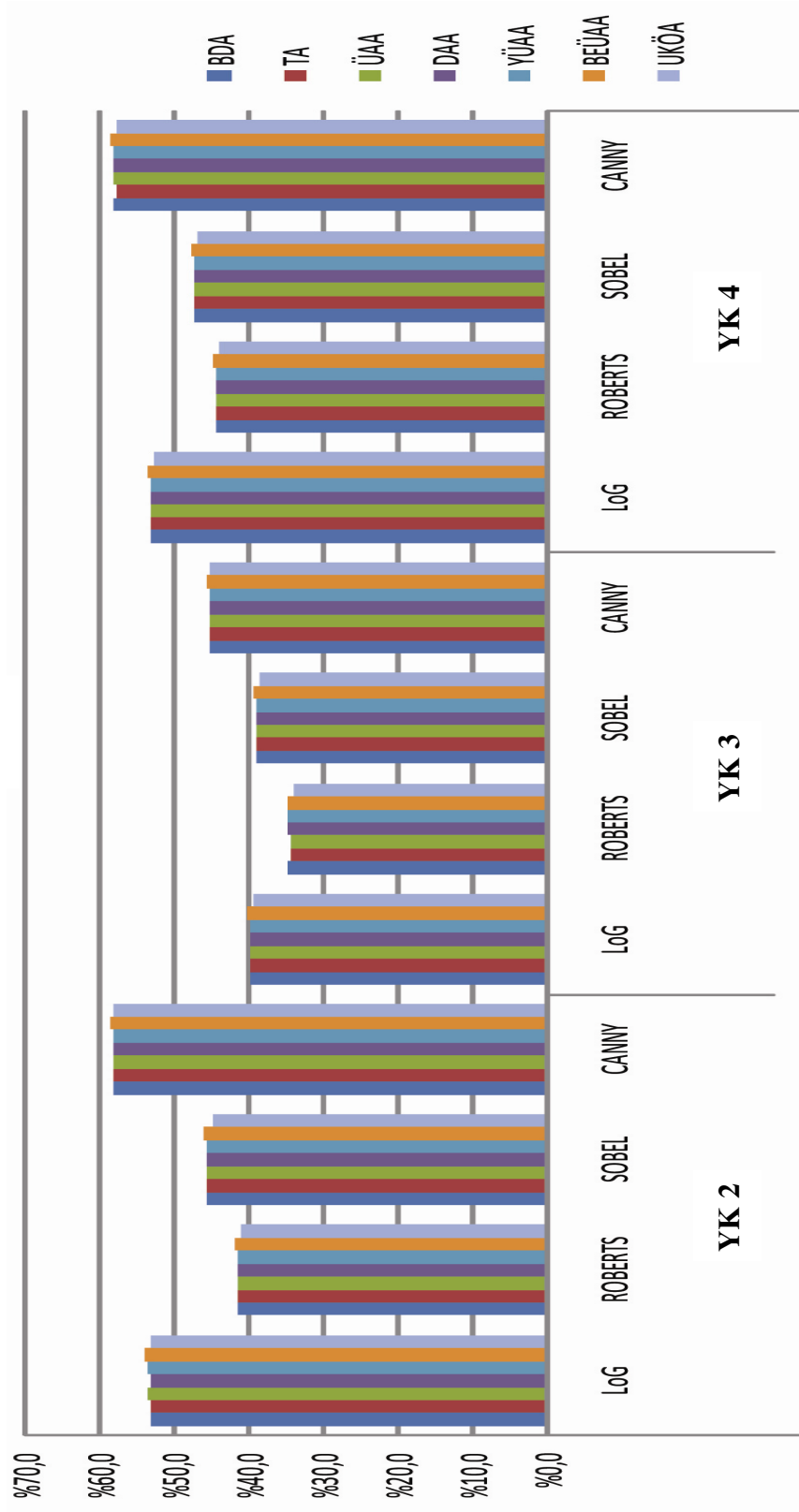
Sekil 4-11 Haberler - HVOÖ

4.2.4 “Haberler” görüntü dizisi için HVBY

Çizelge 4.12 Haberler - HVBY

	HVBY											
	YK2				YK3				YK4			
	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	%53,4	%41,5	%45,5	%58,3	%39,9	%34,6	%39,1	%45,5	%53,2	%44,5	%47,4	%58,2
TA	%53,4	%41,5	%45,5	%58,2	%39,7	%34,5	%39,0	%45,2	%53,0	%44,4	%47,3	%58,0
ÜAA	%53,5	%41,6	%45,6	%58,3	%39,9	%34,6	%39,1	%45,4	%53,2	%44,5	%47,4	%58,2
DAA	%53,4	%41,5	%45,5	%58,3	%39,9	%34,6	%39,1	%45,4	%53,1	%44,5	%47,4	%58,2
YÜAA	%53,5	%41,6	%45,6	%58,4	%39,9	%34,6	%39,2	%45,5	%53,2	%44,5	%47,4	%58,2
BEÜAA	%53,9	%41,8	%45,9	%58,8	%40,2	%34,8	%39,5	%45,8	%53,5	%44,8	%47,8	%58,6
UKÖA	%53,1	%40,9	%45,0	%58,1	%39,4	%34,0	%38,7	%45,2	%52,9	%44,1	%47,0	%58,0

HVBY

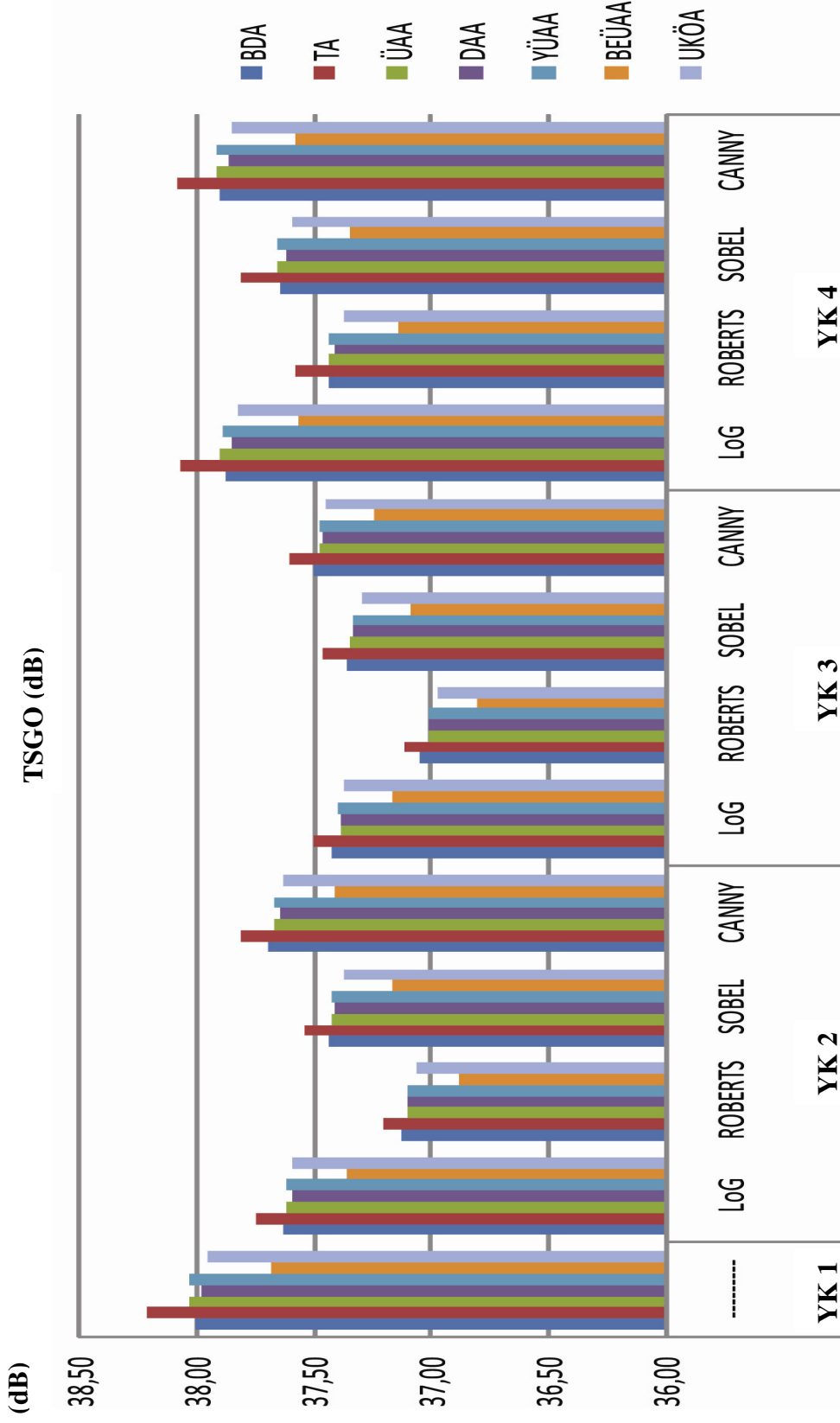


Sekil 4-12 Haberler - HVBY

4.2.5 ‘‘Haberler’’ grnt dizisi iin TSGO (PSNR)

izelge 4.13 Haberler - TSGO

	TSGO (dB)												
	YK1			YK2			YK3			YK4			
	-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	38,01	37,64	37,13	37,44	37,69	37,42	37,04	37,36	37,51	37,88	37,43	37,65	37,90
TA	38,21	37,75	37,21	37,54	37,81	37,51	37,11	37,46	37,61	38,07	37,58	37,82	38,09
AA	38,03	37,61	37,10	37,42	37,67	37,39	37,00	37,34	37,48	37,90	37,44	37,66	37,92
DAA	37,98	37,60	37,10	37,41	37,65	37,38	37,02	37,33	37,47	37,85	37,41	37,62	37,87
YAA	38,03	37,61	37,10	37,42	37,67	37,40	37,01	37,34	37,48	37,90	37,44	37,66	37,92
BEAA	37,68	37,36	36,89	37,17	37,41	37,16	36,81	37,09	37,24	37,57	37,14	37,34	37,59
UKA	37,96	37,59	37,06	37,37	37,64	37,38	36,98	37,29	37,46	37,82	37,38	37,59	37,85

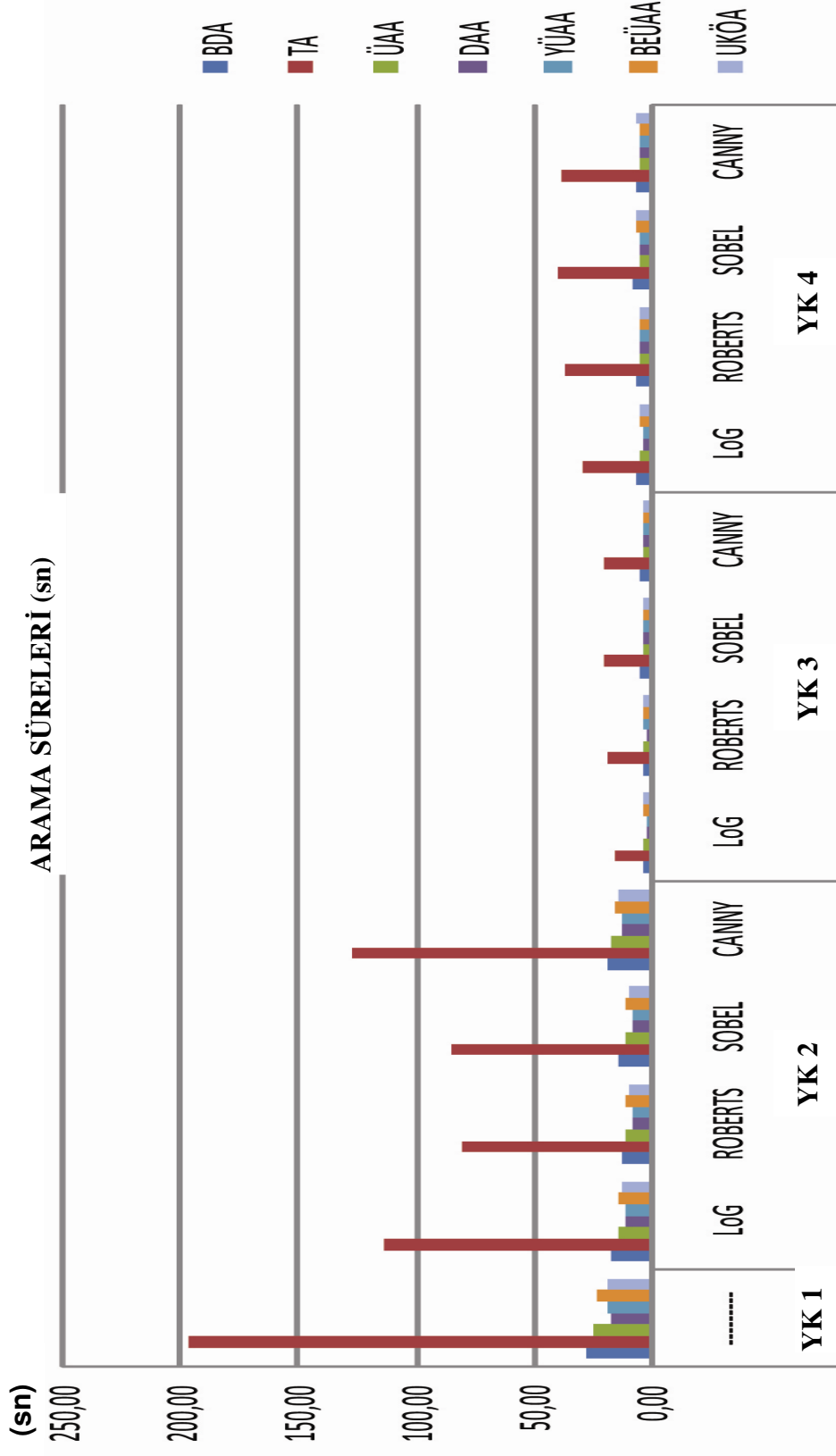


Sekil 4-13 Haberler - TSGO

4.2.6 “Haberler” görüntü dizisi için arama süresi

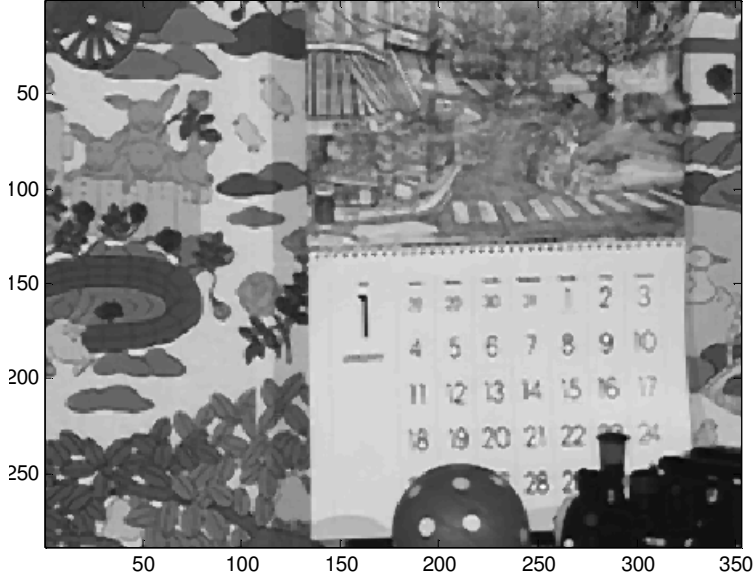
Çizelge 4.14 Haberler – arama süresi

	ARAMA SÜRELERİ (SN)												
	YK1	YK2			YK3			YK4					
	-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	27,58	16,94	12,41	12,99	18,77	3,45	3,71	4,07	3,95	5,49	6,41	6,90	6,67
TA	197,59	112,99	80,23	84,20	126,88	15,01	17,79	20,09	19,00	28,40	36,38	39,56	37,88
ÜAA	24,24	14,21	10,22	10,62	15,92	2,30	2,61	2,88	2,76	3,95	4,89	5,28	5,10
DAA	17,38	10,46	7,62	7,96	11,61	2,11	2,26	2,48	2,40	3,33	3,91	4,18	4,08
YÜAA	17,66	10,39	7,56	7,91	11,61	2,16	2,33	2,55	2,49	3,44	4,01	4,30	4,23
BEÜAA	22,54	13,71	10,02	10,37	15,11	2,38	2,72	2,95	2,73	3,98	4,98	5,31	4,94
UKÖA	18,62	11,54	8,61	9,11	12,92	3,06	3,15	3,50	3,51	4,45	4,98	5,40	5,34



Sekil 4-14 Haberler – arama süresi

4.3 “Mobil” Görüntü Dizisi İçin Bulgular



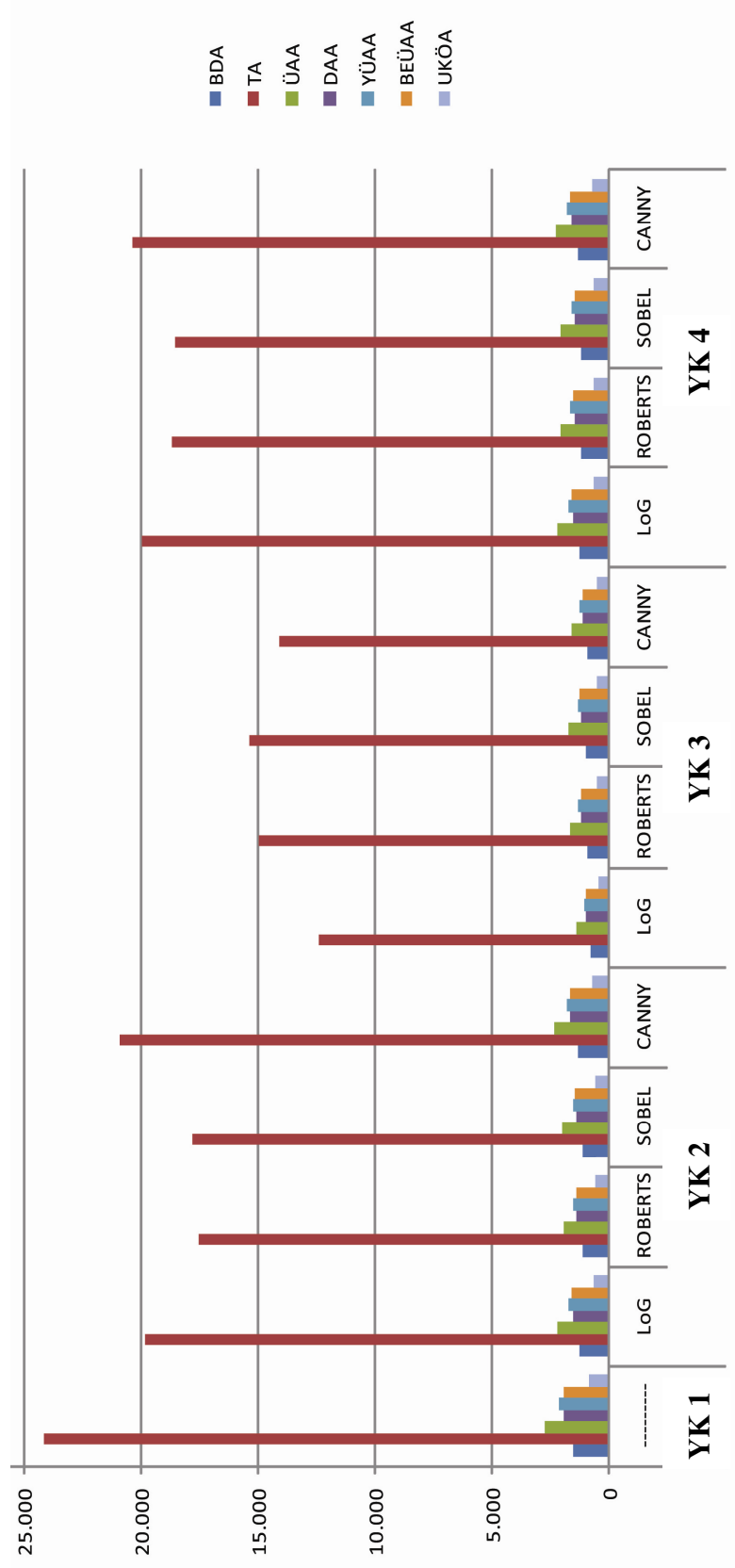
Şekil 4-15 Mobile

4.3.1 “Mobil” görüntü dizisi için hesaplamalar (aday öbek sayısı)

Çizelge 4.15 Mobil-hesaplamalar (aday öbek sayısı)

	HESAPLAMALAR (BİN)												
	YK1	YK2				YK3				YK4			
	-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	1.577	1.283	1.136	1.150	1.353	821	977	1.002	930	1.297	1.211	1.202	1.323
TA	24.188	19.886	17.597	17.834	20.975	12.431	15.017	15.427	14.123	19.967	18.699	18.588	20.379
ÜAA	2.749	2.252	1.992	2.019	2.375	1.408	1.700	1.747	1.599	2.261	2.117	2.105	2.307
DAA	1.927	1.573	1.395	1.414	1.659	991	1.194	1.226	1.124	1.583	1.483	1.474	1.615
YÜAA	2.152	1.758	1.557	1.578	1.854	1.114	1.336	1.372	1.262	1.772	1.657	1.648	1.807
BEÜAA	1.993	1.623	1.433	1.453	1.712	1.009	1.220	1.255	1.148	1.627	1.523	1.514	1.661
UKÖA	860	696	619	628	734	451	535	550	507	706	660	656	720

HESAPLAMALAR (BİN)

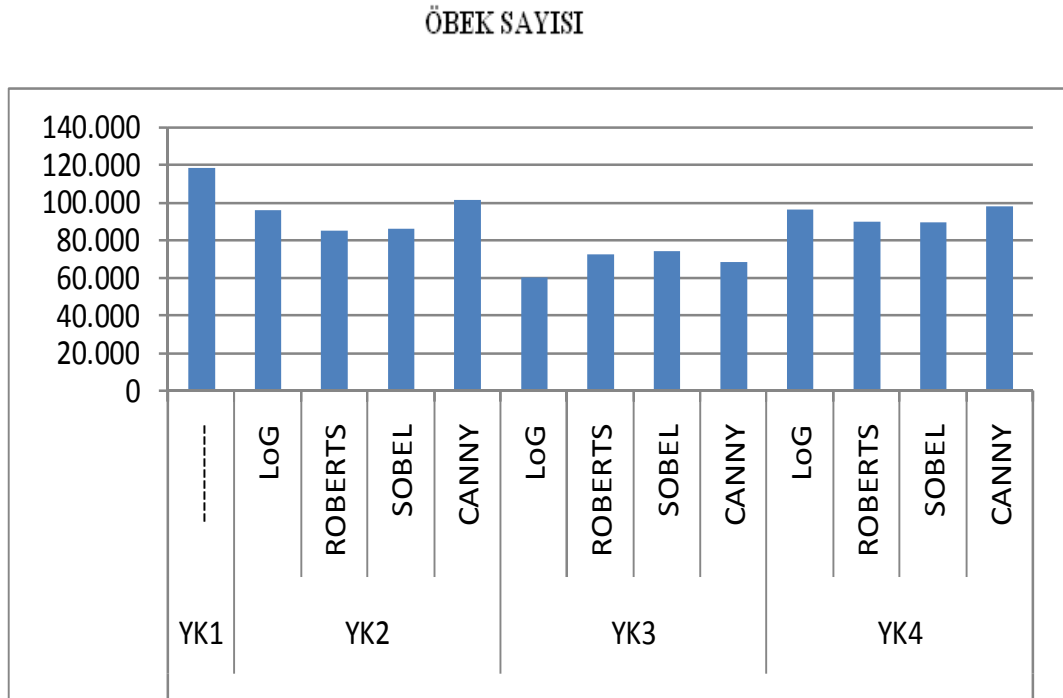


Sekil 4-16 Mobil – hesaplamalar (aday öbek sayısı)

4.3.2 ‘Mobil’ görüntü dizisi için öbek sayısı (ÖS)

Çizelge 4.16 Mobil- öbek sayısı

ÖBEK SAYISI												
YK1	YK2				YK3				YK4			
-----	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
118.404	95.815	84.684	85.931	101.030	59.839	72.208	74.286	68.049	96.229	90.022	89.542	98.143



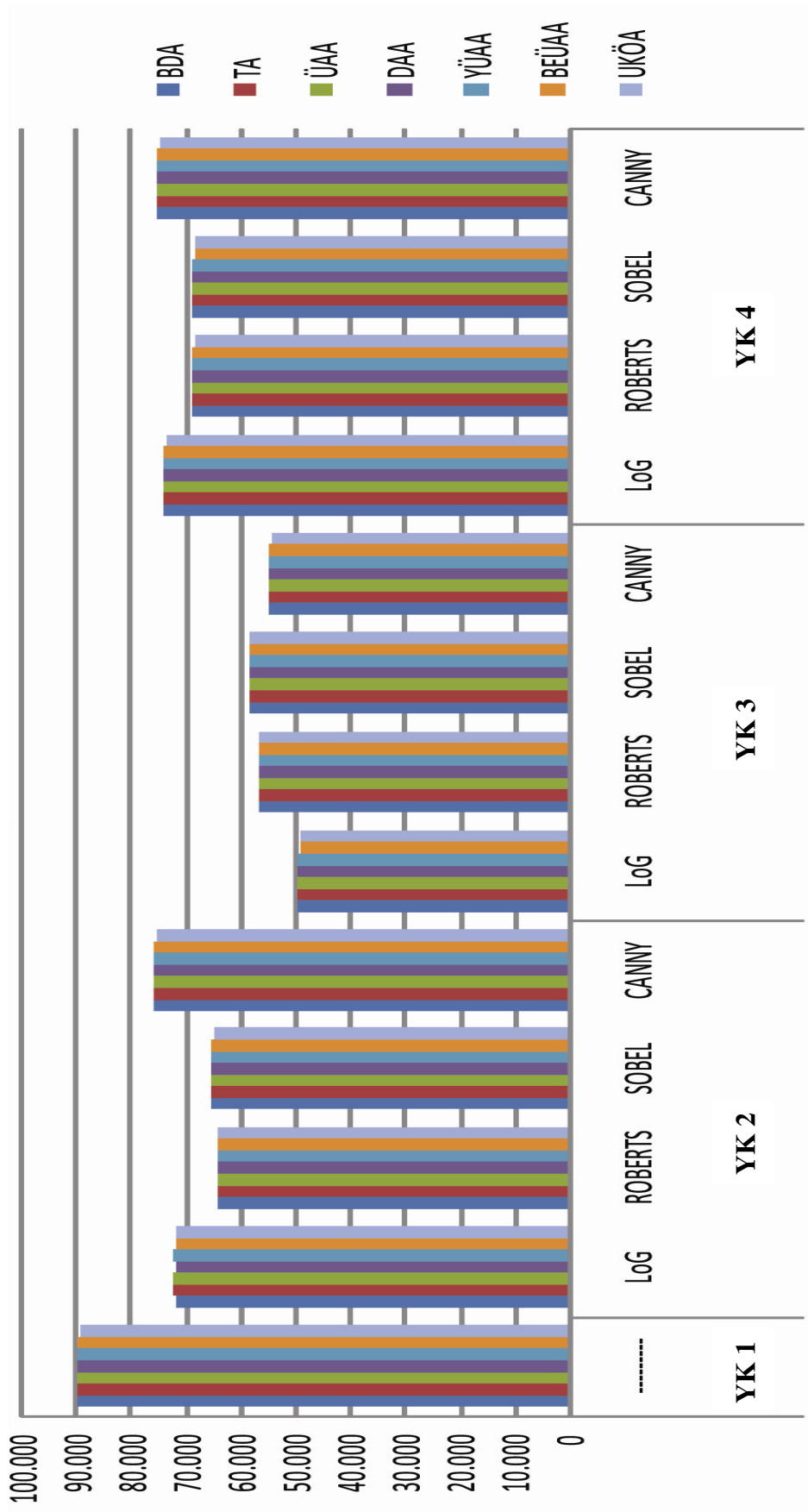
Şekil 4-17 Mobil- öbek sayısı

4.3.3 “Mobil” görüntü dizisi için HVOÖ

Çizelge 4.17 Mobil - HVOÖ

	HVOÖ SAYISI												
	YK1	YK2			YK3			YK4					
		LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	90.051	72.212	64.374	65.423	75.950	49.473	56.831	58.495	54.858	74.253	68.953	68.744	75.416
TA	90.257	72.328	64.496	65.545	76.072	49.537	56.946	58.608	54.942	74.389	69.086	68.875	75.549
ÜAA	90.132	72.270	64.431	65.479	76.010	49.499	56.886	58.545	54.900	74.313	69.013	68.803	75.475
DAA	90.052	72.213	64.375	65.424	75.951	49.474	56.832	58.496	54.859	74.254	68.954	68.745	75.417
YÜAA	90.122	72.266	64.427	65.475	76.006	49.496	56.882	58.541	54.897	74.309	69.009	68.799	75.470
BEÜAA	89.825	72.093	64.255	65.291	75.816	49.366	56.742	58.391	54.770	74.115	68.816	68.600	75.272
UKÖA	89.668	71.886	64.072	65.110	75.603	49.201	56.559	58.223	54.575	73.919	68.629	68.417	75.073

HVOÖ SAYISI



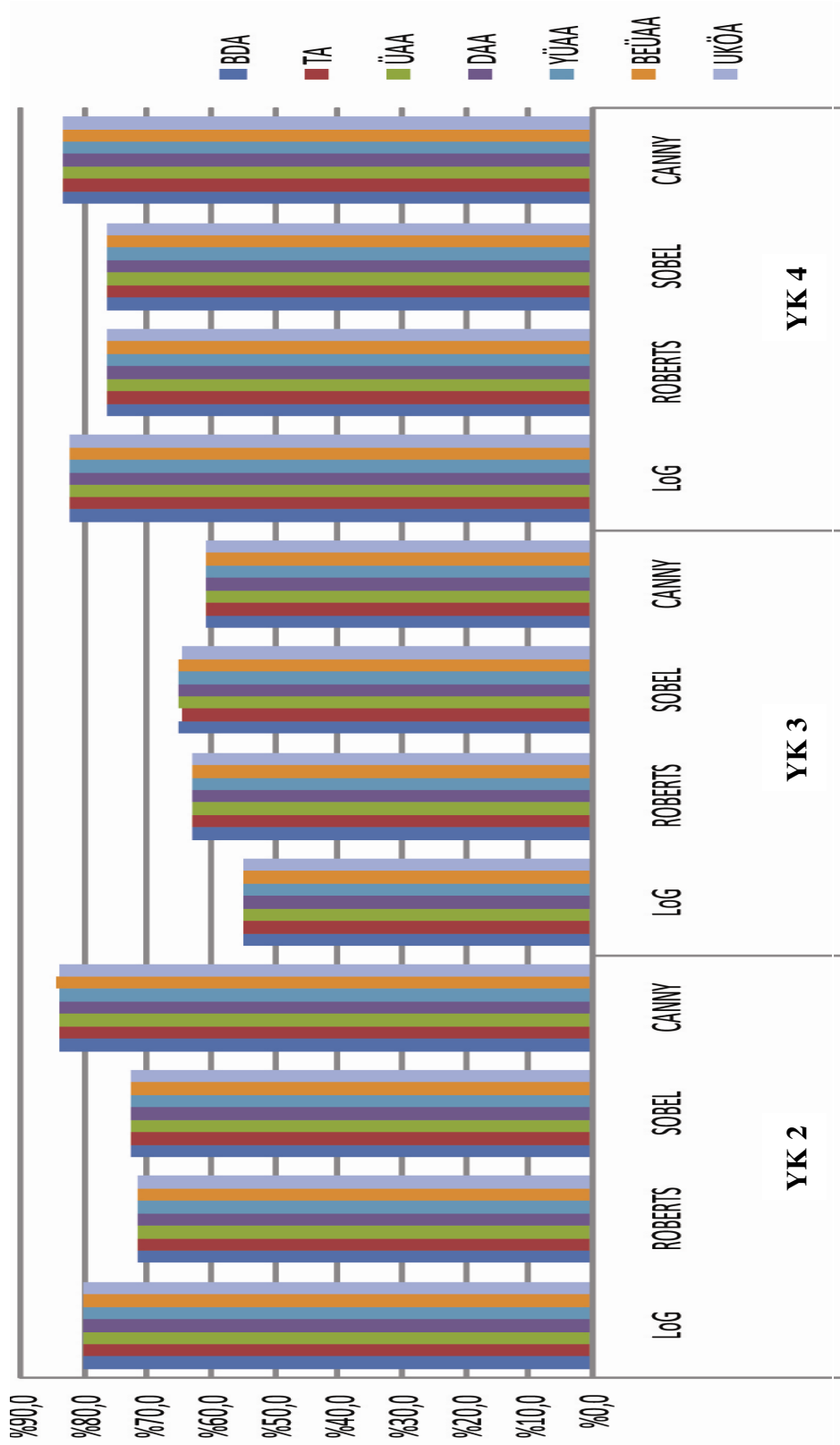
Sekil 4-18 Mobil - HVOÖ

4.3.4 “Mobil” görüntü dizisi için HVBY

Çizelge 4.18 Mobil - HVBY

	HVBY											
	YK2				YK3				YK4			
	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY	LoG	ROBERTS	SOBEL	CANNY
BDA	%80,2	%71,5	%72,7	%84,3	%54,9	%63,1	%65,0	%60,9	%82,5	%76,6	%76,3	%83,7
TA	%80,1	%71,5	%72,6	%84,3	%54,9	%63,1	%64,9	%60,9	%82,4	%76,5	%76,3	%83,7
ÜAA	%80,2	%71,5	%72,6	%84,3	%54,9	%63,1	%65,0	%60,9	%82,4	%76,6	%76,3	%83,7
DAA	%80,2	%71,5	%72,7	%84,3	%54,9	%63,1	%65,0	%60,9	%82,5	%76,6	%76,3	%83,7
YÜAA	%80,2	%71,5	%72,7	%84,3	%54,9	%63,1	%65,0	%60,9	%82,5	%76,6	%76,3	%83,7
BEÜAA	%80,3	%71,5	%72,7	%84,4	%55,0	%63,2	%65,0	%61,0	%82,5	%76,6	%76,4	%83,8
UKÖA	%80,2	%71,5	%72,6	%84,3	%54,9	%63,1	%64,9	%60,9	%82,4	%76,5	%76,3	%83,7

HVBY

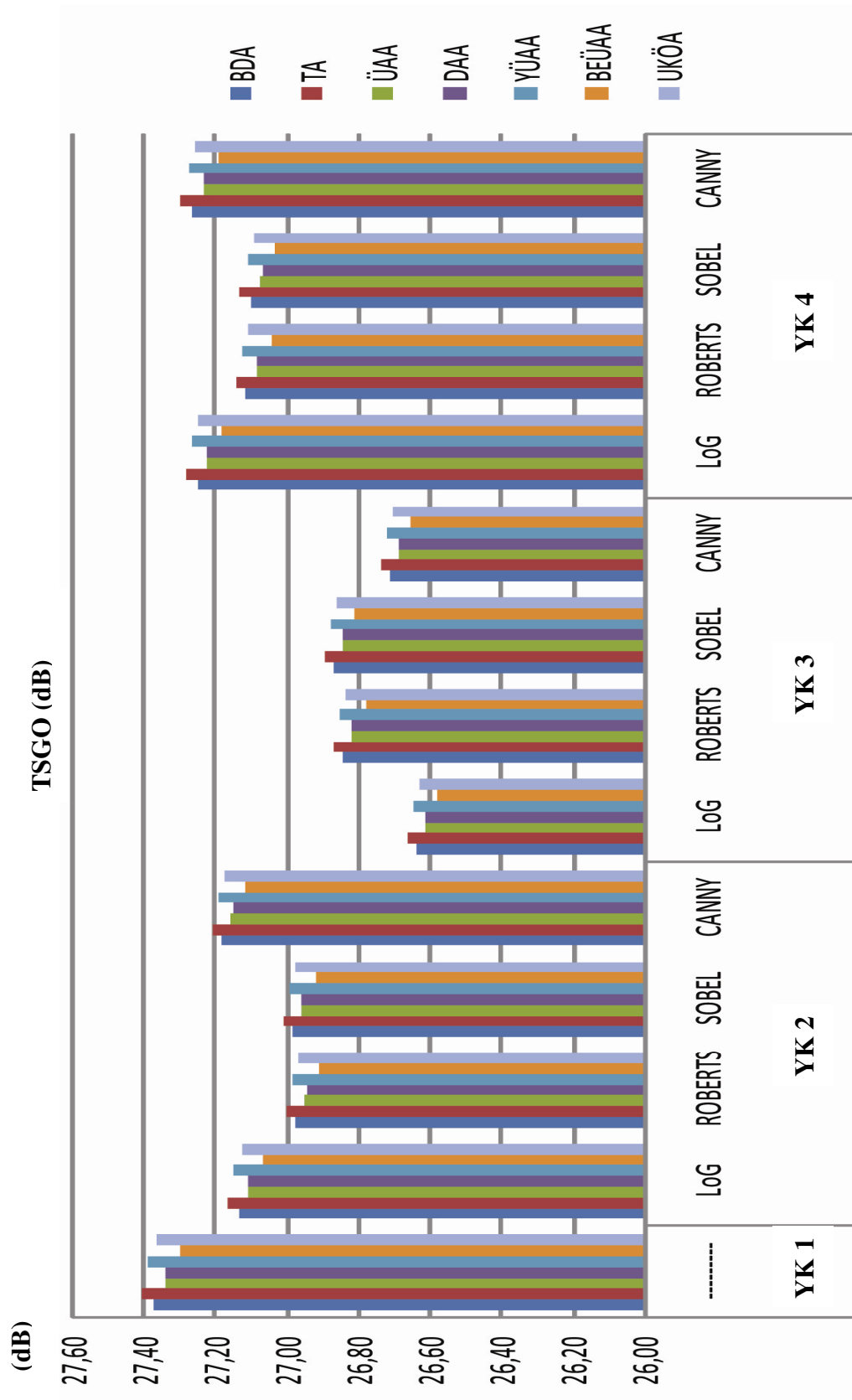


Sekil 4-19 Mobil - HVBY

4.3.5 “Mobil” görüntü dizisi için TSGO (PSNR)

Çizelge 4.19 Mobil - TSGO

	TSGO (dB)												
	YK1			YK2			YK3			YK4			
	LoG	ROBERTS	SOBEL	LoG	ROBERTS	SOBEL	LoG	ROBERTS	SOBEL	LoG	ROBERTS	SOBEL	
BDA	27,37	27,14	26,97	26,99	27,18	27,18	26,64	26,84	26,87	26,71	27,26	27,10	27,26
TA	27,41	27,17	27,00	27,01	27,21	27,21	26,66	26,87	26,90	26,74	27,29	27,15	27,30
ÜAA	27,35	27,11	26,95	26,96	27,16	27,16	26,62	26,82	26,85	26,69	27,23	27,09	27,24
DAA	27,34	27,11	26,95	26,96	27,15	27,15	26,61	26,82	26,85	26,69	27,23	27,09	27,23
YÜAA	27,39	27,15	26,99	27,00	27,20	27,20	26,65	26,85	26,88	26,72	27,27	27,13	27,28
BEÜAA	27,30	27,07	26,91	26,92	27,12	27,12	26,58	26,78	26,81	26,65	27,18	27,05	27,19
UKÖA	27,37	27,13	26,97	26,98	27,18	27,18	26,63	26,84	26,86	26,70	27,25	27,11	27,26

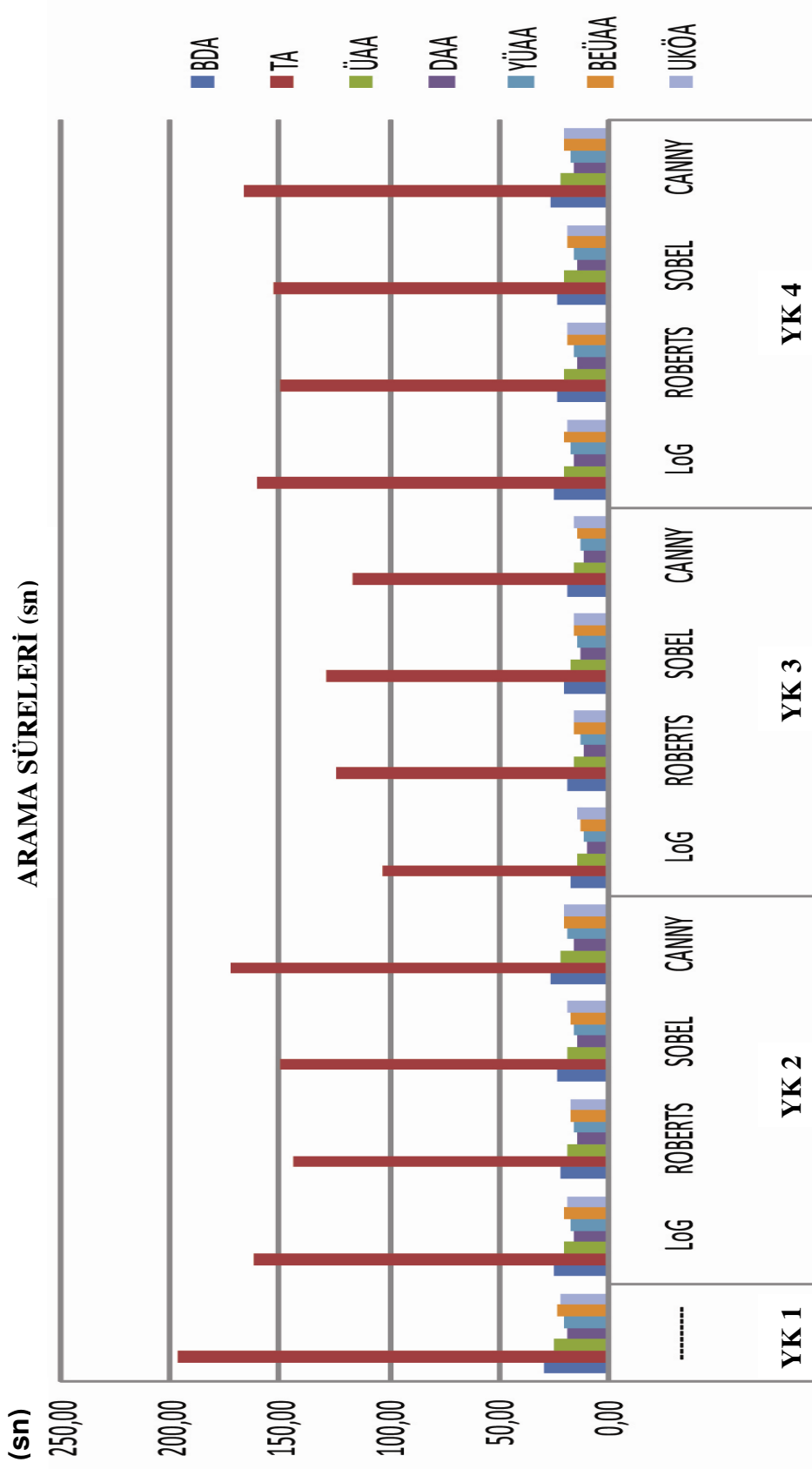


Sekil 4-20 Mobil - TSGO

4.3.6 “Mobil” görüntü dizisi için arama süresi

Çizelge 4.20 Mobil – arama süresi

	ARAMA SÜRELERİ (SN)											
	YK1			YK2			YK3			YK4		
	LoG	ROBERTS	SOBEL	LoG	ROBERTS	SOBEL	LoG	ROBERTS	SOBEL	LoG	ROBERTS	SOBEL
BDA	28,43	24,59	21,81	22,31	25,93	16,15	18,80	19,81	18,19	24,59	22,97	23,05
TA	196,53	162,32	144,57	149,50	173,34	102,29	123,61	128,59	117,19	160,74	150,66	152,50
ÜAA	24,34	20,36	18,10	18,55	21,72	13,04	15,59	16,21	14,87	20,32	19,02	19,25
DAA	17,49	14,75	13,10	13,47	15,58	9,55	11,29	11,73	10,81	14,76	13,80	13,88
YÜAA	19,76	16,56	14,74	15,13	17,66	10,78	12,72	13,25	12,26	16,59	15,50	15,71
BEÜAA	22,38	19,03	16,94	17,27	20,02	12,42	14,65	15,06	13,92	19,06	17,85	17,90
UKÖA	21,88	18,86	16,95	17,57	19,89	12,90	14,85	15,50	14,61	18,78	17,67	18,08



Şekil 4-21 Mobil – arama süresi

4.4 Literatür Karşılaştırması

4.4.1 “Pazarlamacı” görüntü dizisi için TSGO (PSNR) karşılaştırması

Çizelge 4.21 Pazarlamacı- TSGO, Nasır (2007)

	Pazarlamacı					
	5*eşik		10*eşik		15*eşik	
	TA	BDA	TA	BDA	TA	BDA
TSGO (dB)	40,57	39,52	40,33	39,48	40,09	39,42

Çizelge 4.22 Pazarlamacı- TSGO, kenar algılama

	Pazarlamacı					
	YK1		YK4-Sobel		YK4-Canny	
	TA	BDA	TA	BDA	TA	BDA
TSGO (dB)	39,63	39,54	38,66	38,61	39,37	39,29

4.4.2 “Pazarlamacı” görüntü dizisi için arama süresi karşılaştırması

Çizelge 4.23 Pazarlamacı- arama süresi, Nasır (2007)

	Pazarlamacı					
	5*eşik		10*eşik		15*eşik	
	TA	BDA	TA	BDA	TA	BDA
Arama Süresi (sn)	189,08	74,14	87,39	64,17	57,5	60,12

Çizelge 4.24 Pazarlamacı- arama süresi, kenar algılama

	Pazarlamacı					
	YK1		YK4-Sobel		YK4-Canny	
	TA	BDA	TA	BDA	TA	BDA
Arama Süresi (sn)	222,57	33,82	49,41	8,82	69,60	12,10

4.4.3 “Haberler” görüntü dizisi için TSGO (PSNR) karşılaştırması

Çizelge 4.25 Haberler- TSGO, Nasır (2007) yöntemi

	Haberler					
	5*eşik		10*eşik		15*eşik	
	TA	BDA	TA	BDA	TA	BDA
TSGO (dB)	42,41	40,32	42,34	40,31	42,16	40,28

Çizelge 4.26 Haberler- TSGO, kenar algılama temelli yöntem

	Haberler					
	YK1		YK4-Sobel		YK4-Canny	
	TA	BDA	TA	BDA	TA	BDA
TSGO (dB)	40,54	40,25	39,12	38,95	40,33	40,07

4.4.4 “Haberler” görüntü dizisi için arama süresi karşılaştırması

Çizelge 4.27 Haberler– arama süresi, Nasır (2007) yöntemi

	Haberler					
	5*eşik		10*eşik		15*eşik	
	TA	BDA	TA	BDA	TA	BDA
Arama Süresi (sn)	114,6	65,93	92,41	63,69	74,08	61,68

Çizelge 4.28 Haberler– arama süresi, kenar algılama temelli yöntem

	Haberler					
	YK1		YK4-Sobel		YK4-Canny	
	TA	BDA	TA	BDA	TA	BDA
Arama Süresi (sn)	225,36	34,02	22,40	4,75	30,18	6,16

5. TARTIŞMA VE SONUÇ

Bu yüksek lisans tezinde hareket kestirimi algoritmalarından öbek eşleme temelli hareket kestirimi için tam arama ve hızlı öbek eşleme kestirim teknikleri kullanılarak iki boyutlu hareket kestirimi ve kenar algılama algoritmaları incelenmiştir. Öbek eşleme temelli hareket kestirim algoritmalarından baklava desenli arama, tam arama, üç adımda arama, dört adımda arama, yeni üç adımda arama, basit ve etkili üç adımda arama, uyarlanırlı kök örüntü arama teknikleri üzerine MATLAB 7.5 programı kullanılarak program yazılmış, sonuçları değişik test görüntü örnekleri üzerinde incelenmiştir. Bu arama tekniklerden elde edilen sonuçları arama maliyetinin iyileştirilmesi açısından ele alındığında kenar algılama algoritmalarına dayalı hareket kestirim yöntemi önerilmiştir. Önerilen yöntem, görüntünün birbirlerinden bir kenar ile ayrılmış bölgelerden oluşması bilgisine dayanmaktadır. Görüntüyü 3 bölgede inceleyebiliriz. Bunlar:

- **Geri Plan Bölgesi**

Kameranın sabit olduğu bir görüntü dizisinde, hareketin olması için hareket eden bir nesne gerekir. Geri plan tek düze, kenar oluşturacak kadar renk değişiminin olmadığı fon diyebileceğimiz bölgedir ve nesne (obje) bulunmaz. Önerilen yöntemle geri plan bölgesini elemeye yönelik çalışarak hareketli nesne olmayan bölge için gereksiz arama yapılmamış olur. Bu işlem zamandan önemli ölçüde tasarruf sağlar.

- **Sabit Nesneler Bölgesi**

Kameranın sabit olduğu bir görüntü dizisinde hareket etmeyen nesnelere (masa, vazo, telefon, bina, ağaç, vb) konumu ardışık çerçevelerde aynı kalır. Ardışık çerçevelerin kenar haritalarının “VE kapısından” geçirilmesi sonucu sabit nesnelere bölgesi elde edilir. “1” ve “0” lardan oluşmuş iki kenar haritasının birbiri ile çarpımı, bu kenar haritalarının “VE kapısından” geçirilmesi ile aynıdır. Bu sabit nesnelere her iki çerçevede aynı kaldığından (hareket olmadığından) burada arama yapılmasına gerek yoktur. YK3 ve YK4 için hareketsiz nesnelere olduğu bölge işlem dışına itilir.

- **Hareketli Nesneler Bölgesi**

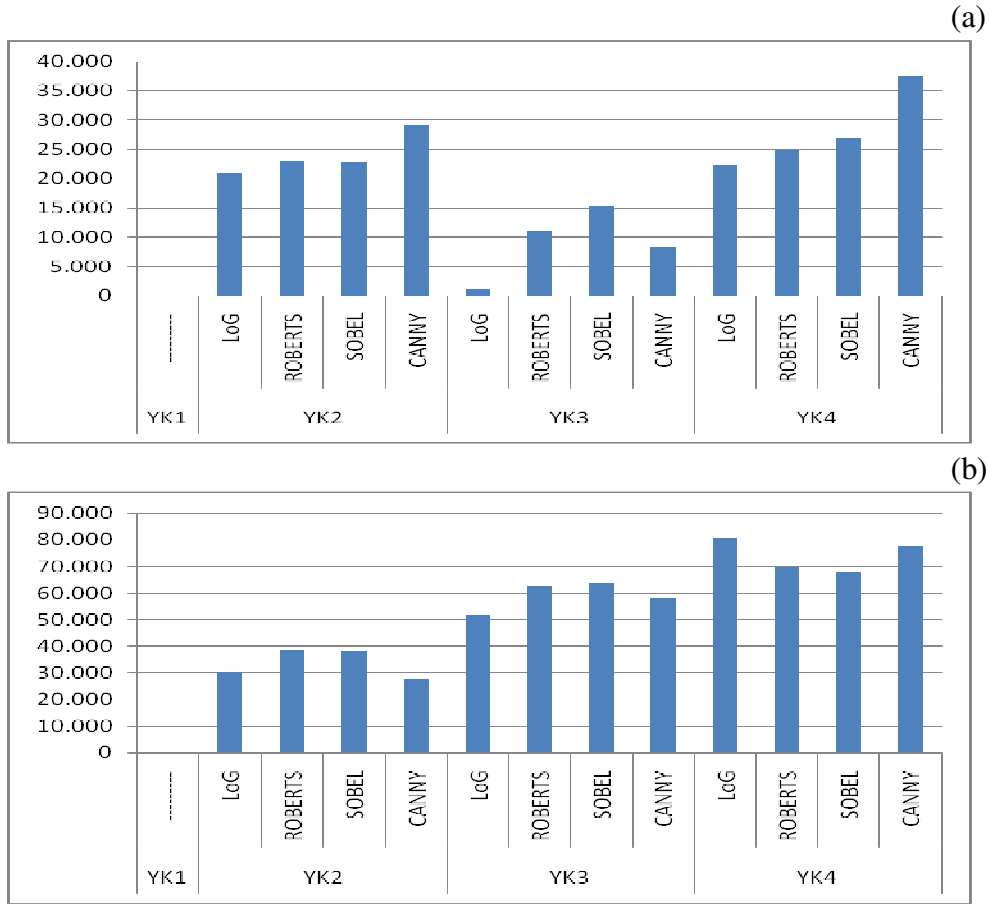
Kameranın sabit olduğu bir görüntü dizisinde hareketli nesne bölgesi, hareket eden nesnenin olduğu bölgedir.

Geri plan ve sabit nesnelere elendiği önerilen bu yöntemde, eleme için belirlenen eşik değeri, üzerinde çalışılan öbekte arama tekniklerinin uygulanıp uygulanmayacağını belirler. Canny ve LoG kenar algılama algoritmalarının kenar haritasında geri plan bölgesinde yüksek düzeyde kenar algılaması olduğu için eşik değeri 10 olarak belirlenmiştir. Robert ve Sobel için ise bu eşik değeri 2 olarak belirlenmiştir.

- Bulgular bölümünde (4. bölüm) hesaplamalar, öbek sayısı ve arama süreleri grafiklerine baktığımızda, bu üç grafik birbiri ile ilintilidir. Bu ilinti her üç test görüntü dizisi için de geçerlidir. Çünkü üzerinde çalışılan öbeğe ilişkin arama tekniklerinin uygulanıp uygulanmayacağı yani o öbekte hareketli nesne olup olmadığının sorgulandığı kenar haritası incelemesi arama süresini, hesaplama sayısını ilintili olarak etkiler.

5.1 300 Çerçevesiz Görüntü Dizileri İçin Sonuçlar

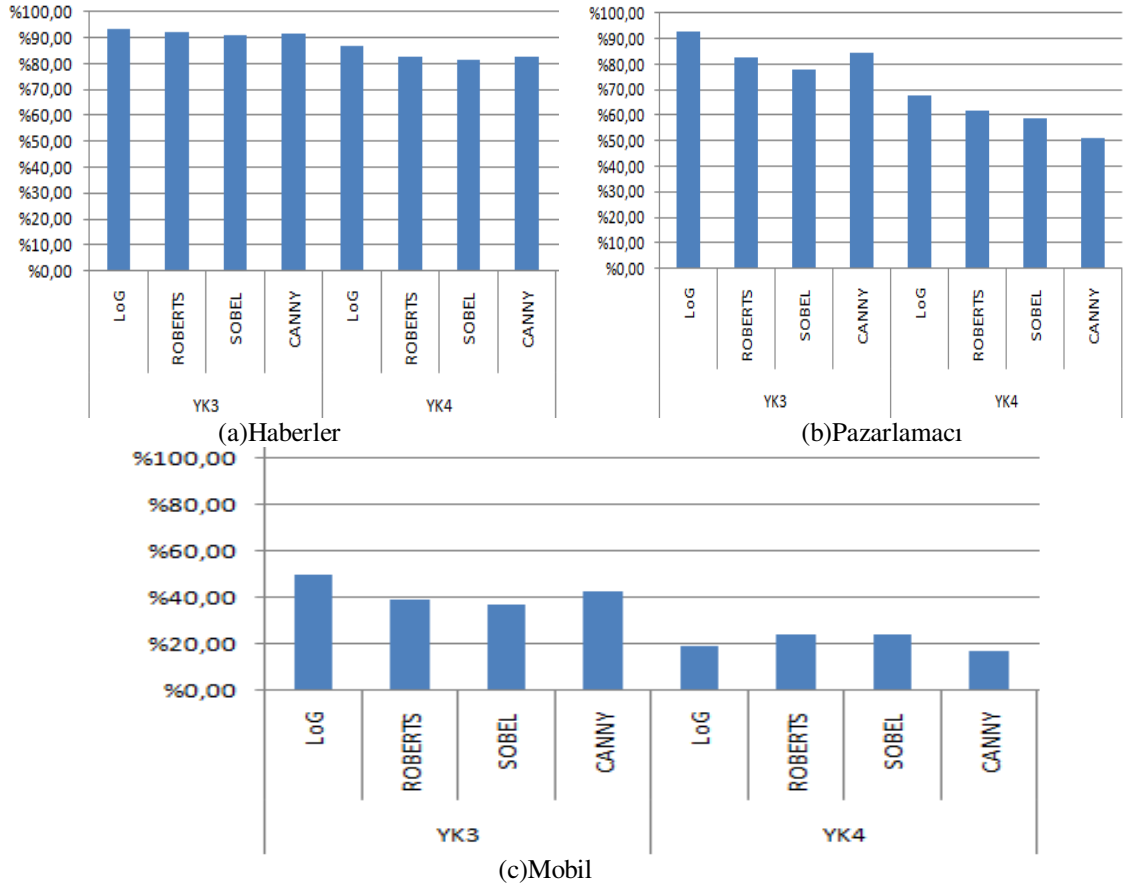
- “Haberler” görüntü dizisinde hesaplanan öbek sayısı diğer iki görüntü dizisinde hesaplanan öbek sayısından belirgin şekilde daha azdır. Şekil 5.1.a’da “Pazarlamacı” görüntü dizisi, Şekil 5.1.b’de ise “Mobil” görüntü dizisi için hesaplanan öbek sayılarının “Haberler” görüntü dizisi için hesaplanan öbek sayısından ne kadar fazla olduğu görülmektedir.



Şekil 5-1.a. Öbek sayısı farkı (pazarlamacı-haberler),
b. Öbek sayısı farkı (mobil-haberler)

- Bunun nedeni geri planın sade olmasıdır. Özellikle “Mobil” görüntü dizisinde geri plandaki fon karmaşıktır. Bu karmaşık geri plan, kenar haritasında yüksek düzeyde nesne olmayan objeleri nesne gibi algılatmaktadır. Kamera hareketi mevcuttur. Kamera hareketi, geri plandaki fonun da hareketli olarak algılanmasına neden olduğu için hareket vektörü bulma yüzdesini düşürdüğü gözlemlenmiştir. “Mobil” görüntü dizisinde ise kamera hareketli olmasına karşın bu karmaşık geri plan nedeni ile bulgulardaki “öbek sayısı” yüksek düzeyde çıkmıştır. Öbek sayısının yüksek çıkması ile TSGO değerini birinci yöntemdeki TSGO değerine yaklaştırır.
- “Mobil” görüntüsü dizisinde sabit nesne bölgesindeki eleme (YK3 ve YK4 ile yapılır) diğer iki görüntü dizisine göre daha düşüktür. Şekil 5.2’de YK3 ve YK4

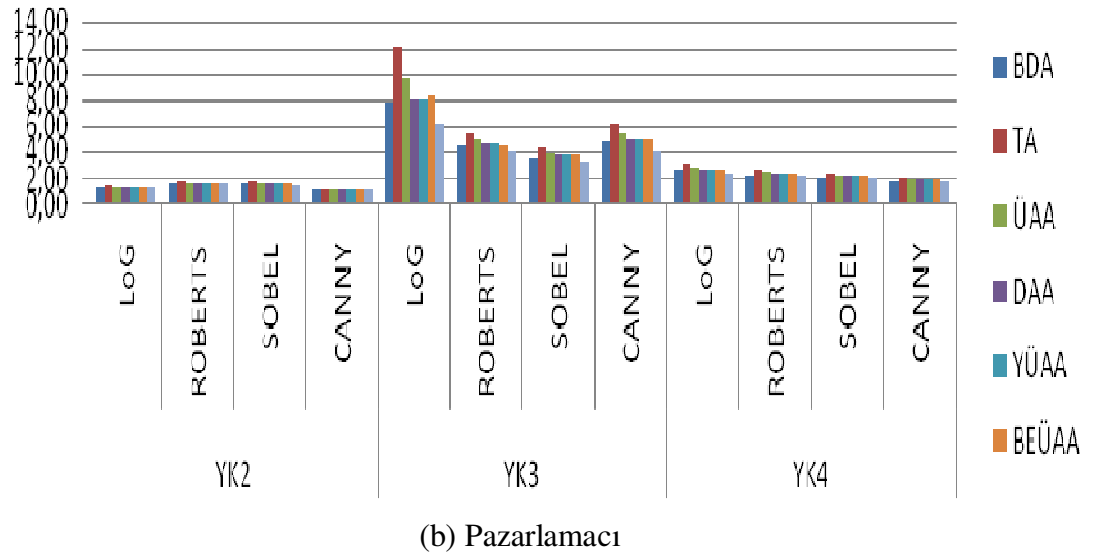
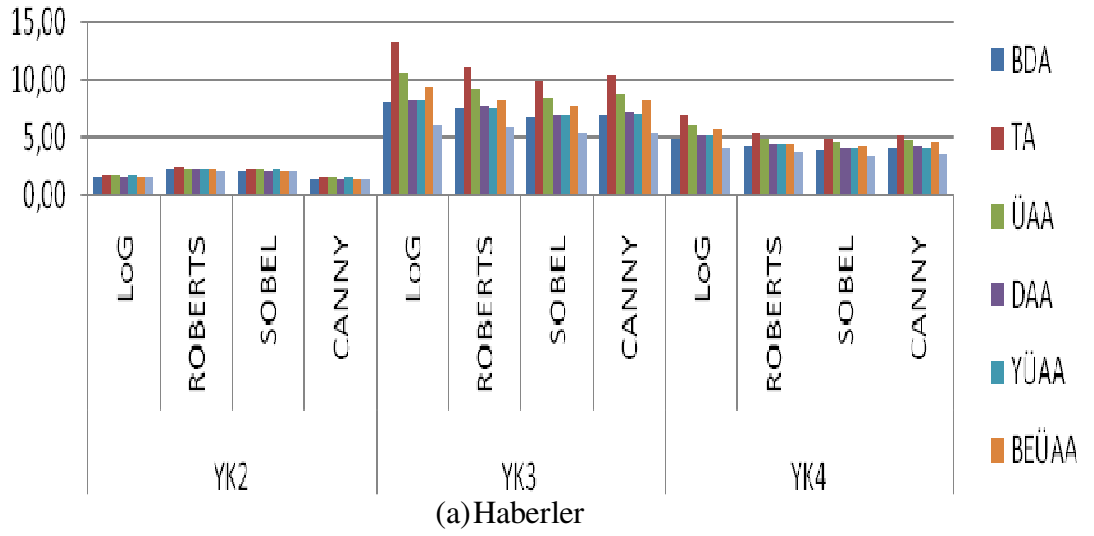
için 118.404 öbeğin yüzde kaçının elendiğini ifade eden grafikler görülmektedir. Burada kameranın sabit olmamasının rolü çok büyüktür. Hareketli kamerada sabit nesnelere konumu da değiştiği için “VE kapısından” geçen kenar haritalarında eleme, sabit kamera ile elde edilen görüntülerdeki elemeye göre çok düşüktür.

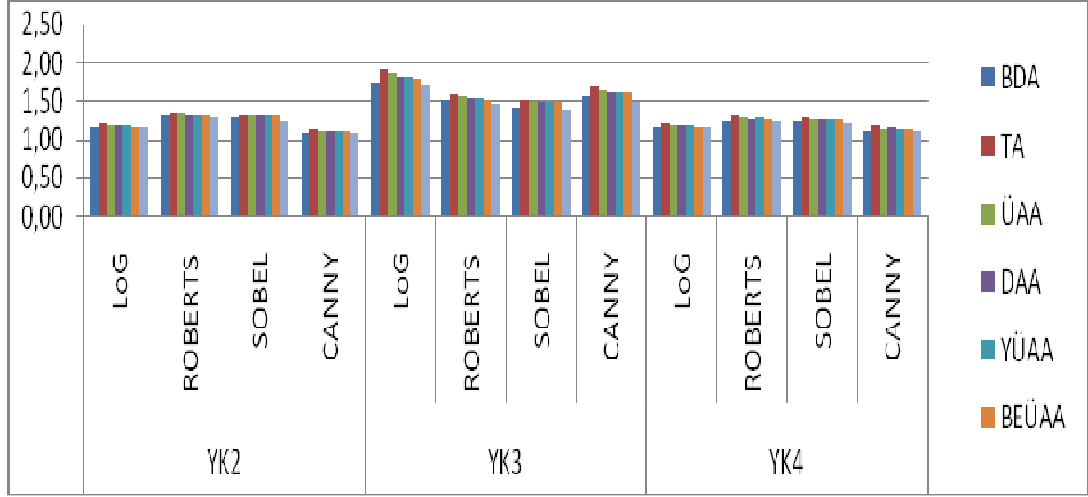


Şekil 5-2 Öbek eleme oranı

- İkinci, üçüncü ve dördüncü yöntemler için arama süresi, eleme edilen öbekler nedeni ile birinci yöntemden daha düşüktür.
- Kenar algılama temelli hareket kestiriminde ise LoG ve Canny kenar algılama algoritmaları Sobel ve Robert kenar algılama algoritmalarına göre daha fazla detay verdiği için, arama algoritmalarının uygulandığı “öbek sayısı, hesaplamalar, arama süreleri” daha fazladır.

- TSGO değerlerinin karşılaştırılması açısından Canny>LoG>Sobel>Robert ve YK1>YK4>YK2>YK3 olur.
- Bulgular bölümünde (4. bölüm) arama süreleri verilerini incelediğimizde, Şekil 5.3'deki grafikler elde edilir. Bu grafikler, birinci yöntemle göre diğer yöntemlerin arama süresi açısından kaç “kat” hızlı arama yaptığı bilgisini verir.





(c) Mobil

Şekil 5-3 Arama süresi hızlanması

- YK4; YK2'yi TSGO ve arama süreleri değerleri açısından iyileştirmiştir.
- Sabit bölgenin elenmediği YK2'de; Sobel kenar algılaması ile elde edilen TSGO değeri, Canny kenar algılaması ile elde edilen TSGO değerinden bu üç görüntü dizisi için 0.19 dB - 0.27 dB arasında daha düşüktür. Bununla birlikte arama süreleri açısından daha iyi sonuç vermektedir.
- Sabit bölgelerin elendiği YK3'de; Sobel kenar algılaması ile elde edilen TSGO değeri, Canny kenar algılaması ile elde edilen TSGO değerinden pazarlamacı ve haberler dizisi için 0.13 dB - 0.17 dB arasında daha düşük, mobil dizisi için 0.16 dB daha yüksektir.
- Sabit bölgenin elendiği YK4'de; Sobel kenar algılaması ile elde edilen TSGO değeri, Canny kenar algılaması ile elde edilen TSGO değerinden bu üç görüntü dizisi için 0.15 dB - 0.27 dB arasında daha düşüktür.
- Canny ve LoG kenar algılama algoritmalarının Robert ve Sobel kenar algılama algoritmalarına göre daha karmaşık olması nedeniyle ve Sobel kenar algılama algoritması ile elde edilen TSGO değerinin Robert kenar algılama algoritması ile

elde edilen TSGO değerinden iyi olması nedeni ile Sobelin bu tez çalışmasında üstünlüğü vardır.

- YK4’de Sobel için elde edilen TSGO değerleri, YK3’de Canny için elde edilen TSGO değerinden daha yüksektir. YK2’de Canny için elde edilen TSGO değerinden 0.01-0.18 dB arasında daha düşük olmasına karşın arama süreleri açısından net bir şekilde üstündür.
- Arama algoritmalarını ikinci, üçüncü ve dördüncü yöntemlere göre incelediğimizde, dört adımda arama ve yeni üç adımda arama teknikleri arama süresi ve TSGO değeri açısından birbirine yakın ve diğer arama algoritmalarına göre bu iki değişkeni (arama süresi ve TSGO) daha uygundur.
- YK4’e göre Sobel kenar algılama algoritması ile kenar haritası çıkmış görüntü dizilerin hareket kestiriminde dört adımda arama veya yeni üç adımda arama tekniklerine göre elde edilen sonuçlar hareket kestirim algoritmalarına arama süresi açısından hareketli kamerada 1.3 kat, sabit kamerada 2.2 ile 4.1 kat arasında hızlanma sağlamıştır.
- İkinci, üçüncü ve dördüncü yöntemlerde kenar bölgeleri içine alan öbekler için hareket kestiriminin yapılması, yüksek piksel değerli ve yer değişiminin daha fazla olduğu bölgelerde hareket kestirimi yapıldığının göstergesi olarak aşağıdaki örneği verebiliriz.

Örnek olarak, “Çizelge 4.13 Haberler-TSGO” ve “Çizelge 4.12 Haberler – HVBY” çizelgelerine baktığımızda, Canny kenar algılama algoritmasına göre ortalama %45 HVBY iken TSGO değeri 37.48 dB (YÜAA) ile birinci yöntemdeki 38.03 dB (YÜAA) değerine 0.55 dB düşüktür. Çizelge 4.10 ve Çizelge 4.11 den Canny kenar algılamasına göre üçüncü yöntemde; 9.966 öbekten 7.308 öbekte hareket vektörü tespit edilmiş ve birinci yöntemde 16.078

hareket vektörünün %45 i ile 0.55 dB düşük değer ile hareket kestiriminde bulunulmuştur. Bu örnekte de görüleceği gibi, Gürültü olarak kabul edilen, arka plandaki küçük renk değişimlerinden kaynaklanan, kenar içermeyen yerlerdeki hareket vektörlerinin TSGO değerine etkisinin kenar bölgelerin TSGO değerine etkisine görece daha düşük olduğunu sonucuna varılır.

5.2 100 Çerçevelik Görüntü Dizileri İçin Sonuçlar

Bölüm 4.4 literatür karşılaştırmasında elde edilen sonuçlara baktığımızda,

- Nasır (2007) yöntemine dayalı elde edilen çerçeveleri temel alan yöntemde BDA için 5, 10 ve 15 ile genişletmenin TSGO değerine etkisi 0.04-0.10 dB arasındadır. Bu nedenle Nasır (2007) metodu ile elde edilen sonuçları 15 ile genişletilmiş eşik seviyesine göre karşılaştırmak daha uygun olacaktır. Çünkü TSGO değerinde 0.1 dB kötüleşmeye karşın arama süresi bu genişletilmiş eşik seviyesinde düşüktür.

Pazarlamacı görüntü dizisi için;

- Nasır (2007) temelli yöntemdeki BDA, kenar algılama uygulanmayan YK1'deki BDA'ya göre yaklaşık 1.8 kat daha yavaştır (60.12 sn/33.82 sn).
- Nasır (2007) temelli yöntemdeki BDA, kenar algılama uygulanmayan YK4 Sobel'deki BDA'ya göre yaklaşık 6.8 kat daha yavaştır (60.12 sn/8.82 sn).
- Üzerinde kenar algılama veya Otsu (1979) eşik seviyesi belirleme algoritmalarının uygulanmadığı YK1 temel alınarak TSGO değerleri karşılaştırıldığında; Nasır (2007) metodunun kullanıldığı algoritma 0.46 dB (TA için 39,63 dB'den 40.09 dB'ye) artış ve 0.12 dB (BDA için 39,54 dB'den 39.42

dB ye) azalış olarak etki etmiştir. YK4 Sobel kenar algılama algoritması 0.97 dB (TA için 39,63 dB'den 38.66 dB'ye) azalış ve 0.93 dB (BDA için 39,54 dB'den 38.61 dB ye) azalış olarak etki etmiştir.

- Nasır (2007) temelli yöntemdeki BDA, kenar algılama uygulanmayan YK4 Canny'deki BDA'ya göre yaklaşık 5 kat daha yavaştır (60.12 sn/12.10 sn). YK4 Canny kenar algılama algoritması 0.26 dB (TA için 39,63 dB'den 39.37 dB'ye) azalış ve 0.26 dB (BDA için 39,54 dB'den 39.29 dB ye) azalış olarak etki etmiştir.

Haberler görüntü dizisi için;

- Nasır (2007) temelli yöntemdeki BDA, kenar algılama uygulanmayan YK1'deki BDA'ya göre yaklaşık 1.8 kat daha yavaştır (61.68 sn/34.02 sn).
- Nasır (2007) temelli yöntemdeki BDA, kenar algılama uygulanmayan YK4 Sobel'deki BDA'ya göre yaklaşık 13 kat daha yavaştır (61.68 sn/4.75 sn).
- YK1 temel alınarak TSGO değerleri karşılaştırıldığında; Nasır (2007) metodunun kullanıldığı algoritma 1.62 dB (TA için 40,54 dB'den 42.16 dB'ye) artış ve 0.03 dB (BDA için 40,25 dB'den 40.28 dB ye) azalış olarak etki etmiştir. YK4 Sobel kenar algılama algoritması 1.42 dB (TA için 40,54 dB'den 39.12 dB'ye) azalış ve 0.18 dB (BDA için 40,25 dB'den 40.07 dB ye) azalış olarak etki etmiştir.
- Nasır (2007) temelli yöntemdeki BDA, kenar algılama uygulanmayan YK4 Canny'deki BDA'ya göre yaklaşık 10 kat daha yavaştır (61.68 sn/6.16 sn). YK4 Canny kenar algılama algoritması 0.21 dB (TA için 40,54 dB'den 40.33 dB'ye) azalış ve 0.18 dB (BDA için 40,25 dB'den 40.07 dB ye) azalış olarak etki etmiştir.

- YK4 Canny kenar algılama algoritması ile Nasır (2007) temelli yöntemle göre 5-10 kat hızlı olmakla birlikte klasik YK1'e göre 0.18-0.26 dB arasında daha düşük TSGO değerine sahiptir kenar algılama tabanlı yöntem Nasır (2007) tabanlı yöntemle göre üstünlük sağlar. Çünkü BDA'ya göre 0.03-0.12 dB'lik kazanım sağlayıp 5-10 kat yavaş olan algoritma yerine 0.18-0.26 dB lik kazanımla 5-10 kat hızlı sonuçlar elde etmek tercih edilir bir kazanımdır.

Sonuç olarak, sabit kamera görüntü dizileri çekiminde nesne olan bölgelerin tespitinde, bu bölgelerde arama tekniklerinin kullanılarak arama sürelerinin belirgin bir şekilde kısalmada vazo, masa, ağaç gibi geri plandaki sabit nesnelerin elenmesinde bu çalışmanın kazanımları ve kenar bölgelerinin TSGO değerlerine etkisi gözlemlenmiştir. Uygulamalarımızda arama süresinin mi, yoksa TSGO değerlerinin mi önem arz ettiği konusu; hareket algılaması mı, hedef takibi mi, güvenlik uygulaması mı yapılacağı konusu; görüntü iletiminde kaliteden ne kadar feragat edilebileceği konusu gibi ihtiyaç ve amaçlara göre bu çalışma gerçek hayatta uygulama alanında yerini alacaktır.

Bu konuda daha ileri bir çalışma olarak, hareketli kamera için kamera hareket hızı ve yönünü, arama tekniklerinin kullanılmadığı geri plan bölgesi için modellemek; sabit olmayan öbek büyüklükleri kullanılarak bütün halinde hareket eden nesnelerin hareketlerinin incelenmesi düşünülebilir.

KAYNAKLAR

- Anuta, P. F. 1969. Digital registration of multispectral video imagery, Soc. Photo-Opt. Instrum. Eng. J., 7, 168-175.
- Bovik, A. 2000. Handbook of image and video processing, Academic Pres.
- Borshukov, G.D., Bozdagi, G., Altunbasak, Y. and Tekalp, A.M. 1997. Motion segmentation by multistage affine classification. Image Processing, IEEE Transactions on Volume 6; Page(s):1591 – 1594.
- Boyaci, F. and Telatar, Z. 2004. Motion Estimation In Video Sequences By Edge Detection. Signal Processing and Communications Applications Conference, Proceedings of the IEEE 12th ; Page(s):474 – 477.
- Chan, M. H., Yu, Y.B. and Constantinides, A. G.1990. Constantinides, Variable size block matching motion compensation with applications to video coding, IEEE Proc., 137(4), 205-212.
- CCITT Report SG xV. 1990. Recommendation H.261- Video Codec for Audio Visual Services at px64 Kbit/s.
- Green, B. 2002. Canny Edge Detection Tutorial. Drexel University Philadelphia.ABD.
- ISO/IEC JTC 1/SC29. 1993. Information Technology. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s. Part2: Video ISO/IEC 11172–2 (MPEG-1 Video), İsviçre.
- ISO/IEC JTC 1/SC29. 1996. Information Technology. Generic Coding of Moving Pictures and Associated Audio Information: Video ISO/IEC 13818–2

- ISO/IEC 14496–2. 1998. Information Technology – Generic Coding of Audio-Visual Objects Part 2: Visual, Ver. 18., ABD.
- ISO/IEC Report JTC1 CD 11172. 1991. Information Technology- Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5Mbit/s- Part 2: Coding of Moving Pictures Information.
- ITU-T SG15. 1993. Video Codec for Audio Visual Services at px64 Kbits/s ITU-T Recommendation H.261 Version 3
- ITU-T SG15. 1996. Video Coding for Low Bitrate Communication Draft ITU-T Recommendation H.263
- Jain, J. and Jain, A. 1981. Displacement Measurement and Its Application in Interframe Image Coding. Communications, IEEE Transactions on Volume 29; Page(s):1799 – 1808.
- Jianhua, L. and Ming, L.L. 1997. A simple and efficient search algorithm for block-matching motion estimation. Circuits and Systems for Video Technology, IEEE Transactions on Volume 7 ;Page(s):429 – 433.
- Kang, C. and Wang, W. 2007. A novel edge detection method on the maximizing objective function. Pattern Recognition. volume 40; Page(s): 609-618.
- Koga, T., Linuma, K., Hirano, A., Lijima, Y. and Ishiguro, T. 1981. Motion Compensated Interframe Coding for Video Conferencing. Proceedings of NTC, pp G5.3.1–G5.3.5, ABD.
- Knuth, D. E. 1973. Searching and Sorting, The Art of Computer Programming, Vol. 3, Addison-Wesley, Reading, MA.
- Lim, J.S. 1990. Two Dimensional Signal And Image Processing, Prentice Hall.

- Marr, D. and Hildreth, E.C. 1980.Theory of Edge Detection. Proc. Roy. Soc. London, Volume 207; Page(s): 187–217.
- Musmann, H. G., Pirsch P. and Grallert, H. J..1985. Advances in picture coding, Proc. IEEE, 73(4), 523-548.
- Nasır, S. 2007. 2-Boyutlu Hareket Kestirimi Ve Video Çerçeve Aradeğerlendirmesinde Uygulanması, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- Nie, Y. and Ma, K.K. 2002. Adaptive Root Pattern Search For Fast Block-Matching Motion Estimation.Image Processing, IEEE Transactions on Volume 11; Page(s):1442 - 1449 .
- Otsu, N.1979. A threshold selection method from gray-level histogram, IEEE Transactions on System Man Cybernetics, Vol.SMC-9, No. 1: 62-66.
- Po, L.M. and Ma, W.C. 1996. A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. Circuits and Systems for Video Technology, IEEE Transactions on Volume 6; Page(s):313 – 317.
- Reoxiang, L., Bing, Z. and Liou, M.L. 1994. A New Three-Step Search Algorithm for Block Motion Estimation. Circuits and Systems for Video Technology, IEEE Transactions on Volume 4;Page(s):438 – 442.
- Srinivasan, R., Rao, K. 1985. Predictive Coding Based on Efficient Motion Estimation. Communications.IEEE Transactions on Volume 33;Page(s):888 – 896.
- Zhu, S. and Ma, K.K. 2000. A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. Image Processing, IEEE Transactions on Volume 9; Page(s):287 – 290.

ÖZGEÇMİŞ

Adı Soyadı :Fatih BOYACI

Doğum Yeri :Ankara

Doğum Tarihi:19-01-1979

Medeni Hali :Evli

Yabancı Dili :İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise :Ankara Keçiören Kanuni Lisesi (1993-1996)

Lisans :Ankara Üniversitesi Mühendislik Fakültesi

Elektronik Mühendisliği Anabilim Dalı (1997-2002)

Yüksek Lisans: Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektronik

Mühendisliği Anabilim Dalı (Mart 2010)

Çalıştığı Kurum/Kurumlar ve Yıl:

Türk Telekomünikasyon A.Ş. (2003-2006)

T.C. Milli Eğitim Bakanlığı (2006-...)

Yayımları (SCI ve diğer):

Boyaci, F. and Telatar, Z. 2004. Motion Estimation In Video Sequences By Edge Detection. Signal Processing and Communications Applications Conference, Proceedings of the IEEE 12th 28-30; Page(s):474 – 477