

124066

**YÜZ RESMİ ARŞİVLEMeye YÖNELİK GÖRÜNTÜ
SİKİŞTİRMA**

Hatice ÇINAR
Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Mayıs – 2003

JÜRİ VE ENSTİTÜ ONAYI

Hatice Çınar'ın **Yüz Resmi Arşivlemeye Yönelik Görüntü Sıkıştırma** başlıklı **Elektrik-Elektronik Mühendisliği** Anabilim Dalındaki, Yüksek Lisans tezi ..26.15/2003 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	İmza
Üye (Tez Danışmanı)	: Yrd. Doç. Dr. ÖMER NEZİH GEREK	
Üye	: Prof. Dr. Hüseyin Akçay	
Üye	: Yrd. Doç. Dr. Yusuf Oysal	

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 28.05.2003 tarih ve17/2. sayılı kararıyla onaylanmıştır.

Enstitü Müdürü
Prof. Dr. Ömer ÖZER
Fen Bilimleri Enstitüsü
Müdürü

ÖZET

Yüksek Lisans Tezi

YÜZ RESMİ ARŞİVLEMeye YÖNELİK GÖRÜNTÜ SIKIŞTIRMA

HATİCE ÇINAR

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Ömer Nezih Gerek
2003, 60 sayfa

Bu çalışmada insan yüzü resimlerinde, göz, kaş, burun, dudak gibi yüz bölgelerinin ayrı ayrı değerlendirildiği bir kodlama yöntemi geliştirildi. Bu uygulanan yöntemin genel amacı diğer sıkıştırma yöntemleriyle karşılaştırılabilir sonuçlar elde edilmesi olmuştur. Bölgeler yüz resimlerinden çıkarıldıktan sonra her ayrı bölgenin kodlanması amacıyla, çeşitli vektör nicemeleme yöntemleri kullanıldı. Yüz resimlerinin geriye kalan bölgelerinin arasında çıkarım sonucu oluşan boşluklar uygun bir aradeğerleme ile dolduruldu. Daha sonra geri kalan yüz resminin entropisinin düştüğü gözlenerek ayrı bir yöntemle kodlandı. Ayrıca yüz bölütlerinin ayrı kodlanması kolay bir görüntü geri elde etme sistemine ulaşılmasını sağladı. Yüz resmindeki belirleyici özellikler olan organların vektör nicemlenmesiyle yüz benzerlikleri bilgisine karşılık gelen dizinler elde edildi.

Anahtar Kelimeler: İnsan Yüzü Resimleri, Görüntü Bölütleme, Sıkıştırma, Vektör Nicemeleme

ABSTRACT

Master of Science Thesis

SEGMENTATION BASED CODING OF HUMAN FACE IMAGES FOR RETRIEVAL

HATİCE ÇINAR

**Anadolu University
Graduate School of Natural and Applied Sciences
Electrical and Electronics Engineering Program**

**Supervisor: Assist. Prof. Ömer Nezh GEREK
2003, 60 pages**

In this study, a coding method for human face images by segmenting the face regions such as eyes, eyebrows, mouth, etc. is introduced. Each segmented region is removed from the face image, and these small images are coded using various vector quantization techniques. The removed segments on the human face are interpolated smoothly. Then the residual face image is observed to have reduced entropy and therefore it is encoded separately. The overall method is found to achieve comparable compression ratios with compatible general purpose coding methods. Furthermore, the distinct encoding of face segments enables us to accomplish an efficient face image retrieval system. Vector quantization of semantically meaningful face regions is observed to supply indexes that correspond to information about face similarities.

Keywords: Human Face Images, Image Segmentation, Compression, Quantizing and indexing

TEŞEKKÜR

Bu çalışma süresince yardımlarını ve değişik fikir ve önerilerini esirgemeyen ve beni tezimde doğru yönlendiren danışman hocam Yrd. Doç. Dr. Ömer Nezih GEREK'e en içten teşekkürlerimi sunuyorum.

Her zaman yanımda olan ve bana her türlü desteği veren ailemede sonsuz saygı ve sevgilerimi sunuyorum. Ve tez süresince bana destek olan tüm arkadaşlarımda ayrı ayrı teşekkür ediyorum.

Benim için hayatı her zaman daha güzel kılan ve hep yanımda olduğunu bildiğim, bana cesaret veren ve desteğini esirgemeyen Tümer Akakın'a çok teşekkür ediyorum.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
1. GİRİŞ	1
1.1. Genel Bakış	1
1.2. Örnek Çalışmalar	2
1.3. Bu Çalışmanın İçeriği	3
2. BÖLGELENDİRME	5
2.1. Yüz Resminin Bölgelere Ayrılması	5
2.2. Bölütlenmiş Yüzün Aradeğerlenmesi	8
3. BÖLÜTLERİN VEKTÖR NİCEMLENMESİ	11
3.1. En İyi Ölçütü	13
3.1.1. En Yakın Komşu Koşulu	13
3.1.2. Merkez Koşulu	13
3.1.3. LBG VQ Tasarım Algoritması	14
3.2. Merkez Noktalar İçin Mesafe Ölçütleri	16
3.2.1. Öklit Uzaklık (Euclidean Distance)	19
3.2.2. Manhattan Uzaklık (City-block veya Mutlak Değer Uzaklık)	20
3.2.3. Maximum Uzaklık (Chebychev Uzaklık)	20
3.2.4. Hamming Uzaklık	20
3.3. Yüz Organlarının LBG VQ ile Kodlanması	21
3.3.1. Vektörlerin Oluşturulması	22

4. GERİ KALAN YÜZ RESMİNİN SIKIŞTIRILMASI.....	23
5. VERİ TABANINDA SORGU RESMİNİN ARANMASI	24
6. DENEYSEL ÇALIŞMALAR	27
6.1 Sıkıştırma Sonuçları	27
7. SONUÇLAR.....	51
8. KAYNAKLAR.....	53
9. EKLER	56
EK-1 Öklit Uzaklığı Ve Ortalama	56
EK-2 City-block Uzaklığı ve Ortanca.....	56
EK-3 Hamming Uzaklığı	57
EK-4 Mahalanobis Uzaklığı.....	58
EK-5 SPIHT (Set Partitioning in Hierarchical Trees)	59
EK-6 LOCO-I (LOw COMplexity LOssless COMpression for Images)	59

ŞEKİLLER DİZİNİ

2.1. Yüz organlarının arama isleminin gösterilmesi.....	7
2.2. Örnek bir resim üzerinde arıylacak bölgelerin gösterilmesi.....	7
2.3. Çıkarılan bazı organlara ait görüntüler	8
2.4. (x,y) noktasında parçalı türeve sayısal bir yaklaşım oluşturmak için ızgara noktaların kullanılması.....	9
2.5. Örnek resim üzerinden önce gözler ve kaşlar, ardından burun, son olarak da ağzın çıkarılıp, geri kalan kısma aradeğerleme yapılması.....	10
3.1. Basit bir VQ örneğinin gösterimi.....	11
3.2. Öznitelik uzayının koordinatlarının ölçeklenmesi Öklit metrik ile hesaplanan mesafe ilişkisini değiştirebilir. Birinci grafikte x test noktası siyah daireye daha yakın iken, $\alpha=1/3$ seçilerek koordinat ölçeklenmesinden sonra kırmızı üçgenin x test noktasına daha yakın olmuştur.....	17
3.3. Yüz resimlerinden çıkarılan aynı piksel koordinatlara sahip göz bölgesinden X vektörünün elde edilmesi	22
5.1. Örnek bir sorgu resminin sorgulanması	25
5.2. f vektörünün P vektörüne olan Hamming uzaklığının bulunması	25
6.1. $BR\approx 0.6$, LOCO-I yöntemi, bayan yüz resimleri VQ1a	29
6.2. $BR\approx 0.6$, LOCO-I yöntemi, erkek yüz resimleri VQ1a	30
6.3. $BR\approx 0.6$, Spiht yöntemi, bayan yüz resimleri, VQ1a	30
6.4. $BR\approx 0.6$, Spiht yöntemi, erkek yüz resimleri, VQ1a.....	31
6.5. $BR\approx 1.5$, LOCO-I yöntemi, bayan yüz resimleri, VQ1a.....	31
6.6. $BR\approx 1.5$, LOCO-I yöntemi, erkek yüz resimleri, VQ1a	32
6.7. $BR\approx 1.5$, Spiht yöntemi, bayan yüz resimleri, VQ1a	32
6.8. $BR\approx 1.5$, Spiht yöntemi, erkek yüz resimleri, VQ1a.....	33
6.9. $BR\approx 0.6$, LOCO-I yöntemi, bayan yüz resimleri, VQ1b	35
6.10. $BR\approx 0.6$, LOCO-I yöntemi, erkek yüz resimleri, VQ1b.....	35
6.11. $BR\approx 0.6$, Spiht yöntemi, bayan yüz resimleri, VQ1b	36
6.12. $BR\approx 0.6$, Spiht yöntemi, erkek yüz resimleri, VQ1b.....	36
6.13. $BR\approx 1.5$, LOCO-I yöntemi, bayan yüz resimleri, VQ1b	37

6.14. BR≈1.5, LOCO-I yöntemi, erkek yüz resimleri, VQ1b	37
6.15. BR≈1.5, Spiht yöntemi, bayan yüz resimleri, VQ1b	38
6.16. BR≈1.5, Spiht yöntemi, erkek yüz resimleri, VQ1b.....	38
6.17. BR≈0.6, LOCO-I yöntemi, bayan yüz resimleri, VQ2a.....	40
6.18. BR≈0.6, LOCO-I yöntemi, erkek yüz resimleri, VQ2a	40
6.19. BR≈0.6, Spiht yöntemi, bayan yüz resimleri, VQ2a	41
6.20. BR≈0.6, Spiht yöntemi, erkek yüz resimleri, VQ2a.....	41
6.21. BR≈1.5, LOCO-I yöntemi, bayan yüz resimleri, VQ2a.....	42
6.22. BR≈1.5, LOCO-I yöntemi, erkek yüz resimleri, VQ2a	42
6.23. BR≈1.5, Spiht yöntemi, bayan yüz resimleri, VQ2a	43
6.24. BR≈1.5, Spiht yöntemi, erkek yüz resimleri, VQ2a.....	43
6.25. BR≈0.6, LOCO-I yöntemi, bayan yüz resimleri, VQ2b	45
6.26. BR≈0.6, LOCO-I yöntemi, erkek yüz resimleri, VQ2b	45
6.27. BR≈0.6, Spiht yöntemi, bayan yüz resimleri, VQ2b	46
6.28. BR≈0.6, Spiht yöntemi, erkek yüz resimleri, VQ2b.....	46
6.29. BR≈1.5, LOCO-I yöntemi, bayan yüz resimleri, VQ2b	47
6.30. BR≈1.5, LOCO-I yöntemi, erkek yüz resimleri, VQ2b	47
6.31. BR≈1.5, Spiht yöntemi, bayan yüz resimleri, VQ2b	48
6.32. BR≈1.5, Spiht yöntemi, erkek yüz resimleri, VQ2b.....	48

ÇİZELGELER DİZİNİ

3.1. merkez noktası için mesafe ölçütleri.....	18-19
3.2. ortanca fonksiyonunun gösterimi.....	22
5.1. Örnek sorgu resminin en benzer resimle karşılaştırılması	26
6.1. VQ1a ile kodlanmış resimlerin farklı kodlanmış yüz resimleriyle görsel karşılaştırılması	28
6.2. kodlanmış resimlerin bit sayılarının gösterimi	29
6.3. VQ1b ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması	34
6.4. VQ2a ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması	39
6.5. VQ2b ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması	44
6.6. BR=0,6 iken 50'şer resimli deney kümelerindeki kodlanmış resimlerin toplam ortalama karesel hataları	50
6.7. BR=1,5 iken 50'şer resimli deney kümelerindeki kodlanmış resimlerin toplam ortalama karesel hataları	50

SİMGELER ve KISALTMALAR DİZİNİ

- MSE1 : VQ Kodlanmış organların, uygun bir kodlama yöntemiyle kodlanmış olan geri kalan yüz resmine yerleştirilmiş yüzün orijinal yüzle arasındaki ortalama karesel hatayı gösterir
- MSE2 : Orijinal yüz resminin geri kalan yüz resmiyle aynı kodlama yöntemiyle yaklaşık olarak aynı sıkıştırma oranına sahip olacak şekilde kodlanmasıyla elde edilen yeni yüzün ortalama karesel hatasını gösterir
- VQ1a : Orijinal LBG VQ yöntemi (Mean kullanılan)
- VQ1b : Orijinal LBG VQ yöntemi (Median kullanılan)
- VQ2a : Sonradan uygulanan LBG VQ yöntemi (Mean kullanılan)
- VQ2b : Sonradan uygulanan LBG VQ yöntemi (Median kullanılan)

1. GİRİŞ

1.1. Genel Bakış

Günümüz toplumunda bir çok farklı biçimleriyle faydalı ve değerli olan bilgi, büyük bir oranla artmaktadır. Sonuç olarak da bilginin verimli bir şekilde saklanabilmesi, ulaşılabilmesi ve iletilmesi önem kazanır. Bu özellikle sayısal görüntülerde geçerlidir. Tek bir sayısal görüntüyü temsil etmek için bile büyük miktarda sayısal boş yere ihtiyaç duyulur; alıcı teknolojisindeki ve sayısal elektronikteki hızlı ilerlemeler yeni jenerasyon ürünlerle, ihtiyaç duyulan bit sayısını daha da artırmıştır. Sayısal görüntülerden verimli bir şekilde faydalanmak için de görüntünün ifadesinde gerekli olan bit sayısını azaltacak yeni spesifik tekniklere ihtiyaç vardır [1]. Sayısal görüntü işlemenin bu problemle ilgilenen dalına görüntü sıkıştırma (kodlama) denmektedir. Sayısal görüntülerin hem miktar, hem kullanım olarak artması sonucunda, 1950'lerden beri üzerinde çalışılan görüntü sıkıştırma konusu daha fazla önem kazanmıştır. Çevremizdeki görüntüleri tanıyabilen bilgisayarların tasarlanması ve kurulması araştırmaların doğal olarak yöneldiği bir konudur. İnsan yüzü tanımadan parmak izi tespitine, optik karakter tanımadan DNA dizisi tespiti ve daha fazlası için bilgisayarla hassas görüntü tanıma oldukça faydalıdır.

Görüntü sıkıştırma, video-konferans, uzaktan algılama, doküman ve tıp görüntüleri ve faks iletimi gibi; ikili, gri ölçekli veya renkli görüntülerin verimli işlenmesi, saklanması ve iletilmesi uygulamalarında çok önemli bir rol oynar. Görüntü sıkıştırma teknikleri kayıplı ve kayıpsız olmak üzere iki kategoride sınıflandırılabilir. Kayıpsız metotlarla orijinal resim kesin olarak geri elde edilebilirken, kayıplı metotlarla ise sadece orijinal bilgiye yakın bir görüntü elde edilebilir.

Genel görüntüler için muhtelif algoritmalar geliştirilmiştir. Bütün yöntemlerin temel amacı, görüntüyü oluşturan bileşenlerin birbiriyle en az ilişkili olduğu gösterim şeklini bulmaktır, başka bir deyişle gereksiz bilgi göz ardı edilerek sadece faydalı bilginin kullanılacağı bir gösterim şekli bulmaktır.

Görsel bilgi geri kazanımı bilgi teknolojisinde yeni bir konu oluşturmaktadır. Amacı ise bir veri tabanından, görüntüden veya görüntü dizisinden sorguya ilişkin olanları geri elde etmektir. Görsel içerik ile etkileşim halinde olmak görsel bilgi kazanımı için oldukça gereklidir. Yeni araçlar ve etkileşim örnekleri görsel verinin direkt içeriğini kapsayan aramalara izin vermelidir. Renk, doku ve nesne şekli, uzamsal ilişki gibi görsel elemanlar görüntü içeriğinin algısal görünüşüyle direkt olarak ilgilidir. Buna bağlı olarak görüntü analizi, örüntü tanıma ve bilgisayarla görü gibi konular görsel bilgi geri kazanımı sistemlerinde önemli bir role sahiptir. Görüntü geri kazanımı görüntüler arasındaki benzerlik kriterlerine göre bölümlere ayrılmaktadır. Bunlar renk benzerliğine göre, doku benzerliğine göre, şekil benzerliğine göre yada uzamsal bağlantıya göre görüntü geri kazanımı yapılabilmektedir [23]. Görüntülerin algısal özelliklerine göre geri kazanımı için, en basit örnek her görüntünün ayırt edici özelliklerinin (model parametrelerinin) bir kümesinin bulunmasıdır. Sorgular görsel örneklerle ifade edilmektedir. Bir sorguyu başlatmak için, kullanıcının önemli olan özellik ve aralıktaki model parametrelerini seçerek bir benzerlik ölçütü bulması gerekir. Sistem kullanıcının sorgusunun görsel içeriği ile veri tabanı görüntüleri arasındaki benzerliği kontrol eder.

1.2. Örnek Çalışmalar

Daha özelleşmiş konularda görüntü kodlamanın önemi, yakın dönemde daha da artmıştır. Örneğin insan yüzü resimleri, özel içerikleri ve önemli kullanım alanları nedeniyle sayısal görüntü işleme ve kodlama alanlarında pek çok araştırmacının ilgisini çekmiştir [2-3,21-22]. Böylece yüz resimleri önemli bir sayısal arşiv konusu oluşturmuştur [4]. Yüz resimlerinde belirleyici nitelikler olan göz, ağız, burun gibi organların yerlerinin tespiti bir çok araştırmada incelenmiştir [25,26]. Yüz resimlerini sıkıştırmak için literatürde bir kaç sayıda yöntem bulunmaktadır. Resmin küçük bölümlere ayrılması analizi ve kodlanması açısından iyi bir performans sağladığı gösterilmiştir [5],[6]. Özellikle görüntülü telefon ve video-konferans uygulamaları sırasında yüz resminin bölgelere ayrılarak incelenmesinin önemli bir avantaj getirdiği gözlenmiştir. Bu tür

bölgelendirme getirdiği avantaj MPEG-4 video standardı ile genel video kodlamaya da uygulanmış, MPEG-7 standardı ile de şu andaki halini almıştır. Ayrıca büyük görüntü veri tabanlarının (>10,000) oluşmasıyla, görüntülerin taranması ve otomatik düzenlenmesi için bazı araçlara ihtiyaç doğmuştur. Geçmiş on sene içerisinde görüntü veri-tabanlarını kontrol etmek için içerik-tabanlı metotlar büyük ilgi toplamıştır. Geri elde edilebilir görüntü veritabanlarındaki çoğu içerik-tabanlı ilk çalışmalar sorgulayarak-arama yaklaşımında yoğunlaşmıştır.

1.3. Bu Çalışmanın İçeriği

Bu çalışmada, hem sıkıştırma için hem de geri kazanımdaki benzerlik ölçütü için bölgelendirme ele alınmıştır. Kodlama verimini enbüyüten uygun bölütlerin bulunması, daha büyük değişimlerin olduğu bölgelerde daha küçük bölütlerin elde edildiği [7]' te araştırılmıştır. Diğer bölütleme yaklaşımları genellikle bir görüntü modelini ele alır ve veriyi tavsiye edilen modele uydurur.

İnsan yüzünde doğal olarak kullanılan bölgelendirme unsurları, yüz resmindeki en yüksek piksel değişkenliğine sahip yerler olan, aynı zamanda da insan yüzünün doğal organlarına karşılık gelen saç, göz, kaş, burun, ve dudaktır [6]. Bizim çalışmamızda da insan yüzü görüntüleri ele alınmış ve yukarıda bahsedilen bölgelerin ayrı ayrı kodlandığı algoritmalar geliştirilmiştir. Bu yöntem doğal olarak sorgulama ve veri tabanı yaklaşımlarını da gündeme getirmiştir. Sıkıştırma ve sorgulamaya uygunluğun bir arada bulunması, arzu edilen bir durumdur. Bunun yanında bu bölgeler birçok yüz resmi arasında önemli benzerlikler içermektedir. Bu nedenle bu bölgeler vektör nicemleme (VQ) gibi kümeleme yöntemli kodlayıcıların kullanılması için uygundur. VQ ile devam edilen bölütlemenin bir avantajı da VQ kod çizelgesi dizinlerinin yüz resimleri için kolayca bazı ayırma ölçütleri sağlamasıdır. Çünkü bu yüksek piksel değişimli bölgeler aslında insanların yüzleri tanımak için dikkate aldığı öznelik bilgisini içeren önemli bölgelerdir [8]. Bu bölütleri kümeleyerek ve kod numaraları atayarak, VQ kodlamayı ve geri kazanma parametrelerini aynı anda elde edilmiş oluyor.

Bölgelere ayırma sonucu elde edilen resimlerin küçük olması, hatta VQ ile kodlanarak sadece birer rakama dönüşmesi, veri tabanında sorgulama yapmada bize büyük bir kolaylık sağlamıştır.

Kullanılan yöntem temelde üç aşamadan oluşmaktadır:

1. Resimden yüze ait kısmın çıkartılıp, göz, kaş, burun ve ağız bölgelerinin ayrılması.
2. Bölgelere ait küçük resimlerin kodlanması: Bunun için VQ kullanıldı.
3. Çıkarılmış bölgelerin ardından kalan ve daha düz olan yüz bölgesinin kodlanması.

Bu aşamaların ardında her bölge için VQ tarafından üretilen numaraların, ve bölgelerin yüz resmi üzerindeki koordinatlarının bulunduğu öznitelik vektörleri (feature vectors) veri tabanına yüklenmiştir. Resmin geri çatılabilmesi için de bölgelerden geriye kalan kısımlara ait resimler ayrı bir yerde depolanmıştır. Geliştirdiğimiz yöntemin ilk aşamasında benzer boyuttaki yüz resimlerinden kullanıcı yardımıyla sabit büyüklük ve şekilde bölge çıkarma işlemi yapılmaktadır. Elimizdeki yüz resimlerine uygun bir şekilde, organları kapsayacak büyüklükte ve çokgen olarak seçilen zarf şekilleri kullanılmıştır. Bu şekillerin iç bölgelerinde kalan sol göz, sağ göz, sol kaş, sağ kaş, burun ve ağız kısımları, oluşturulan ayrı veri tabanları içerisinde depolanmıştır. Yüz organ resimlerinin VQ ile en küçük sapma seviyesine ulaşması için eniyileme metotları elde edilmiştir. Bu amaçla özyineli bir yöntem olan LBG [9] algoritması kullanılmıştır.

Veritabanındaki kullanılan fotoğraf sayısı 118'dir. Bunların 62'si bayanlara, 56'si erkeklere ait yüz resimleridir.

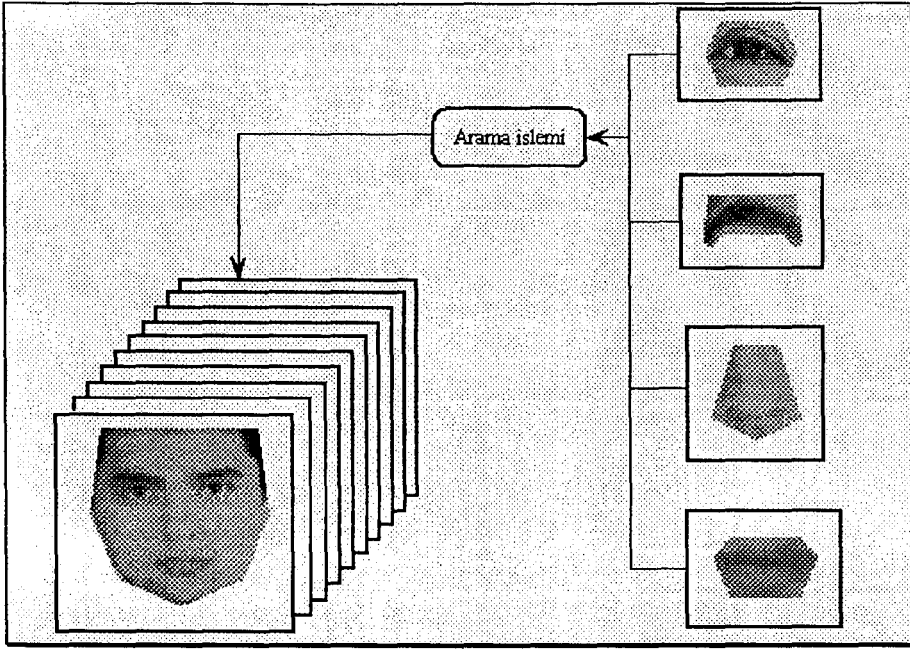
2. BÖLGELENDİRME

İnsan yüzü bir görüntüde veya videodaki önemli nesnelere biridir. Bir görüntüdeki insan yüzünün ve yüz organ yerlerinin algılanması, insan yüzü tanıma, gözetim sistemleri, insan-bilgisayar arabası, video-konferans gibi geniş bir uygulama alanı olan önemli bir işidir. Görüntü bölütleme, bir görüntü kodlamada bit hızını azaltmak için insan görsel sistem özelliğinin kullanılmasına yardımcı olur. İnsan görsel sistemi resimdeki kenarlara özel vurgular koyar, bu nedenle, bütün bir görüntüyü aynı biçimde kodlamaktansa, kenarlara göre bloklara ayrılmış kısımlara daha farklı kodlama uygulanabilir [10]. Bölütleme işlemi BSP-ağaç (Binary Space Partitioning) ve dördün ağaç yöntemleri ile de yapılabilmektedir [5]. Bu yöntemde dördün ağaç gösterimi sadece kare bölütlemeye izin verirken, BSP-ağaç gösterimi bölütlenmiş bölgelerin rastgele çokgen şekillerde olmasına izin vererek sayısal görüntülerin daha verimli ve az yer kaplayan gösterimini sağlamış olmaktadır. Bu çalışmada ise örnek çokgen şekilli görüntüler bölütlere ayrılacak yüz görüntüsü üzerinde taratılarak her piksel için istatistiksel dağınıklık hesaplaması yapılmıştır. Minimum istatistiksel dağınıklığa sahip olan nokta bize bölütleme yapacağımız bölgeyi göstermektedir.

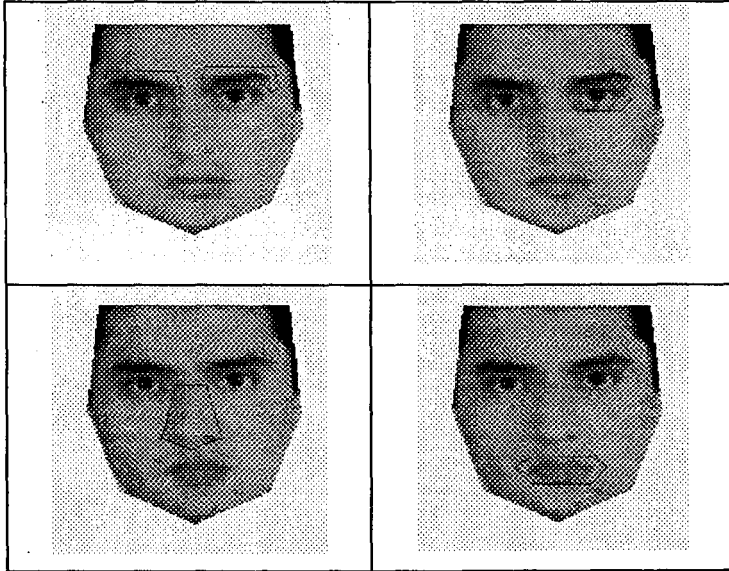
2.1. Yüz Resminin Bölgelere Ayrılması

Bu çalışmadaki görüntüler sadece gri seviyeli vesikalık ön yüz resimlerinden oluşmaktadır. Gri seviyeli yüz resimlerinde karşılaşılan sorunlardan biri ten renginin kişiden kişiye farklılık göstermesidir. Ten renklerindeki en büyük farklılık ise ışık yoğunluğundan kaynaklanmaktadır. Bundan dolayı işleme başlamadan önce resimlerin parlaklık seviyeleri normalize edilmiştir. Daha sonra yüz görüntülerinden yüz organ yerlerinin belirlenmesi için çokgen zarf şekilli örnek yüz organ resimleri oluşturulmuştur. Bu organ resimleri sağ kaş, sol kaş, sağ göz, sol göz, burun ve dudak görüntülerini içermektedir. Örnek organ resimlerini her bir yüz görüntüsü üzerinde taratarak istatistiksel dağınıklığının en küçük olduğu noktalar aranmaktadır. Buna benzer bir çalışmada ise görüntüye ait şablonlar kullanılarak kovaryans matris hesaplanmaktadır. Kovaryans matris bir kare matristir ve tüm şablonlara ilişkin bilgileri içermektedir [25].

Yüz organ resimlerinin yüz resmi üzerinde aranması Şekil 2.1.'de gösterilmektedir. Her bir organ resmi yüz resmi üzerinde taratılarak istatistiksel dağılımının en küçük değere sahip olduğu noktalar yani taratılan resim ile yüz resmi arasındaki benzerliğin en çok olduğu noktalar aranmaktadır. Bu noktalar bize o organın yüz görüntüsü üzerindeki yerlerinin koordinatlarını vermektedir. Bu işlem her bir organ için bütün yüz resimlerinde yapılmış ve referans olarak kullanacağımız yüz organ koordinat verileri bulunmuştur. Bu koordinat verileri ise daha sonraki işlemlerde kullanılmak üzere saklanmıştır. Sonraki işlemde ise bu veriler kullanılarak her bir yüzden uygun çokgen şekillerde sağ kaş, sağ göz, sol kaş, sol göz, burun ve dudaklar çıkartılmıştır. Şekil 2.2.'de zarf şekilleri ve içerisinde kalan bölgeler bir örnek yüz resmi üzerinde gösterilmektedir.

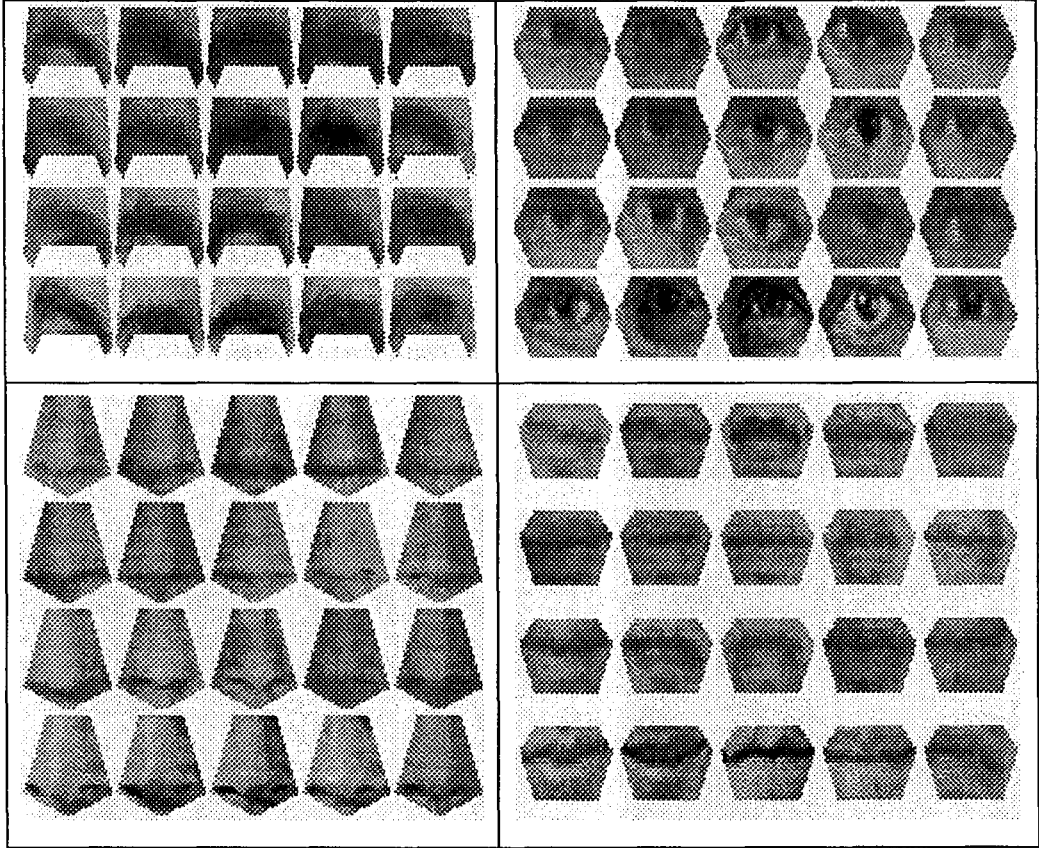


Şekil 2.1. Yüz organlarının arama işleminin gösterilmesi



Şekil 2.2. Örnek bir resim üzerinde ayrılacak bölgelerin gösterilmesi

Şekil 2.3.'te yukarıda bahsedilen zarf şekilleri ve içerisinde kalan bölgeleri göstermek açısından 20 deney resminden çıkarılan sol göz, sol kaş, burun ve ağız bölgelerine ait kısımlar gösterilmektedir.



Şekil 2.3. Çıkarılan bazı organlara ait görüntüler

2.2. Bölütlenmiş Yüzün Aradeğerlenmesi

Yüzdeki organ kısımlarının geri kalan bölgelerden ayrılması işleminde eğer çıkarılan bölge sıfırlarla doldurulursa, geriye kalan resimdeki varyans çok artmakta, dolayısıyla kodlanması zorlaşmaktadır. Bu nedenle, geriye kalan resimde, çıkarılan bölgelerin uygun bir aradeğerleme ile doldurulması düşünülmüştür. Aradeğerleme işleminin standard yordamı veriyi sürekli bir fonksiyonla oluşturmak ve istenilen ince aralıklarla fonksiyonu örneklemektir. En basit aradeğerleme fonksiyonu, aradeğerleme noktasındaki değeri için kendisine

en yakın olan ızgara noktanın deęerini alan, en yakın komşu fonksiyonudur. Doğrusal aradeęerleme de , bilinmeyen nokta, ızgara noktalar arasında doğrusal olarak aradeęerlendirilir [11]. Bu alıřmada ise öngörücü kodlamaya en elverişli olacak şekilde aradeęerleme yöntemi olarak, bölge kenarlarındaki iki boyutlu Laplace denklemini özerek yumuřak geiři saęlayan bir yöntem kullanılmıřtır [12]. İki boyutlu Laplace denklemi řu şekildedir:

$$h_{xx} + h_{yy} = 0 \quad (2.1)$$

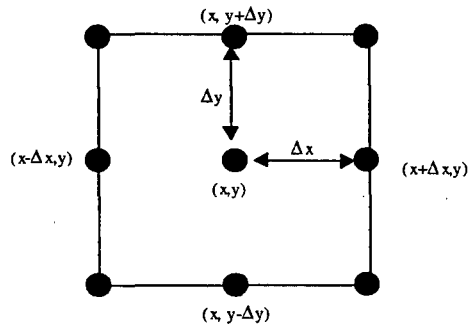
Denklem 2.1 akılda tutularak, sayısal özümlemeden birkaç temel kavram hatırlamırsa, bir fonksiyonun türevine řu şekilde yaklařılabilir;

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \quad (2.2)$$

Denklem 2.2'ye ayrıca türeve merkezi fark yaklařımında denir. İkinci derece türeve yaklařımdan ařaęıdaki gibi benzer bir denklem ıkmaktadır.

$$f''(x) \approx \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2} \quad (2.3)$$

Denklem 2.3'e de ikinci dereceden türeve merkezi fark yaklařımı denir [13].



řekil 2.4. (x,y) noktasında paralı türeve sayısal bir yaklařım oluřturmak için ızgara noktaların kullanılması

Şekil 2.4.'teki ek nohtalarda h değerlerinin bilindiği varsayılarak, ikinci derece türevine merkezi fark yaklaşımı uygulanırsa;

$$h_{xx}(x, y) \approx \frac{h(x + \Delta x, y) - 2h(x, y) + h(x - \Delta x, y)}{(\Delta x)^2} \quad (2.4)$$

$$h_{yy}(x, y) \approx \frac{h(x, y + \Delta y) - 2h(x, y) + h(x, y - \Delta y)}{(\Delta y)^2} \quad (2.5)$$

Denklem 2.4 ve 2.5 Laplace denkleminde (denklem 2.1) yerine konduğunda

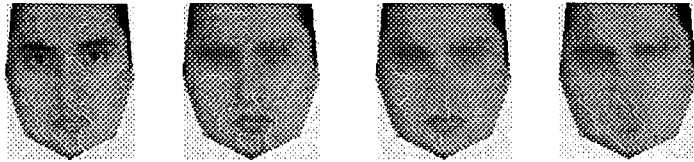
$$\frac{h(x + \Delta x, y) - 2h(x, y) + h(x - \Delta x, y)}{(\Delta x)^2} + \frac{h(x, y + \Delta y) - 2h(x, y) + h(x, y - \Delta y)}{(\Delta y)^2} = 0 \quad (2.6)$$

denklemini oluşur.

Eğer rastgele kare bir ızgara seçilirse $\Delta x = \Delta y$, sadeleştirmeden sonra denklem 2.6

$$h(x, y) = \frac{h(x + \Delta x, y) + h(x, y - \Delta y) + h(x - \Delta x, y) + h(x, y + \Delta y)}{4} \quad (2.7)$$

denklemine dönüşür. Bu da bize bir noktanın değerini komşuluğunda ki dört değerlerin ortalaması alınarak hesaplanabileceğini gösterir. Matlab programında bu yöntemi kullanan bir komutla, zarf şeklindeki çıkarılan bölgelerin sınırlarındaki piksel değerlerinden içeri doğru düzgün bir aradeğerleme yapılmıştır. Bölgelerin kademe kademe çıkarılması ve kalan yerlerin aradeğerlenmesi Şekil 2.5.'te örnek bir resim üzerinde gösterilmektedir.



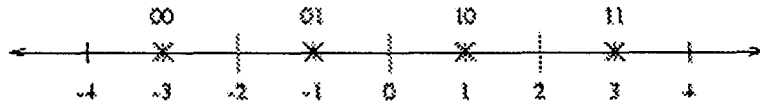
Şekil 2.5. Örnek resim üzerinden önce gözler ve kaşlar, ardından burun, son olarak ağız çıkarılıp, geri kalan kısma aradeğerleme yapılması

3. BÖLÜTLERİN VEKTÖR NİCEMLENMESİ

Tipik bir kodlama algoritmasında ki en önemli basamaklardan biri, bilgi kaybının olduğu nicemlemedir. Nicemlemenin amacı şu şekilde özetlenebilir: belli bir olasılık yoğunluk fonksiyonu ile tanımlanmış bir kaynaktan gelen işaretlerin mümkün olan en düşük bit sayısı ile kodlamak, öyle ki bu kodun çözülmesi ile elde edilen işaretin mümkün olduğu kadar yüksek kalitede olmasını sağlamaktır. Bir nicemleyiciye optimum diyebilmek için çıkıştaki bozulmanın minimum olması gerekmektedir. Bir nicemleyicide bozulma yerel olarak (blok içinde) minimum ise yapılan nicemleme genel olarak minimumdur. Böylece, orijinal görüntü sinyal bloklarının en iyi temsil edileceği bir kod kitabı oluşturulabilecektir.

Vektör nicemleme (VQ) sayısal görüntülerin sıkıştırılmasında kullanılan kayıplı bir sıkıştırma tekniğidir. 1980 yılında, Linde, Buzo, ve Gray (LBG) eğitici dizi tabanlı bir VQ tasarım algoritması önermişlerdir. Eğitici dizinin kullanılması çok boyutlu tümlev hesaplama gereksinimini ortadan kaldırmıştır [13]. Bu yöntemi kullanan VQ tasarımına LBG-VQ denmektedir.

VQ yaklaşık değer bulmadan başka birşey değildir. Basit olarak düşünüldüğünde bir yuvarlamadır (mesela en yakın tamsayıya). Bir boyutlu bir VQ örneği aşağıda gösterilmektedir.



Şekil 3.1. Basit bir VQ örneğinin gösterimi

Burada, -2'den küçük her sayı -3 ile gösterilmektedir. -2 ve 0 arasındaki her sayı -1 ile gösterilmektedir. 0 ve 2 arasındaki her sayı 1 ile gösterilmektedir. Ve 2'den büyük her sayı 3 ile gösterilmektedir. Bu örnek 1-boyutlu 2-bitlik bir vektör nicemlemedir. 2 bit/boyut oranına sahiptir. Yukarıdaki örnekte kırmızı yıldızlı yerler kod vektörlerini, mavi sınırlarla tanımlanan yerlerse kodlama bölgelerini göstermektedir. Bütün kod vektörler kümesine kod çizelgesi denmektedir ve bütün kodlama bölgeleri kümesine de uzayın üleşimi denmektedir.

VQ problem tasarımı şu şekilde ifade edilebilir. İstatistiksel özellikleri bilinen bir vektör kaynağı, kod vektör sayısı, ve bozunma ölçüsü verildiğinde, kod çizelgesi ve en küçük bozunma ortalamasına sahip bir üleşim bulunur.

Eğitici dizinin M kaynak vektörden oluştuğu düşünülürse:

$$\mathbf{T} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} \quad (3.1)$$

Bu eğitici dizi büyük veri tabanlarından elde edilebilir. Kaynak vektörlerinin k -boyutlu olduğu varsayılırsa, örneğin,

$$\mathbf{x}_m = (x_m(1), x_m(2), \dots, x_m(k)), \quad m = 1, 2, \dots, M \quad (3.2)$$

N kod vektör sayısı olsun ve

$$\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N\}, \quad (3.3)$$

\mathbf{C} kod çizelgesini gösterebilir. Her kod vektör k -boyutlu, örneğin

$$\mathbf{c}_n = (c_n(1), c_n(2), \dots, c_n(k)), \quad n = 1, 2, \dots, N \quad (3.4)$$

S_n , kod vektör \mathbf{c}_n tarafından ilişkilendirilmiş olan kodlama bölgesini gösterebilir ve

$$\mathbf{P} = \{S_1, S_2, \dots, S_N\} \quad (3.5)$$

uzayın üleşimini gösterir. Eğer \mathbf{x}_m kaynak vektörü S_n kodlama bölgesinin içindeyse yaklaşımı ($\mathbf{Q}(\mathbf{x}_m)$ ile belirtilen) \mathbf{c}_n :

$$\mathbf{Q}(\mathbf{x}_m) = \mathbf{c}_n, \quad \text{eğer } \mathbf{x}_m \in S_n \text{ ise} \quad (3.6)$$

Kare hata bozunma ölçüsü varsayıldığında, ortalama bozunma şu şekilde verilir:

$$D_{ave} = \frac{1}{Mk} \sum_{m=1}^M \left\| \mathbf{x}_m - \mathbf{Q}(\mathbf{x}_m) \right\|^2 \quad (3.7)$$

Tasarım problemini şu şekilde ifade edilebilir: Verilmiş olan \mathbf{T} ve N 'den D_{ave} enküçültme edilmiş olarak \mathbf{C} ve \mathbf{P} 'nin bulunması.

3.1. Eniyilik Ölçütü

Eğer \mathbf{C} ve \mathbf{P} yukarıdaki enküçültme sorununa bir çözüm oluyorsa, aşağıdaki iki ölçütü de sağlaması gerekir.

3.1.1. En yakın Komşu Koşulu

$$S_n = \left\{ \mathbf{x} : \left\| \mathbf{x} - \mathbf{c}_n \right\|^2 \leq \left\| \mathbf{x} - \mathbf{c}_{n'} \right\|^2, \forall n' = 1, 2, \dots, N \right\} \quad (3.8)$$

Bu koşula göre S_n kodlama bölgesi \mathbf{c}_n kod vektörüne diğer kod vektörlerinden daha yakın olan tüm vektörleri içermelidir.

3.1.2. Merkez Koşulu:

$$\mathbf{c}_n = \frac{\sum_{\mathbf{x}_m \in S_n} \mathbf{x}_m}{\sum_{\mathbf{x}_m \in S_n} 1} \quad n = 1, 2, \dots, N \quad (3.9)$$

Bu koşula göre ise \mathbf{c}_n kodvektörü S_n kodlama bölgesi içindeki tüm eğitici vektörlerin ortalaması olmalıdır. Koşulu uygularken her kodlama bölgesinin en az bir eğitici vektöre sahip olması gerekir (böylece yukarıdaki denklemde payda hiçbir zaman sıfır olmaz).

3.1.3. LBG Tasarım Algoritması

LBG VQ tasarım algoritması yukarıda ki iki eniyilik ölçütlerini alternatif olarak çözen özyineli bir yöntemdir [14]. Algoritma için bir başlangıç \mathbf{c} kod çizelgesi gerekmektedir. Bu başlangıç kod çizelgesi bölümlenme yöntemiyle elde edilir. Bu yöntemde, bütün eğitici dizinin ortalaması olan bir başlangıç kod vektörü belirlenir. Daha sonra bu kod vektör ikiye bölünür. Başlangıç kod vektörü olarak bu iki kod vektörle özyineli algoritma çalıştırılır. Bulunan son iki kod vektörde dörde bölünür ve istenilen kod vektör sayısına ulaşılan kadar işlem tekrar edilir. Algoritma aşağıdaki gibi özetlenebilir.

1. Bir T belirlensin ve $\varepsilon > 0$ ve yeterince küçük bir sayı olsun
2. $N = 1$ olsun

$$\mathbf{c}_1^* = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m \quad (3.10)$$

olsun ve ortalama bozulma;

$$D_{ave}^* = \frac{1}{Mk} \sum \|\mathbf{x}_m - \mathbf{c}_1^*\|^2 \quad (3.11)$$

hesaplansın.

3. Bölümlenme: $i = 1, 2, \dots, N$ 'e kadar,

$$\begin{aligned} \mathbf{c}_i^{(0)} &= (1 + \varepsilon) \mathbf{c}_i^*, \\ \mathbf{c}_{N+i}^{(0)} &= (1 - \varepsilon) \mathbf{c}_i^* \end{aligned} \quad (3.12)$$

$N = 2N$ yapılsın.

4. Özyineleme: $D_{ave}^{(0)} = D_{ave}^*$ olsun ve özyineleme dizini $i = 0$ yapılsın.

- i. $m = 1, 2, \dots, M$ için, aşağıdaki denklemin (3.13) değerini

$$\| \mathbf{x}_m - \mathbf{c}_n^{(i)} \|^2 \quad (3.13)$$

$n = 1, 2, \dots, N$ aralığında bulun. n^* en küçük değere ulaştığı dizin olsun.

$$\mathbf{Q}(\mathbf{x}_m) = \mathbf{c}_{n^*}^{(i)} \quad (3.14)$$

- ii. $n = 1, 2, \dots, N$ için kod vektörünü güncelleyin

$$\mathbf{c}_n^{(i+1)} = \frac{\sum_{\mathbf{Q}(\mathbf{x}_m) = \mathbf{c}_n^{(i)}} \mathbf{x}_m}{\sum_{\mathbf{Q}(\mathbf{x}_m) = \mathbf{c}_n^{(i)}} 1} \quad (3.15)$$

- iii. $i = i + 1$ yapılsın.

- iv. Denklem 3.16 hesaplınsın

$$D_{ave}^{(i)} = \frac{1}{Mk} \sum_{m=1}^M \| \mathbf{x}_m - \mathbf{Q}(\mathbf{x}_m) \|^2 \quad (3.16)$$

- v. Eğer $D_{ave}^{(i-1)} - D_{ave}^{(i)} / D_{ave}^{(i-1)} > \varepsilon$, ise, basamak (i)'e geri dön.

- vi. $D_{ave}^* = D_{ave}^{(i)}$ yapılsın. $n = 1, 2, \dots, N$ için,

$$\mathbf{c}_n^* = \mathbf{c}_n^{(i)} \quad (3.1.)$$

son kod vektörler belirlensin.

5. İstenilen sayıda kod vektöre ulaşılan kadar basamak 3 ve 4 tekrarlınsın.

Yukarıdaki algoritmanın 3. basamağında yani kod vektörünün ikiye bölünmesi sırasında orijinal kod vektöre küçük bir vektör eklenip ve çıkarılarak iki ayrı vektör elde edilir:

$$\mathcal{E} = \{\mathbf{x}_m\}_{m=1}^M$$

'in kovaryans matrisinin en büyük özdeğerine (eigen value) karşılık gelen özvektörün yeteri miktarda büyük bir sayı ile çarpımı sonucu elde edilen bir vektördür. Böylece \mathcal{E} ; kod vektörün temsil ettiği bölgedeki en büyük varyasyonun bulunduğu doğrultuya karşılık gelen vektör seçilmiş olur. Bu şekilde kod vektör, resimleri eskisinden daha kötü temsil etmeyecek iki ayrı vektöre bölünür.

3.2. Merkez Noktalar İçin Mesafe Ölçütleri

Günümüzde bir çok VQ kod çizelgesi en iyilemesi yapan yöntemler vardır, bu çalışmada ise kod çizelgelerinin bulunmasında LBG [9] yöntemi kullanılmıştır. Bu yöntem başlangıç olarak tek bir kod çizelgesiyle başlar ve özyineli olarak kod vektörünü ikiye böler ve istenilen sayıda kod çizelgesine ulaşılan kadar her kod vektöre Lyod-Max algoritması uygulayarak en iyileme yapar. Mesafe ölçütlerinin tanımları ve kod vektörünü ikiye bölme yöntemi uygulamaya uygun olarak tespit edilmiştir [15-17]. En yakın komşu sınıflandırıcıları, mesafe ölçütü olarak örüntüler arasındaki metrik veya "mesafe" fonksiyonlarını kullanır. VQ'nun optimalite şartı olan temsil edici merkez noktalar her türlü mesafe ölçütü ($d(\dots)$) ile hesaplanabilir. Bir metrik olan $d(\dots)$, düşünülen iki örüntü arasındaki genelleştirilmiş skalar mesafeyi veren bir fonksiyondur. Metriğin dört özelliği sağlanması gerekir [24]: bütün \mathbf{a} , \mathbf{b} ve \mathbf{c} vektörleri için, bu özellikler aşağıda ki gibidir:

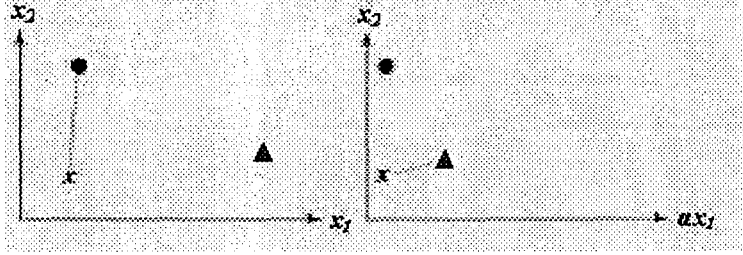
- Negatif olmama (Nonnegativity) : $d(\mathbf{a}, \mathbf{b}) \geq 0$

- Dönüşlülük (Reflexivity): $d(\mathbf{a}, \mathbf{b}) = 0$ iff $\mathbf{a} = \mathbf{b}$
- Simetri (Symmetry): $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$
- Üçgen eşitsizliği (Triangle inequality): $d(\mathbf{a}, \mathbf{b}) + d(\mathbf{b}, \mathbf{c}) \geq d(\mathbf{a}, \mathbf{c})$

n boyutlu mesafeler için Öklit formülünün

$$d(\mathbf{a}, \mathbf{b}) = \left(\sum_{k=1}^n (a_k - b_k)^2 \right)^{1/2} \quad (3.18)$$

metrik özelliklerine sahip olduğu kolayca doğrulanabilir. Sonuç anlamlı olsun veya olmasın; iki vektör arasındaki Öklit mesafe her zaman hesaplanabilir. Mesela her koordinatı rastgele sabitlerle çarpılarak bir uzay dönüşümü olursa dönüştürülen uzaydaki Öklit mesafe ilişkisi orijinal mesafe ilişkisinden oldukça farklı olabilir ki buna rağmen dönüşüm sadece farklı bir birim seçeneği sunmaktadır. Şekil 3.2.'de gösterildiği gibi ölçek değişiklikleri en yakın komşu sınıflandırıcılarında büyük bir etki yapmaktadır.



Şekil 3.2. Öznitelik uzayının koordinatlarının ölçeklenmesi Öklit metrik ile hesaplanan mesafe ilişkisini değiştirebilir. Birinci grafikte x test noktası siyah daireye daha yakın iken, $\alpha=1/3$ seçilerek koordinat ölçeklenmesinden sonra kırmızı üçgenin x test noktasına daha yakın olmuştur.

n -boyutlu örüntülerin bir genel metrik sınıfında, L_k norm olarak bilinen Minkowski metriktir. Böylece L_2 norm Öklit uzaklıktır.

$$L_k(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^n |a_i - b_i|^k \right)^{1/k} \quad (3.19)$$

Buna göre;

- L_1 norm: Manhattan (city block) uzaklık

- L_2 norm: Öklit uzaklık
 - L_∞ norm: Maksimum uzaklık
- olarak adlandırılmaktadır.

Aşağıdaki çizelge 3.1.'de bazı mesafe ölçütleri ve ona karşılık gelen bozulmayı en küçük yapan optimum kod vektörleri verilmiştir.

Çizelge 3.1. merkez noktaları için mesafe ölçütleri ve optimum kod vektörleri

Kullanılan metrik fonksiyon	Bozulmayı minimum yapan optimum kod vektörü C_n
<p>Euclidean Uzaklığı:</p> $d_e(x, y) = \ x - y\ $	$C_n = \text{minimize (MSE)}$ $= \text{mean}\{x_i\} = \frac{1}{M} \sum_{k=1}^M x_k$
<p>Manhattan (City-Block) Uzaklığı:</p> $d_{cb}(x, y) = \sum_{i=1}^N x_i - y_i $	$C_n = \text{minimize (MAE)} = \text{median}\{x_i\}$ $= \{x \mid P(x(j) < x(i)) = P(x(j) > x(i))\}$
<p>Maximum (Chebychev) Uzaklığı:</p> $d_m(x, y) = \max_{i=1}^N \{ x_i - y_i \}$	$C_n(i) = \left\{ \min_{j=1}^M x_j(i) + \max_{j=1}^M x_j(i) \right\} / 2$ <p>$C_n(i)$: C_n vektörünün i'inci bileşeni</p>
<p>Hamming Uzaklığı:</p> $d_h(x, y) = \frac{1}{n} \sum_{i=1}^N \delta(x_i - y_i)$ $\delta(x_i - y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases}$	$C_n(i) = \text{mod}(x_k(i))$ $\{x_k(i) \mid P(x_k(i)) > P(x_l(i)), \forall l\}$

Çizelge 3.1. (Devam) merkez noktaları için mesafe ölçütleri ve optimum kod vektörleri

<p style="text-align: center;">Mahalanobis Uzaklığı:</p> $d_M(x, y) = (\mathbf{x} - \mathbf{y}) \cdot P_x^{-1} \cdot (\mathbf{x} - \mathbf{y})^T$ <p style="text-align: center;">Burada P_x^{-1} X'in Kovaryans matrisinin tersidir.</p>	$C^T = \frac{P_{x1}^{-1}x_1^T + P_{x2}^{-1}x_2^T + \dots + P_{xM}^{-1}x_M^T}{P_{x1}^{-1} + P_{x2}^{-1} + \dots + P_{xM}^{-1}}$ $C_n = \left[\frac{\sum_{j=1}^M C_{xj}^{-1} x_j^T}{C_{xj}^{-1}} \right]^T$
--	--

3.2.1. Öklit Uzaklık (Euclidean Distance):

Minkowsky uzaklık olarak bilinen ve

$$d(x, y) = \left[\sum_{i=1}^n |x_i - y_i|^r \right]^{1/r} \quad (3.20)$$

eşitliği ile verilen fonksiyonun özel bir halidir. Ayarlanabilir parametre adı verilen r değişkeninin $r = 2$ değeri için Minkowsky uzaklık, öklit uzaklık fonksiyonuna dönüşür.

$$\begin{aligned} d_e(x, y) &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \\ &= \left[\sum_{i=1}^n (x_i - y_i)^2 \right]^{1/2} \\ &= \left[(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) \right]^{1/2} \\ &= \|\mathbf{x} - \mathbf{y}\| \end{aligned} \quad (3.21)$$

Burada $\|\cdot\|$ operatörü matematiksel anlamda normu göstermektedir. Literatürde Euclidean uzaklık; $\|\mathbf{x} - \mathbf{y}\|^2$ olarak gösterilir ve MSE uzaklıkta denir.

3.2.2. Manhattan Uzaklık (City-block veya Mutlak Değer Uzaklık):

Manhattan Uzaklık

$$d_{cb}(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (3.22)$$

eşitliği ile verilir. Yani x ve y vektörlerinin karşılıklı bileşenlerinin farkının mutlak değerleri toplamıdır.

3.2.3. Maximum Uzaklık (Chebychev Uzaklık):

$$d_m(x, y) = \max_{i=1}^n \{|x_i - y_i|\} \quad (3.23)$$

eşitliği ile verilir. Yani x ve y 'nin karşılıklı bileşenleri farklarının mutlak değerce en büyüğüne eşittir.

3.2.4. Hamming Uzaklık:

$$d_h(x, y) = \frac{1}{n} \sum_{i=1}^n \delta(x_i - y_i) \quad (3.24)$$

eşitliği ile verilir. Burada;

$$\delta(x_i - y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \quad (3.25)$$

Bölüm 3.2.'te Çizelge 3.2.1.'de verilen farklı uzaklık fonksiyonlarına göre bozulmayı minimize eden optimum C_n vektörlerinin bulunuşuna ait ispatlar EK-1, EK-2, EK-3 ve EK-4'te sırasıyla gösterilmektedir.

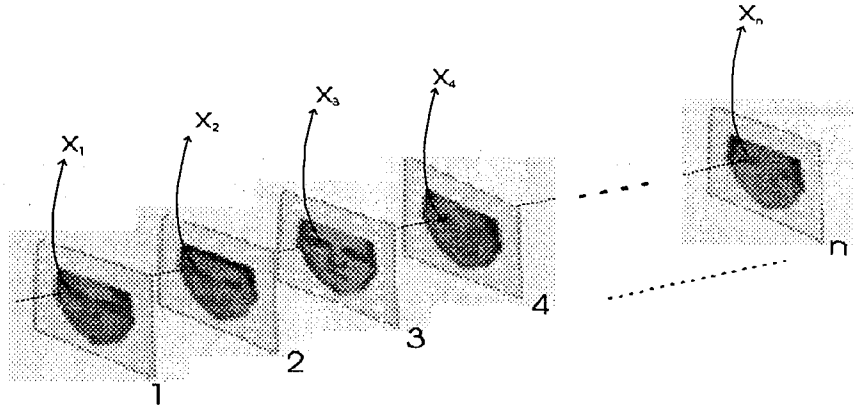
3.3. Yüz Organlarının LBG VQ ile Kodlanması

Bu bölümde ilk olarak VQ (Vektör Nicemlemesi) için LBG algoritması kullanılarak, organ resimleri uzayındaki merkez noktalar tespit edildi. LBG VQ metodu, optimallik kriterine bir çözüm sunan özyineli bir algoritmadır. Algoritma için bir başlangıç kod vektörü $\mathbf{c}^{(0)}$ alınır. Bu vektör için bütün örnek kümenin ortalaması ile başlanabilir. Bu başlangıç kod vektörü ikiye bölünüp, elde edilen her bir vektörün optimallik şartını sağlayacak şekilde Lloyd-Max algoritması uygulanır. Ardından tekrar her bir vektör ikiye bölünerek, bu vektörlere de Lloyd-Max algoritması uygulanır. Bu son iki kod vektörde dörde bölündükten sonra, işlem istenilen kod vektör sayısına ulaşılan kadar tekrar ettirilir. Kod vektörünün ikiye bölünmesi sırasında orijinal kod vektöre küçük bir vektör eklenir ve çıkarılırsa iki ayrı vektör elde edilir:

$$\begin{aligned}\mathbf{c}_2^{(1)} &= \mathbf{c}^{(0)} + \varepsilon \mathbf{v}_e \\ \mathbf{c}_1^{(1)} &= \mathbf{c}^{(0)} - \varepsilon \mathbf{v}_e\end{aligned}\tag{3.26}$$

Burada ε , eklenen vektörün küçük olmasını sağlamaktadır. \mathbf{v}_e vektörü olarak ise, $\mathbf{c}^{(0)}$ vektörü ile temsil edilen bölgedeki en büyük değişimin bulunduğu doğrultuya karşılık gelen vektör seçilmelidir. Tanım gereği bölge içindeki kovaryans matrisinin en büyük özdeğerine karşılık gelen özvektör, bize \mathbf{v}_e vektörünü vermektedir. Bu şekilde elimizdeki kod vektör, resimleri eskisinden daha kötü temsil etmeyecek iki ayrı vektöre bölünür. İstenilen sayıya ulaşıldığında özyineli algoritma son bulur.

3.3.1. Vektörlerin Oluşturulması



Şekil 3.3. Yüz resimlerinden çıkarılan aynı piksel koordinatlara sahip göz bölgesinden X vektörünün elde edilmesi

Şekil 3.3.'te görüldüğü gibi VQ için gerekli olan orta noktaları bulurken her organ resminde aynı koordinatlardaki piksellerin değerleri alınarak bir X vektörü oluşturulur, bu vektör kullanılarak gerekli orta noktalar hesaplanır.

Çizelge 3.2. orta nokta ve karşılık gelen voronoi hücresi

	Orta nokta	Voronoi Hücresi
$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = P \Rightarrow$	ortanca : ortanca (X)	Minimize (MAE)
	ortalama : \bar{X}	Minimize (MSE)
	min - max : $\frac{\max \{x_m\} + \min \{x_m\}}{2}$	Minimize $\max(x_m - y_i)$

Arama problemi sorgu resmi olan P'nin, veritabanındaki resimlerden uzaklık fonksiyonu $d(P, x_m)$ 'yi minimize eden m^{nci} resim x_m 'i bulmaya eşittir. Bu çalışmada min-max yöntemi sonucunda elde edilen VQ resimlerinin en bulanık sonuçları verdiği görüldü. Aynı şekilde Hamming uzaklığını minimize eden $\{\text{mod}(x_m)\}$ işlemi de iyi sonuç vermedi. Bu nedenle çalışmalarımızı ortalama ve ortanca yöntemlerine [18] yoğunlaştırdık.

4. GERİ KALAN YÜZ RESMİNİN SIKIŞTIRILMASI

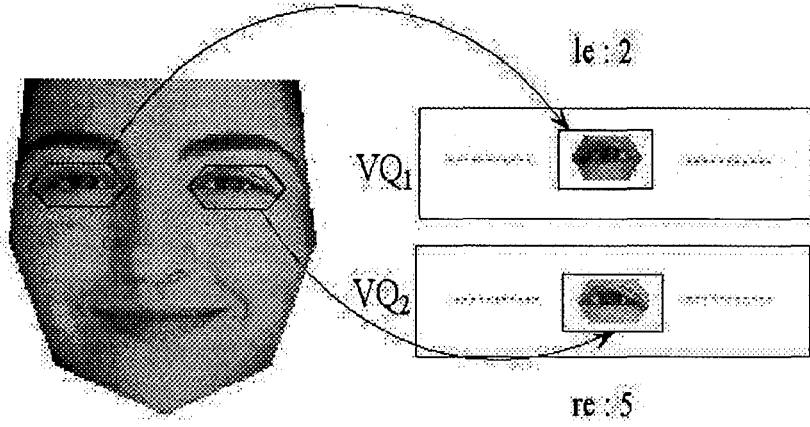
Yüz resminin bütünüyle kodlanmış halini elde etmek için son basamak olarak yüze ait organ bölütlerinin çıkarıldıktan sonra geri kalan yüz resminin sıkıştırılması gerekmektedir. Organlar çıkarıldıktan sonra geri kalan yüz resmi aradeğerleme işleminden kaynaklanan düzgün varyasyonlar içermektedir. Sıkıştırma için SPIHT[19] VE LOCO[20] programlarının kayıplı modları kullanılarak sonuçlar elde edilmiştir. SPIHT ve LOCO EK-5 VE EK-6'da ayrıntılı olarak anlatılmaktadır. Bu çalışmada yüz resimleri direkt olarak hem SPIHT hem de LOCO ile kodlandı ve bu kodlama sonuçları, organ resimlerinin vektör nicemlenmesi ve geri kalan yüz resminin SPIHT ve LOCO ile ayrı ayrı kodlanmasından elde edilen sonuçlarla karşılaştırıldı. Karşılaştırılan bütün kodlamalarda bit hızlarının aynı olması için sıkıştırma parametreleri ayarlanmıştır. Böylece, sonuçlara ait ortalama karesel hatalar karşılaştırılmıştır. Bölüm 6.'da bu sıkıştırma sonuçları ve ortalama karesel hatalar verilmektedir.

5. VERİ TABANINDAN SORGU RESMİNİN ARANMASI

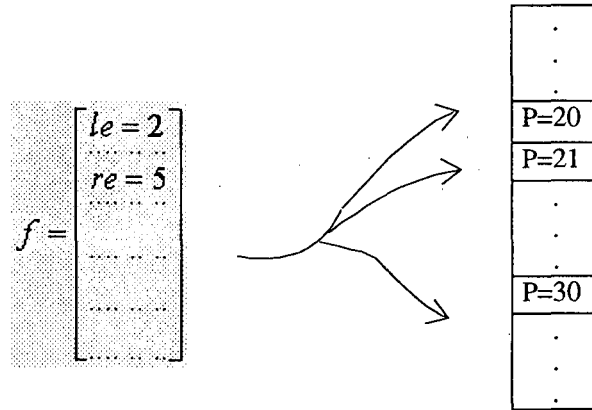
Anlatılan kodlama yapısı aynı zamanda sorgulamayada uygun olarak ortaya çıkmıştır [2,4]. Bölgeleştirme sonucunda sadece organ bölgelerinin VQ numaraları kullanılarak yapılan bir ön tarama ile sorgulama cevabına aday organ resimleri tespit edildikten sonra, bu resimlerin arasından arka fon (yüzün geri kalanı) karşılaştırması ile en iyi uyum yapan resim bulunmaktadır. Sıkıştırılmış görüntülerden oluşan bir veri tabanında bir yüz görüntüsünün en çok benzediği resmi bulmak için şu basamaklar kullanılmaktadır:

- 1) Sorgulanan yüz görüntüsündeki organlar bölgeleştirilir.
- 2) Bu organlardan her birisinin, kendi grubu içinde en yakın olduğu görüntü vektörü tespit edilir.
- 3) Her bölge için elde edilen vektör numaraları, ayrı bir veri tabanında bulunan ve her resmin organlarının vektör numaralarını gösteren tablo ile karşılaştırılarak, sorgulanan ile uyumlu olması muhtemel, ve tüm resimlerin sayısından daha az sayıda bir yüz görüntüsü kümesi elde edilir.
- 4) Küme içinde yüz arka planına ait resimler karşılaştırılarak sonuç elde edilir.

Şekil 5.1.'de örnek bir sorgu resminin sorgulama basamaklarının bir kısmı gösterilmektedir. Buna göre sorgu resminin sol gözü, VQ ile kodlanmış sol göz veritabanında 2 nolu sol göz kod çizelgesine en yakın ($le=2$), sağ göz resmi ise sağ göz veri tabanında 5 nolu sağ göz kod çizelgesine en yakın ($re=5$) bulunmaktadır. Bu işlem diğer yüz organları içinde ayrı ayrı yapılır. Sorgu resminin en yakın olduğu organ numaraları f vektörü içinde toplanır. P matrisinin her bir satırı ise veri tabanındaki bir yüze ait VQ ile kodlanmış organ numaralarını içermektedir. Sonra f vektörünün P matrisinin her bir satırına olan Hamming uzaklığına $\left(d(x, f) = \begin{cases} 0, & x = f \\ 1, & x \neq f \end{cases} \right)$ bakılır. Hamming uzaklığı en az olan satır sorgu resminin yüz organlarına karşılık gelen VQ ile kodlanmış organ numaralarını içerir. Buna göre sorgulanan resim artık bu organ numaralarıyla oluşturulur.



Şekil 5.1. Örnek bir sorgu resminin sorgulanması



Şekil 5.2. f vektörünün P vektörüne olan Hamming uzaklığının bulunması

Çizelge 5.1. Örnek sorgu resminin en benzer resimle karşılaştırılması

Ait Olduğu VQ Kümesi	Sağ Kaş	Sol Kaş	Sağ Göz	Sol Göz	Burun	Dudak
VQ kümesi içinde bulunan en yakın organ VQ numaraları	2	3	3	1	5	1
Hamming uzaklığın en az olduğu yüze ait organ numaraları	2	1	3	1	1	1
Hamming Uzaklık	0	1	0	0	1	0
Toplam Hamming Uzaklık	2					

Yukarıdaki Çizelge 5.1.'de sorgu resminin organlarının vektör nicemleme yöntemi ile kodlanmış olan örnek yüz organları arasında en benzer yüz organlarına sahip olan yüzün ait olduğu vektör nicemleme kümesindeki organ numaraları verilmiştir. Burada Hamming uzaklığının 2 olduğu görülmektedir


6. DENEYSEL ÇALIŞMALAR

Elde edilen deneysel sonuçlara göre, yüzden organ kısımlarının çıkarılıp yerlerinin interpolate edilmesi sonucunda resmin entropisinde %3-4 civarında azalma meydana gelmiştir. Organlara ait resimlerin VQ ile sıkıştırılması sonucunda elde edilen vektör tablosu, ve organ bölgelerinin yüz üzerindeki koordinatlarının kapladığı yer, geri kalan yüze ait veriye nazaran ihmal edilebilecek kadar küçüktür. Bu durumda yüzün kalanının iyi kodlanması, toplam sıkıştırma oranını önemli ölçüde belirleyecektir. Bu çalışmada kalan yüz resminin entropisinin düşürüldüğü göz önünde bulundurulursa, ne tür yöntem kullanılırsa kullanılsın, organları ile birlikte bulunan yüz resmine nazaran daha fazla sıkıştırma oranı elde edileceği tahmin edilmektedir.

6.1. Sıkıştırma Sonuçları

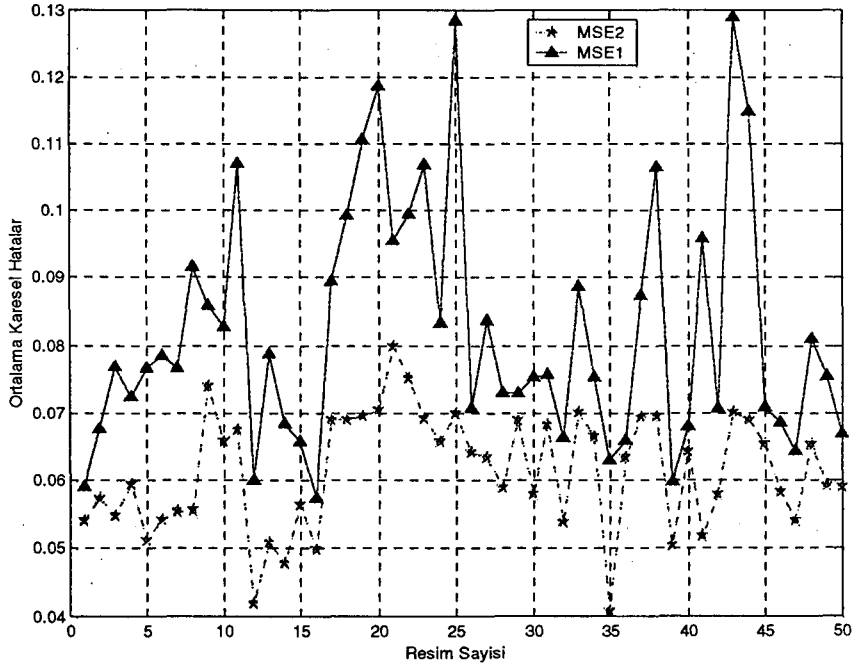
Bu bölümde gösterilmiş olan tüm ortalama karesel hata grafiklerinde mavi düz çizgi ve üçgenlerle gösterilenler kendi yöntemizle kodlanmış olan yüzlerin ortalama karesel hataları gösteriyor, kırmızı kesik çizgi ve yıldızlarla gösterilenler orijinal yüz resminini kodlanmasıyla elde edilen yüzlerin ortalama karesel hatalarını veriyor. Aşağıdaki şekil ve tablolarda gösterilen VQ1; bölüm 3.1.3.'te gösterilen LBG VQ'dur, VQ2 ise VQ1'den farklı olarak kod vektörünün ikiye bölünmesinde bir kriter daha esas alınmıştır ki bu da; bölünecek kod vektörün bulunmasında kullanılmıştır. Kod çizelgesindeki en büyük öz değerli kovaryans matrisine sahip olan kod vektör en kalabalık olan kod vektörün yarısından daha az eleman sayısına sahip olmamak şartıyla ikiye bölünür, diğer durumda en kalabalık eleman sayısına sahip olan kod vektör ikiye bölünür ve istenilen sayıda kod vektör elde edilene kadar işlem özyineli olarak devam eder.

Çizelge 6.1. VQ1a ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması

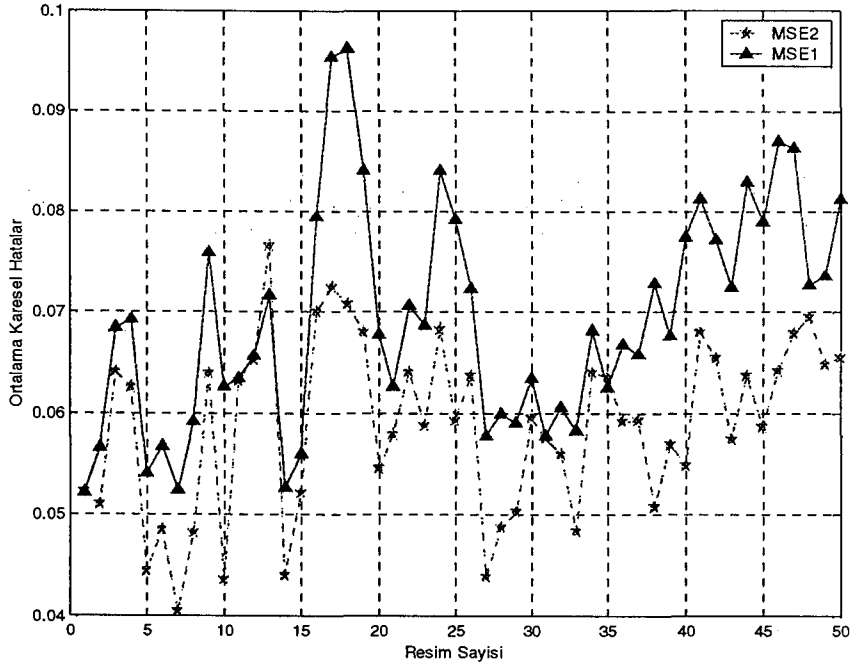
Özgün Görüntü	Özgün Görüntü + LOCO-I BR≈1.5	Özgün Görüntü + Spiht BR≈1.5	VQ1a + LOCO-I BR≈1.5	VQ1a + Spiht BR≈1.5	Özgün Görüntü + LOCO-I BR≈0.6	Özgün Görüntü + Spiht BR≈0.6	VQ1a + LOCO-I BR≈0.6	VQ1a + Spiht BR≈0.6
								
								
								
								
								
								
								
								
								
								

Çizelge 6.2. kodlanmış resimlerin bit sayılarının gösterilmesi

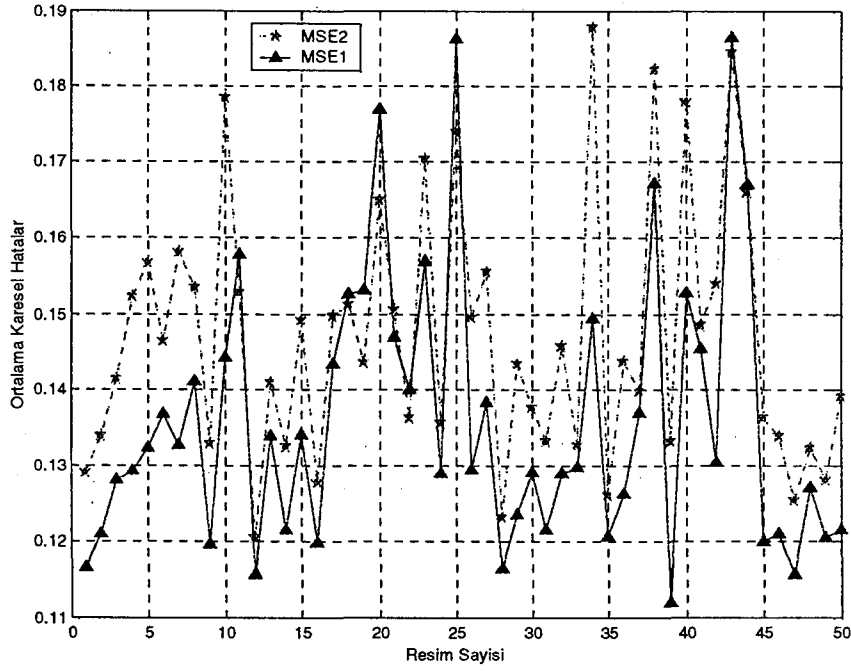
Resim	BR≈0,6 iken		Erkek Yüz Resimleri için		Bayan Yüz Resimleri için	
	Kodlanmamış yüz resminin toplam bit sayısı	LOCO-I ve Spiht ile Kodlanmış Geri Kalan Resim	LOCO-I ve Spiht ile Kodlanmış Orijinal Resim	LOCO-I ve Spiht ile Kodlanmış Geri Kalan Resim	LOCO-I ve Spiht ile Kodlanmış Orijinal Resim	
Pic1	153600	11296	11376	10992	11280	
Pic2	153600	11400	11392	11760	11400	
Pic3	153600	11704	11176	11008	11448	
.	
.	
.	
Pic48	153600	11848	11232	11072	11456	
Pic49	153600	11400	11104	11344	11200	
Pic50	153600	11272	11200	11800	11104	
Ortalama Değerler	153600	11443,04	11340,8	11419,2	11339,2	



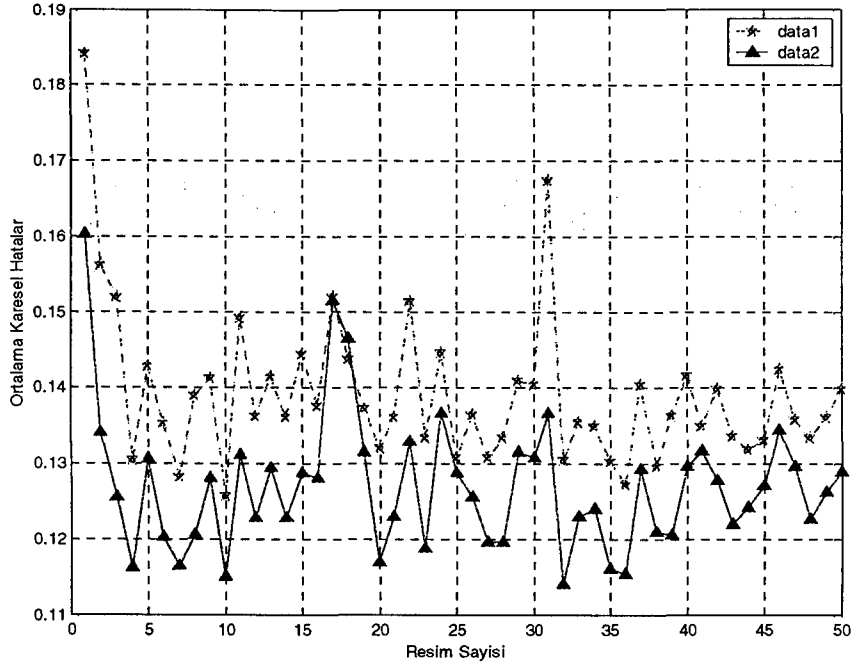
Şekil 6.1. BR≈0.6, LOCO-I yöntemi, bayan yüz resimleri VQ1a



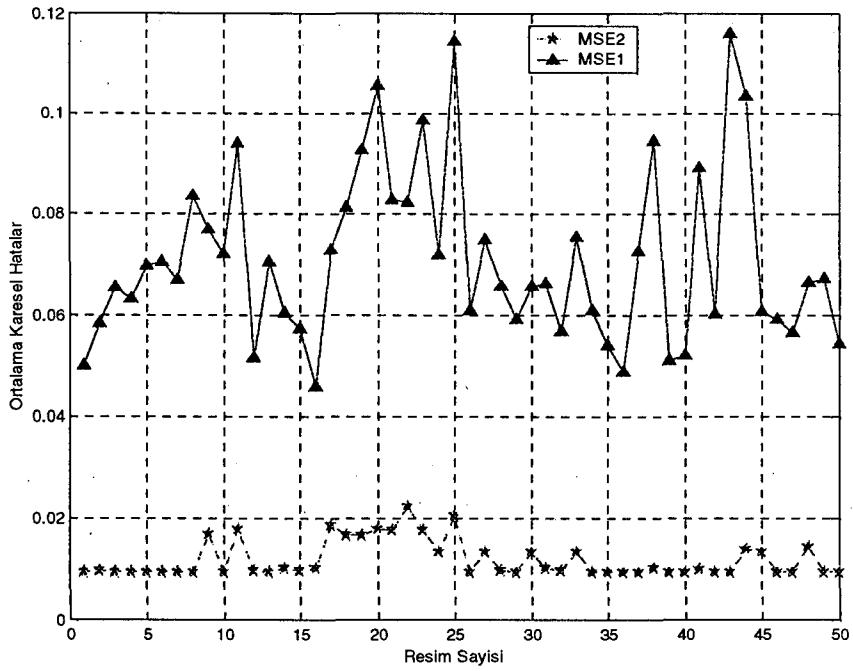
Şekil 6.2. BR≈0.6, LOCO-I yöntemi, erkek yüz resimleri, VQ1a



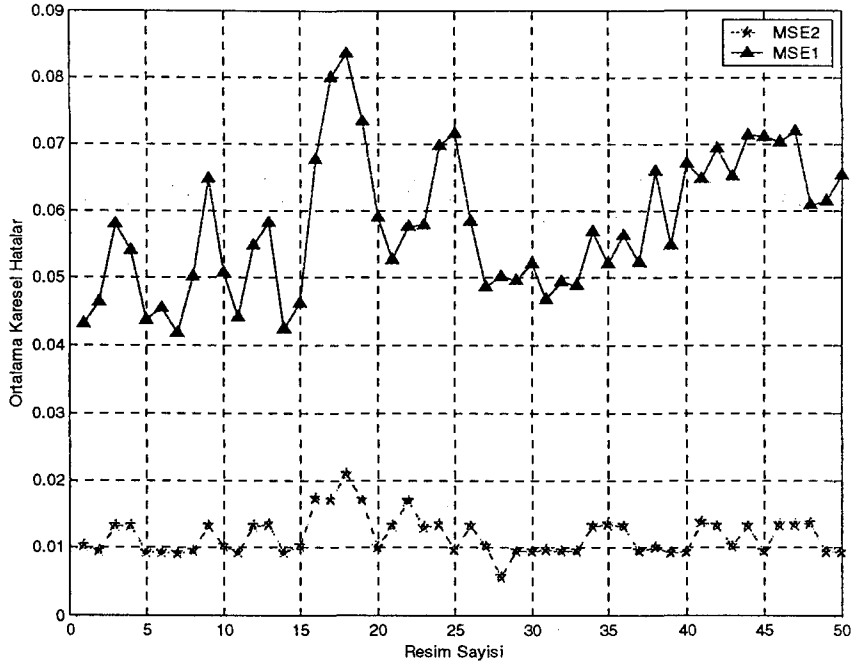
Şekil 6.3. BR≈0.6, Spiht yöntemi, bayan yüz resimleri, VQ1a



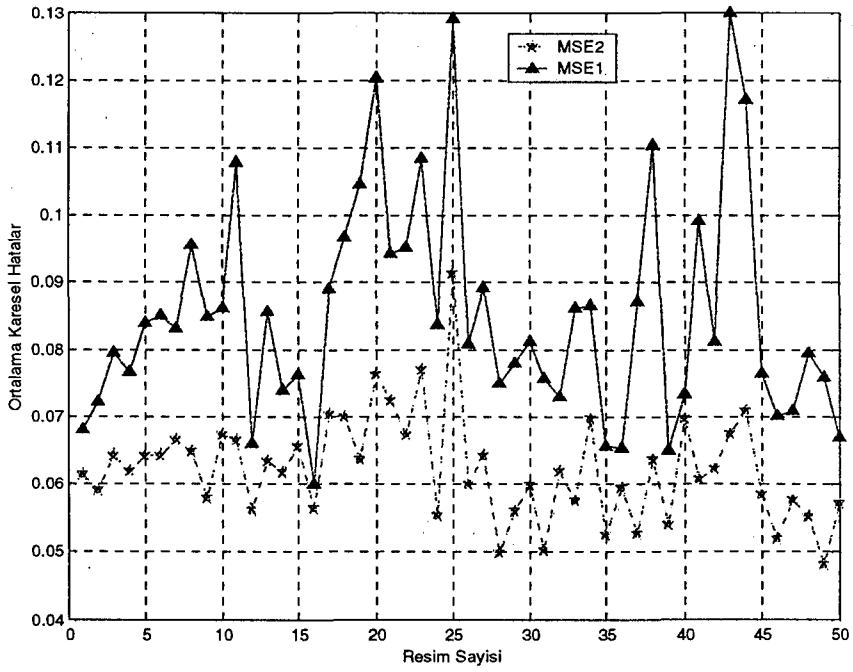
Şekil 6.4. $BR \approx 0.6$, Spiht yöntemi, erkek yüz resimleri, VQ1a



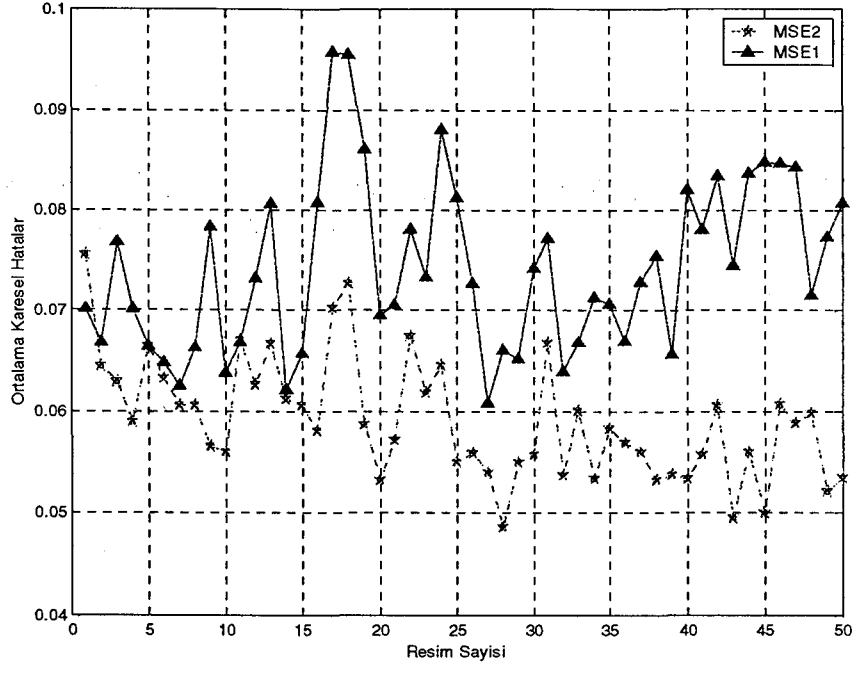
Şekil 6.5. $BR \approx 1.5$, LOCO-I yöntemi, bayan yüz resimleri, VQ1a



Şekil 6.6. $BR \approx 1.5$, LOCO-I yöntemi, erkek yüz resimleri, VQ1a



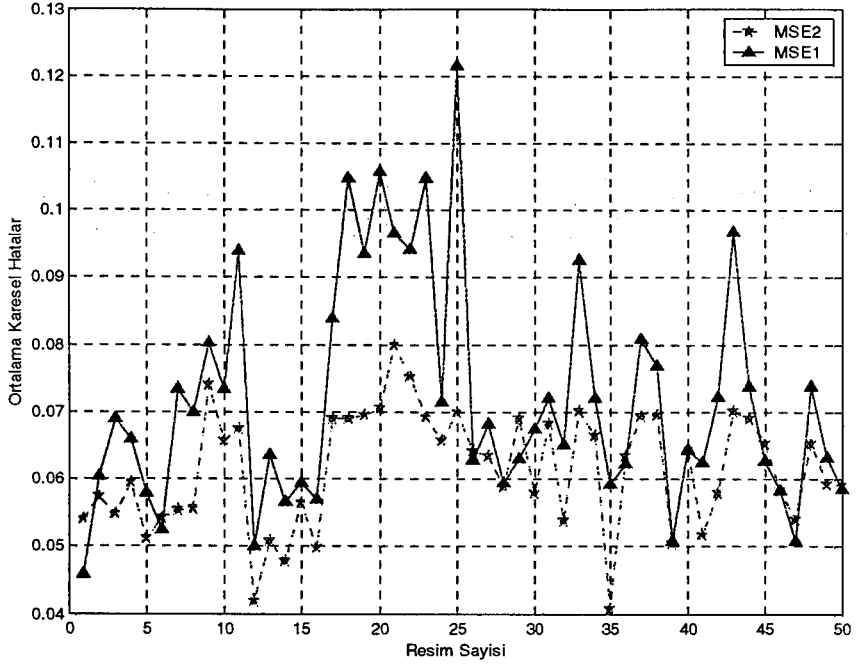
Şekil 6.7. $BR \approx 1.5$, Spiht yöntemi, bayan yüz resimleri, VQ1a



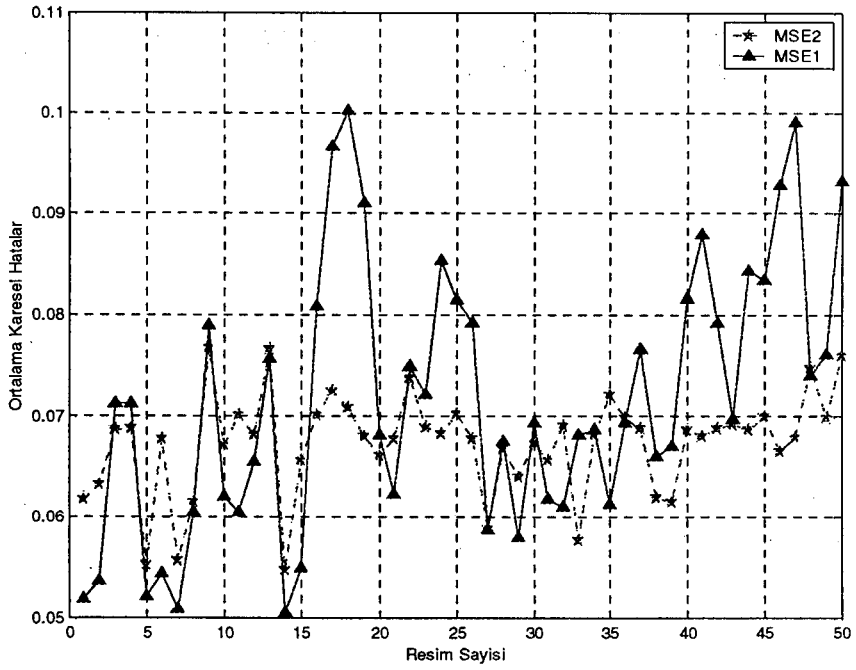
Şekil 6.8. $BR \approx 1.5$, Spiht yöntemi, erkek yüz resimleri, VQ1a

Çizelge 6.3. VQ1b ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması

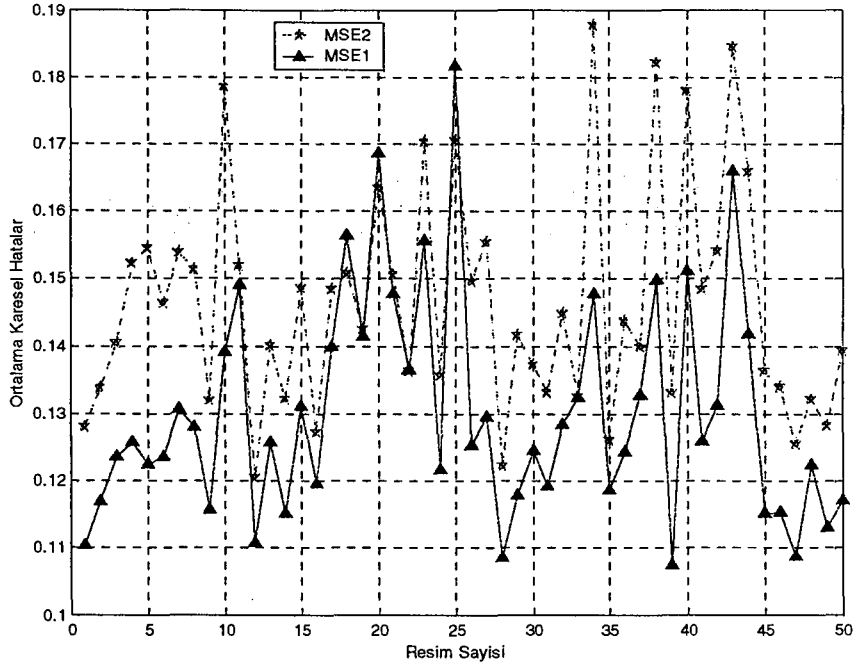
Özgün Görüntü	Özgün Görüntü + LOCO-I BR≈1.5	Özgün Görüntü + Spiht BR≈1.5	VQ1b + LOCO-I BR≈1.5	VQ1b + Spiht BR≈1.5	Özgün Görüntü + LOCO-I BR≈0.6	Özgün Görüntü + Spiht BR≈0.6	VQ1b + LOCO-I BR≈0.6	VQ1b + Spiht BR≈0.6
								
								
								
								
								
								
								
								
								
								



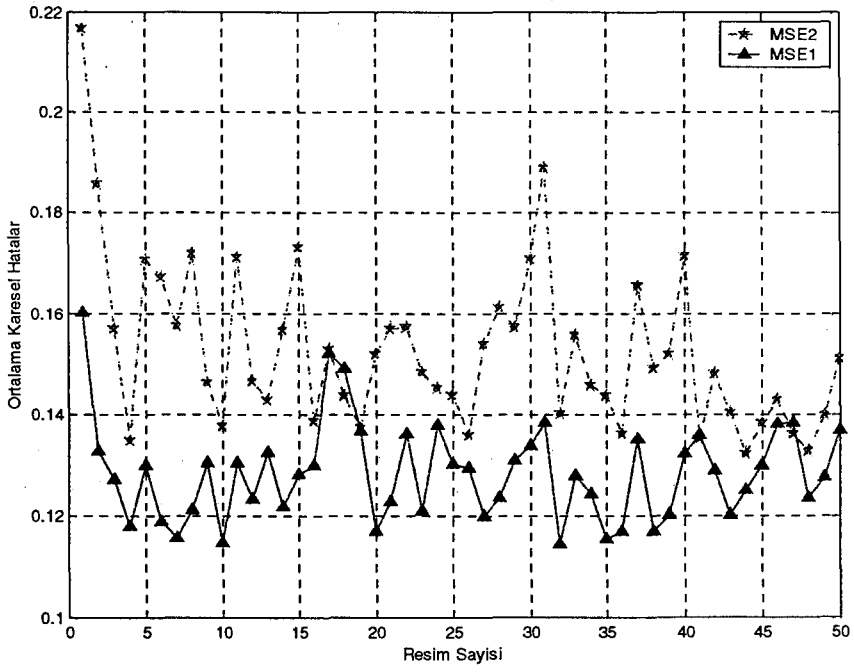
Şekil 6.9. $BR \approx 0.6$, LOCO-I yöntemi, bayan yüz resimleri, VQ1b



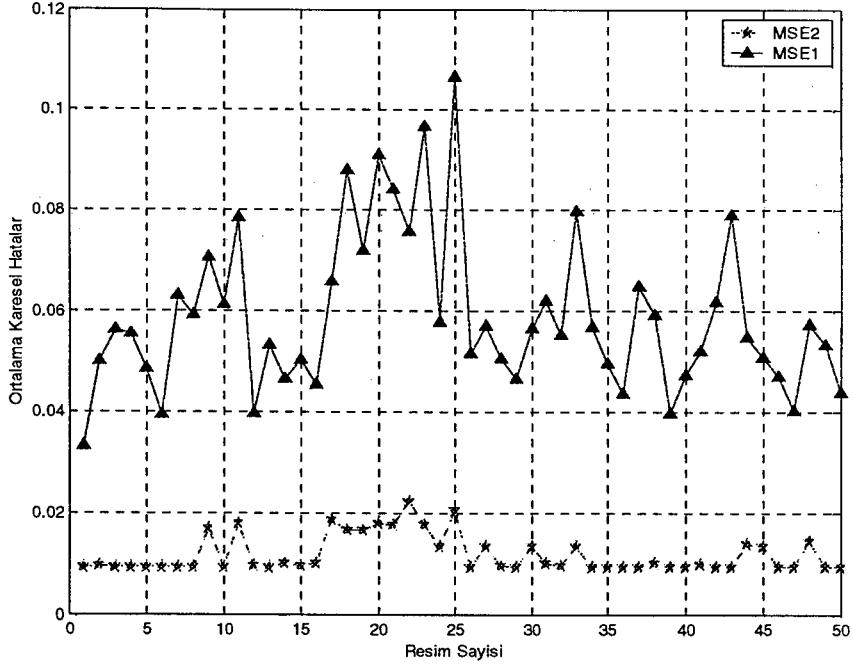
Şekil 6.10. $BR \approx 0.6$, LOCO-I yöntemi, erkek yüz resimleri, VQ1b



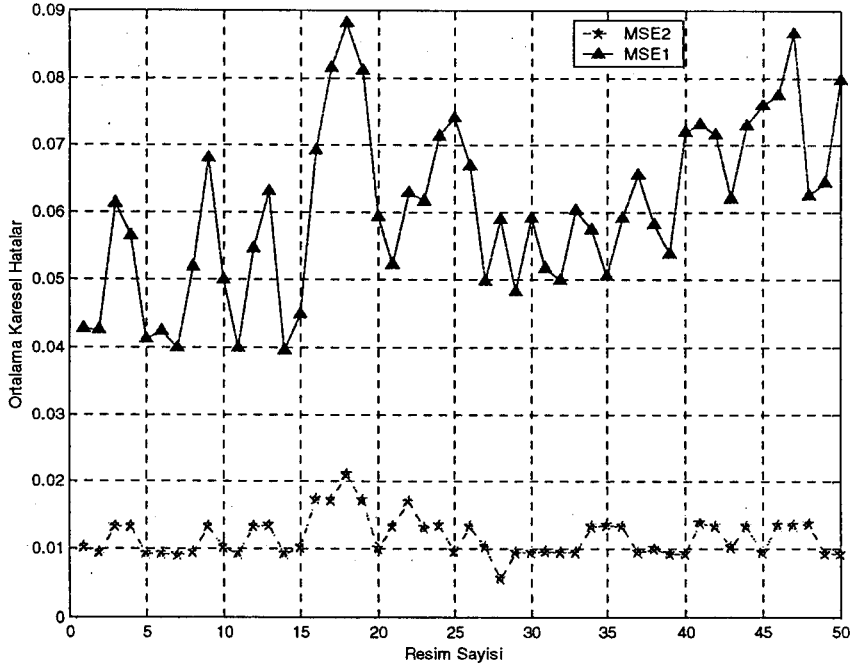
Şekil 6.11. BR≈0.6, Spiht yöntemi, bayan yüz resimleri, VQ1b



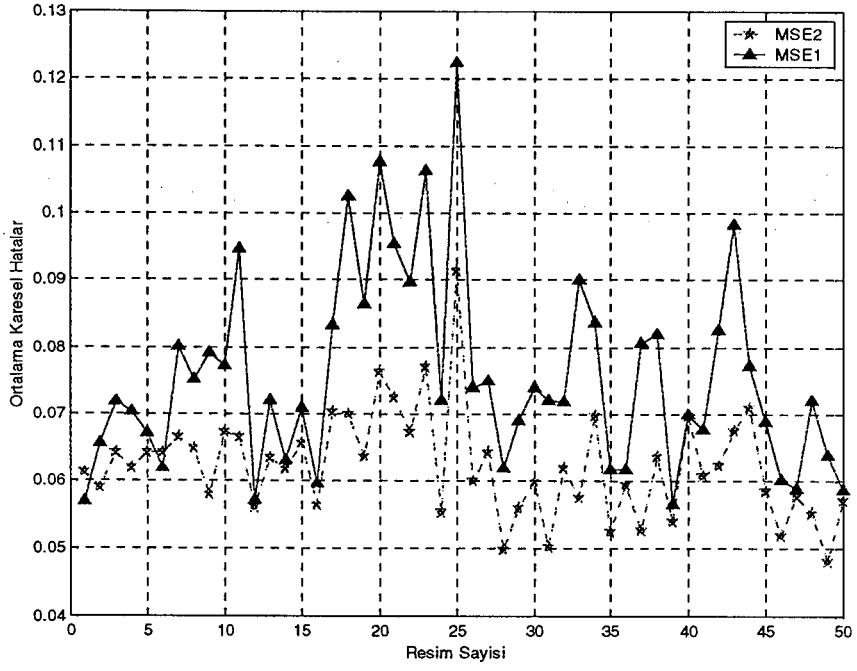
Şekil 6.12. BR≈0.6, Spiht yöntemi, erkek yüz resimleri, VQ1b



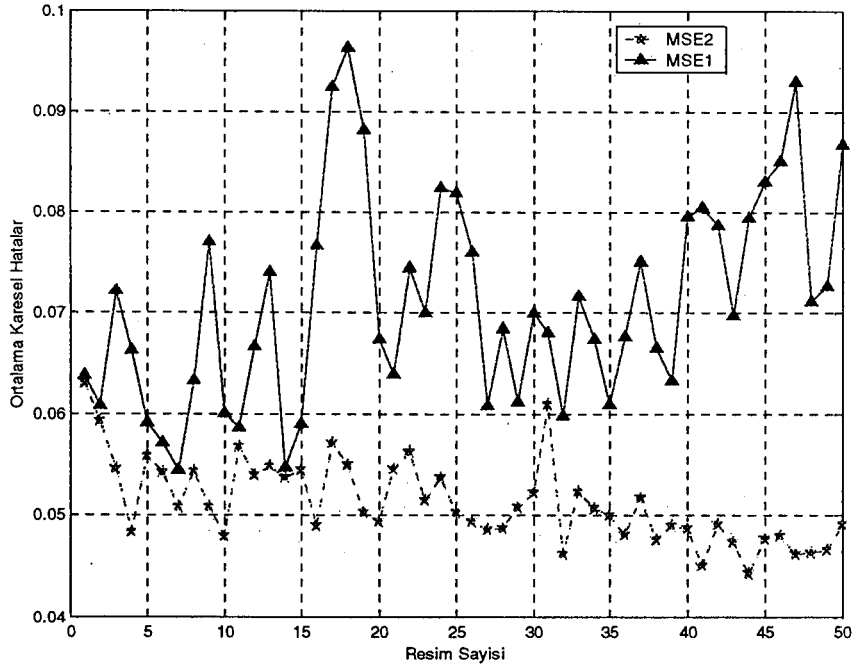
Şekil 6.13. BR≈1.5, LOCO-I yöntemi, bayan yüz resimleri, VQ1b



Şekil 6.14. BR≈1.5, LOCO-I yöntemi, erkek yüz resimleri, VQ1b



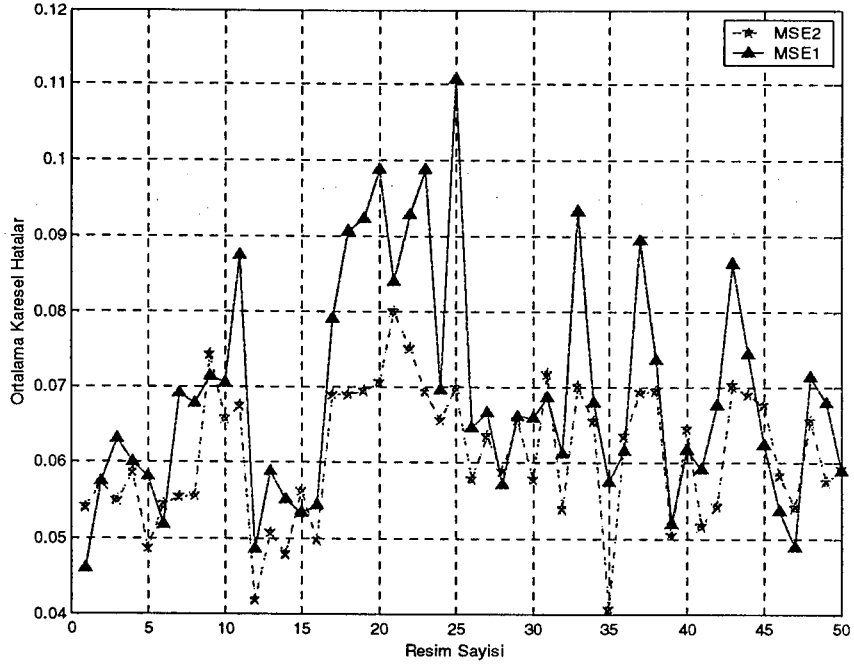
Şekil 6.15. $BR \approx 1.5$, Spiht yöntemi, bayan yüz resimleri, VQ1b



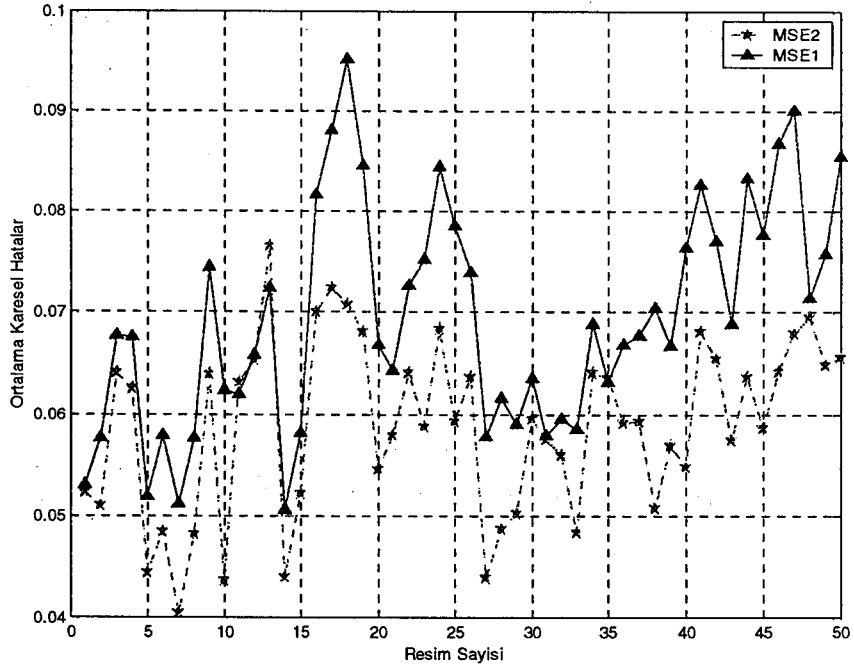
Şekil 6.16. $BR \approx 1.5$, Spiht yöntemi, erkek yüz resimleri, VQ1b

Çizelge 6.4. VQ2a ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması

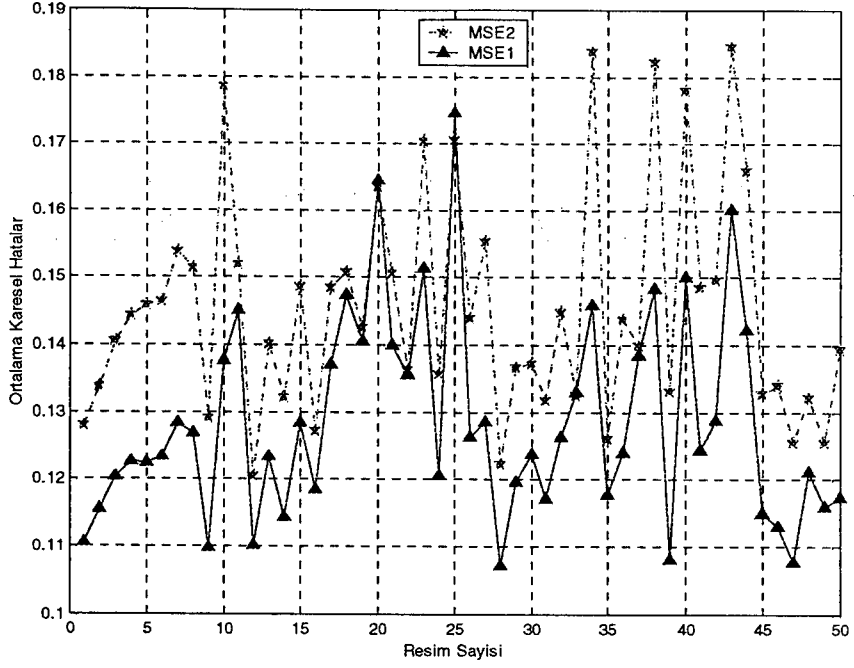
Özgün Görüntü	Özgün Görüntü + LOCO-I BR≈1.5	Özgün Görüntü + Spiht BR≈1.5	VQ2a + LOCO-I BR≈1.5	VQ2a + Spiht BR≈1.5	Özgün Görüntü + LOCO-I BR≈0.6	Özgün Görüntü + Spiht BR≈0.6	VQ2a + LOCO-I BR≈0.6	VQ2a + Spiht BR≈0.6
								
								
								
								
								
								
								
								
								
								



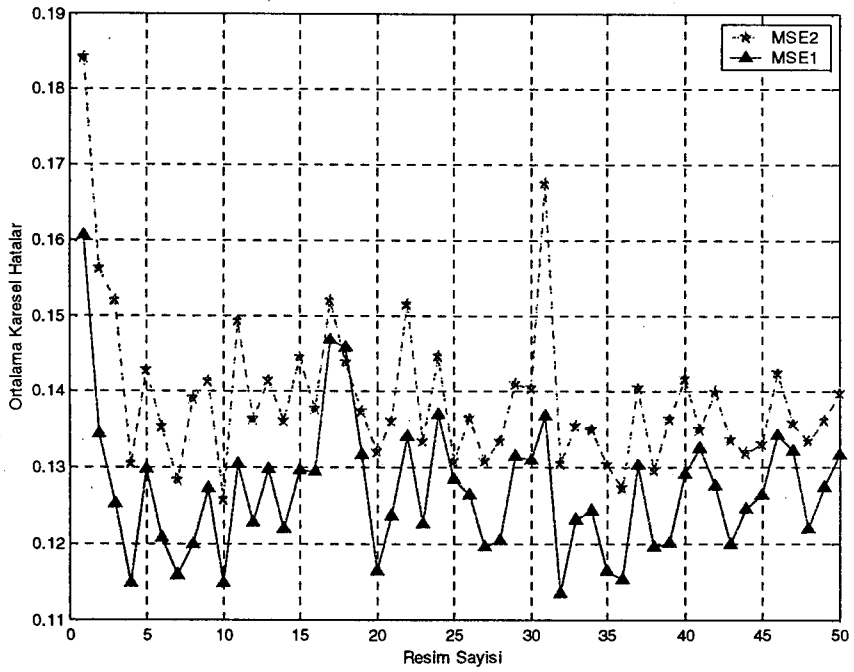
Şekil 6.17. $BR \approx 0.6$, LOCO-I yöntemi, bayan yüz resimleri, VQ2a



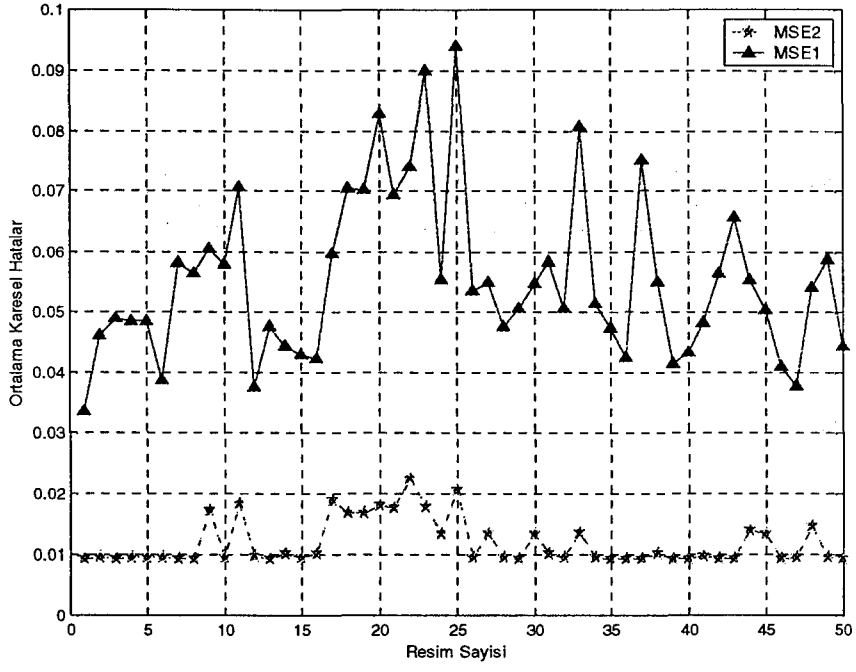
Şekil 6.18. $BR \approx 0.6$, LOCO-I yöntemi, erkek yüz resimleri, VQ2a



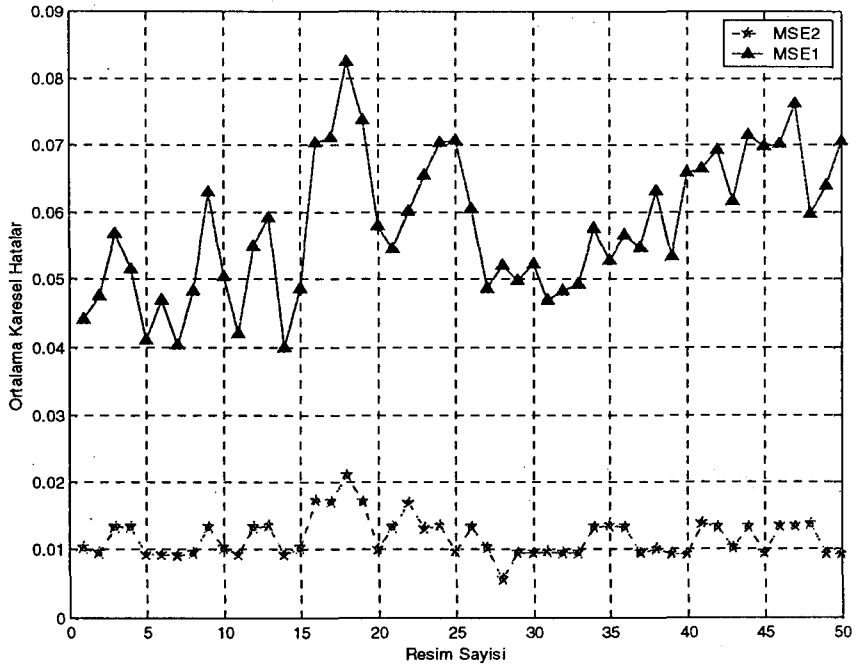
Şekil 6.19. BR≈0.6, Spiht yöntemi, bayan yüz resimleri, VQ2a



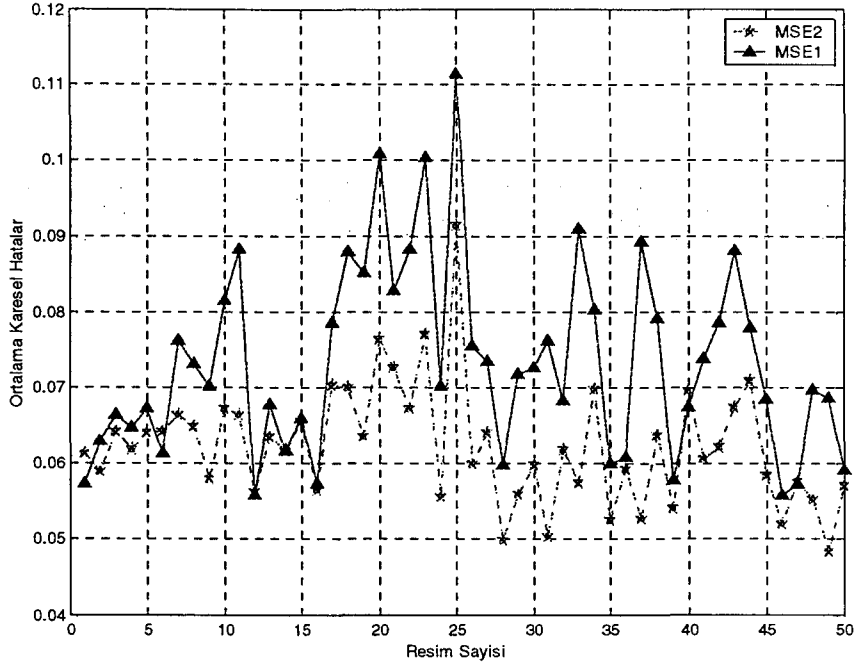
Şekil 6.20. BR≈0.6, Spiht yöntemi, erkek yüz resimleri, VQ2a



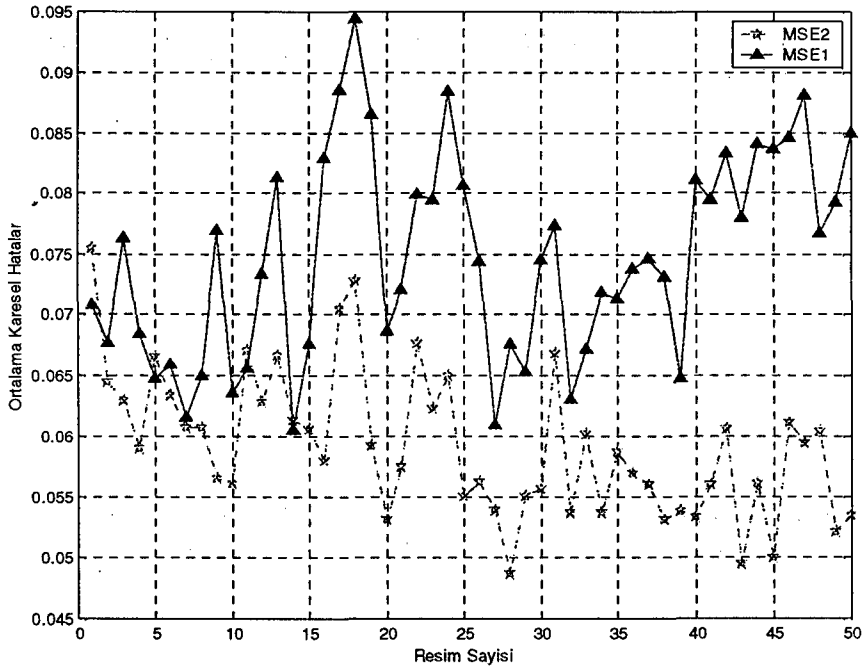
Şekil 6.21. $BR \approx 1.5$, LOCO-I yöntemi, bayan yüz resimleri, VQ2a



Şekil 6.22. $BR \approx 1.5$, LOCO-I yöntemi, erkek yüz resimleri, VQ2a



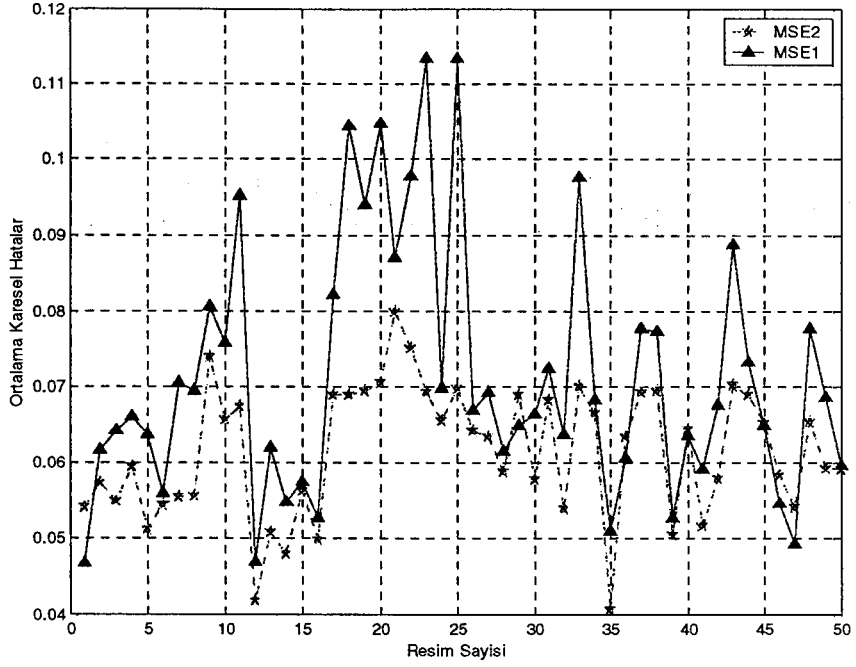
Şekil 6.23. $BR \approx 1.5$, Spiht yöntemi, bayan yüz resimleri, VQ2a



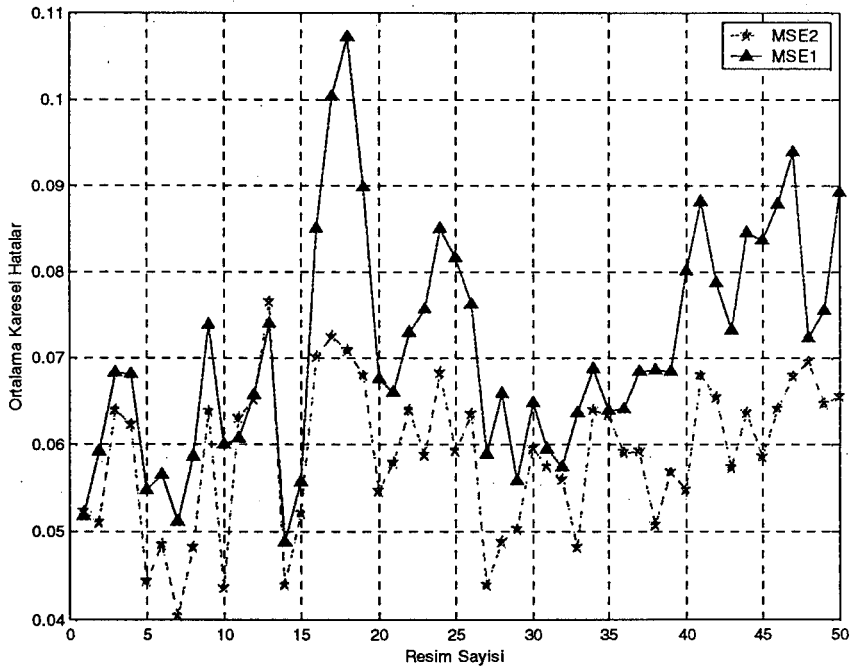
Şekil 6.24. $BR \approx 1.5$, Spiht yöntemi, erkek yüz resimleri, VQ2a

Çizelge 6.5. VQ2b ile kodlanmış organların farklı kodlanmış geri kalan yüz resimleriyle görsel karşılaştırılması

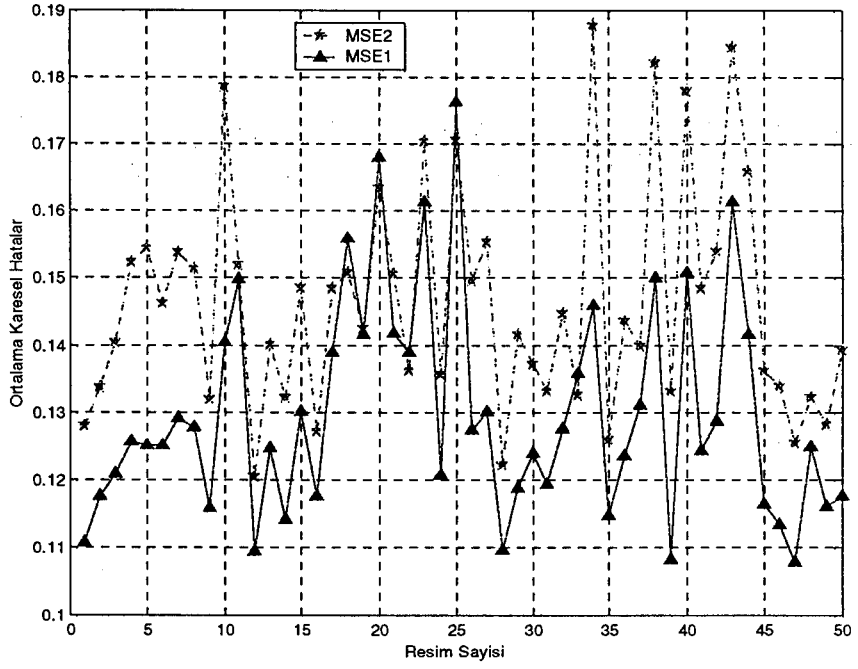
Özgün Görüntü	Özgün Görüntü + LOCO-I BR≈1.5	Özgün Görüntü + Spiht BR≈1.5	VQ2b + LOCO-I BR≈1.5	VQ2b + Spiht BR≈1.5	Özgün Görüntü + LOCO-I BR≈0.6	Özgün Görüntü + Spiht BR≈0.6	VQ2b + LOCO-I BR≈0.6	VQ2b + Spiht BR≈0.6
								
								
								
								
								
								
								
								
								
								



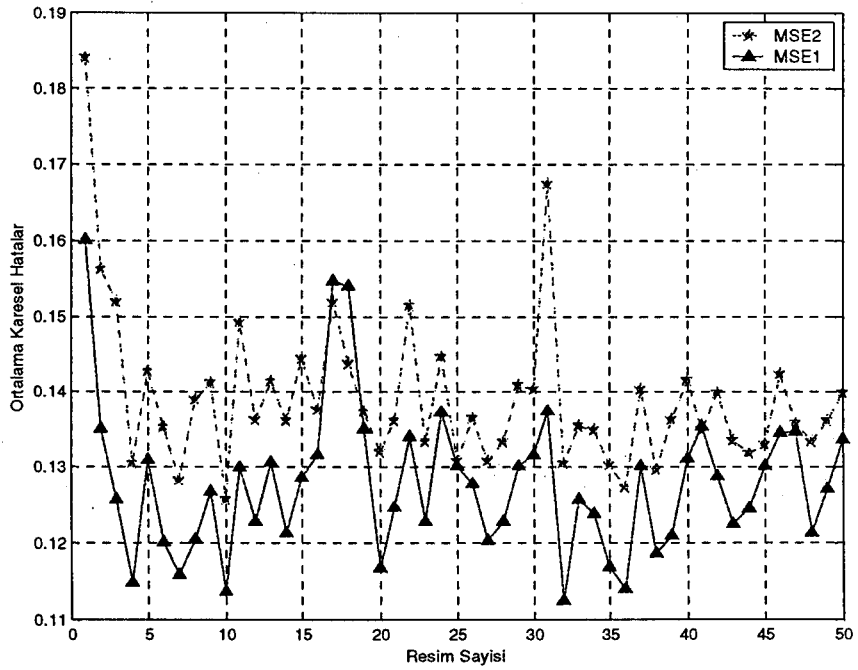
Şekil 6.25. BR≈0.6, LOCO-I yöntemi, bayan yüz resimleri, VQ2b



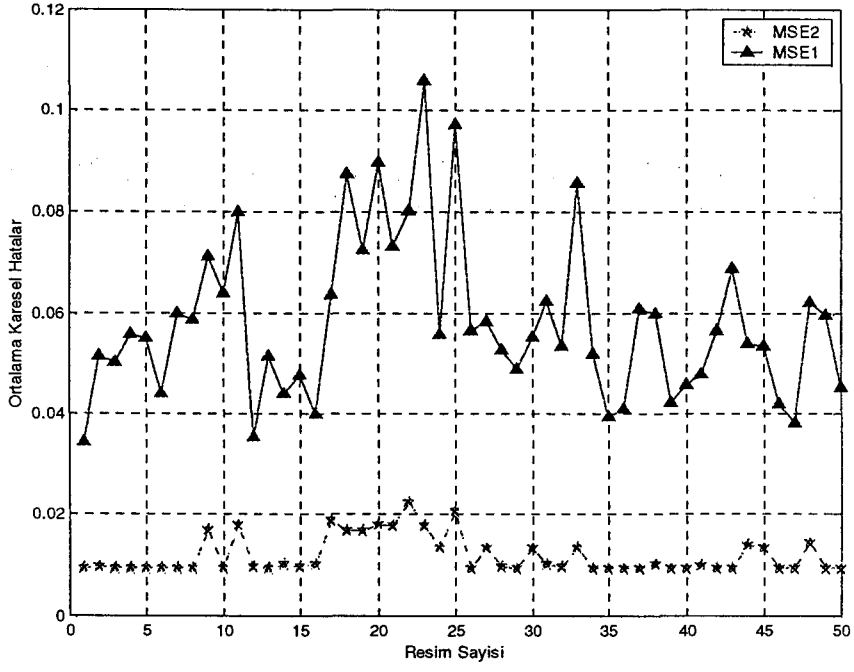
Şekil 6.26. BR≈0.6, LOCO-I yöntemi, erkek yüz resimleri +VQ2b



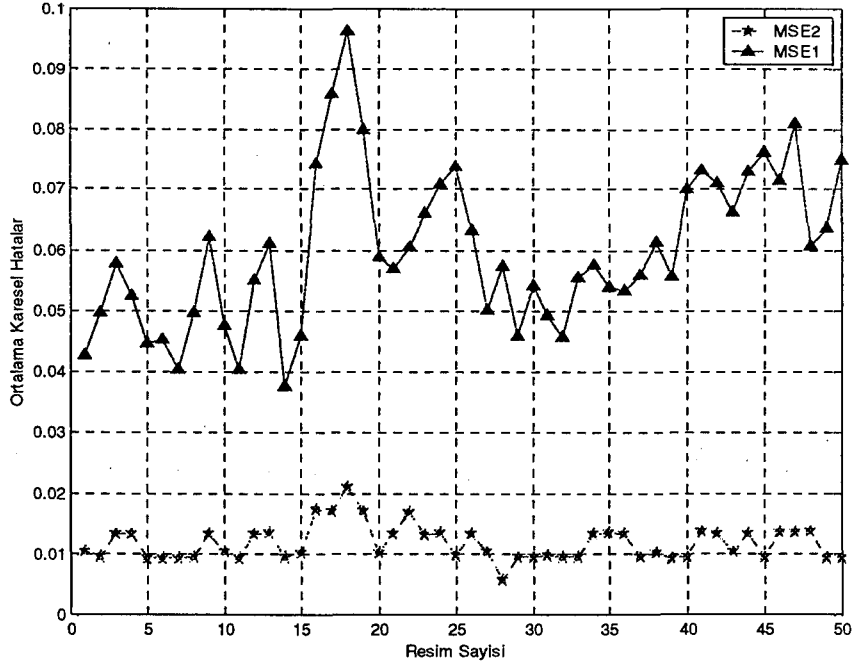
Şekil: 6.27. BR≈0.6, Spiht yöntemi, bayan yüz resimleri , VQ2b



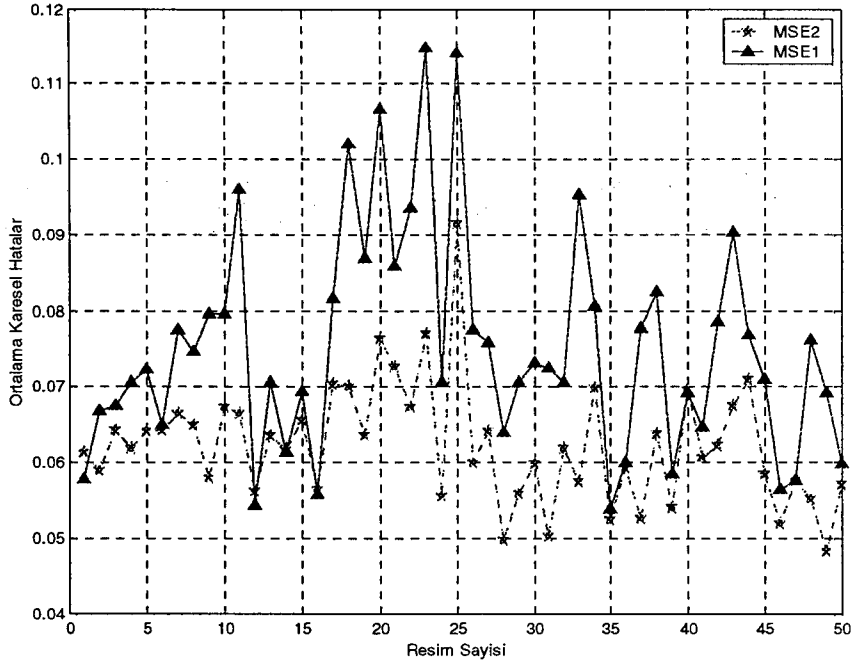
Şekil 6.28. BR≈0.6, Spiht yöntemi, erkek yüz resimleri, VQ2b



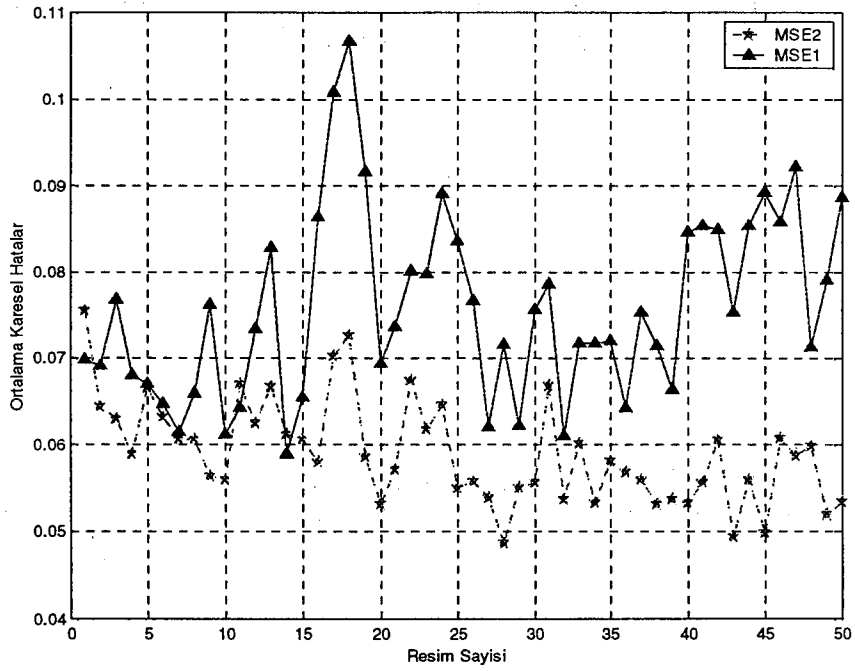
Şekil 6.29. BR≈1.5, LOCO-I yöntemi, bayan yüz resimleri, VQ2b



Şekil 6.30. BR≈1.5, LOCO-I yöntemi, erkek yüz resimleri, VQ2b



Şekil 6.31. BR≈1.5, Spiht yöntemi, bayan yüz resimleri, VQ2b



Şekil 6.32. BR≈1.5, Spiht yöntemi, erkek yüz resimleri, VQ2b

Çizelge 6.6. BR=0,6 iken 50'şer resimli deney kümelerindeki kodlanmış resimlerin toplam ortalama karesel hataları (Mse: önerilen yöntemle kodlanmış resimlerin ortalama karesel hatası, Mseo: direkt olarak kodlanmış resmin ortalama karesel hatası)

BR=0,6		LOCO-I bayanlar		LOCO-I erkekler		Spiht bayanlar		Spiht erkekler	
		Mse	Mseo	Mse	Mseo	Mse	Mseo	Mse	Mseo
VQ1	A	3,422	3,074	3,471	2,948	6,412	7,318	6,349	6,958
	B	3,596	3,074	3,582	2,948	6,513	7,318	6,425	6,958
VQ2	A	3,549	3,074	3,475	2,948	6,430	7,318	6,351	6,958
	B	3,569	3,074	3,554	2,948	6,501	7,318	6,396	6,958

Çizelge 6.7. BR=1,5 iken 50'şer resimli deney kümelerindeki kodlanmış resimlerin toplam ortalama karesel hataları (Mse: önerilen yöntemle kodlanmış resimlerin ortalama karesel hatası, Mseo: direkt olarak kodlanmış resmin ortalama karesel hatası)

BR=1,5		LOCO-I bayanlar		LOCO-I erkekler		Spiht bayanlar		Spiht erkekler	
		Mse	Mseo	Mse	Mseo	Mse	Mseo	Mse	Mseo
VQ1	A	2,734	0,592	2,901	0,587	3,658	3,117	3,711	2,955
	B	2,946	0,592	3,031	0,587	3,781	3,117	3,795	2,955
VQ2	A	2,768	0,592	2,906	0,587	3,664	3,117	3,735	2,955
	B	2,910	0,592	2,996	0,587	3,756	3,117	3,790	2,955

Yukarıda ki ortalama karesel hata çizelgelerinden (Çizelge 6.6. ve Çizelge 6.7.) anlaşılacağı gibi Öklit uzaklık kullanılarak yüz organlarına LBG VQ yönteminin uygulanması City-Block uzaklık kullanılan yöntemle karşılaştırıldığında tüm yüz resminin kodlanmasında bize daha iyi bir sonuç sağlamaktadır. Bunun yanında VQ1 yöntemi Euclidean uzaklık için VQ2 yöntemine göre daha iyi kodlama yaparken VQ2 yönteminde City-Block uzaklık

kullanılarak yapılan kodlamalar VQ1 yöntemine göre daha iyi bir sıkıştırma sonucu vermektedir.

7. SONUÇLAR

Bu çalışmada, hem sıkıştırma için hem de geri kazanımdaki benzerlik ölçütü için bölgelendirme ele alınmıştır. İnsan yüzünde doğal olarak kullanılan bölgelendirme unsurları, yüz resmindeki en yüksek piksel değişkenliğine sahip yerler olan, aynı zamanda da insan yüzünün doğal organlarına karşılık gelen saç, göz, kaş, burun, ve dudaktır [6]. Bizim çalışmamızda da insan yüzü görüntüleri ele alınmış ve yukarıda bahsedilen bölgelerin ayrı ayrı kodlandığı algoritmalar geliştirilmiştir. Bu yöntem doğal olarak sorgulama ve veri tabanı yaklaşımlarını da gündeme getirmiştir. Sıkıştırma ve sorgulamaya uygunluğun bir arada bulunması, arzu edilen bir durumdur. Bunun yanında bu bölgeler birçok yüz resmi arasında önemli benzerlikler içermektedir. Bu nedenle bu bölgeler vektör niceme (VQ) gibi kümeleme yöntemli kodlayıcıların kullanılması için uygundur. VQ ile devam edilen bölütlemenin bir avantajı da VQ kod çizelgesi dizinlerinin yüz resimleri için kolayca bazı ayırma ölçütleri sağlamasıdır. Çünkü bu yüksek piksel değişimli bölgeler aslında insanların yüzleri tanımak için dikkate aldığı öznelik bilgisini içeren önemli bölgelerdir [8]. Bu bölütleri kümeleyerek ve kod numaraları atayarak, VQ kodlamayı ve geri kazanma parametrelerini aynı anda elde edilmiştir.

Bölgelere ayırma sonucu elde edilen resimlerin küçük olması, hatta VQ ile kodlanarak sadece birer rakama dönüşmesi, veri tabanında sorgulama yapmada bize büyük bir kolaylık sağlamıştır.

Kullanılan yöntem temelde üç aşamadan oluşmaktadır:

1. Resimden yüze ait kısmın çıkartılıp, göz, kaş, burun ve ağız bölgelerinin ayrılması.
2. Bölgelere ait küçük resimlerin kodlanması: Bunun için VQ kullanıldı.
3. Çıkarılmış bölgelerin ardından kalan ve daha düz olan yüz bölgesinin kodlanması.

Bu aşamaların ardında her bölge için VQ tarafından üretilen numaraların, ve bölgelerin yüz resmi üzerindeki koordinatlarının bulunduğu öznelik vektörleri (feature vectors) veri tabanına yüklenmiştir. Resmin geri çatılabilmesi için de bölgelerden geriye kalan kısımlara ait resimler ayrı bir yerde depolanmıştır.

Geliřtirdiđimiz yntemin ilk ařamasında benzer boyuttaki yz resimlerinden kullanıcı yardımıyla sabit byklk ve řekilde blge ıkarma iřlemi yapılmıřtır. Elimizdeki yz resimlerine uygun bir řekilde, organları kapsayacak byklkte ve okgen olarak seilen zarf řekilleri kullanılmıřtır. Bu řekillerin i blgelerinde kalan sol gz, sađ gz, sol kař, sađ kař, burun ve ađız kısımları, oluřturulan ayrı veri tabanları ierisinde depolanmıřtır. Yz organ resimlerinin VQ ile en kk sapma seviyesine ulařması iin eniyileme metotları elde edilmiřtir. Bu amala zyineli bir yntem olan LBG [9] algoritması kullanılmıřtır.

Bu alıřma sonucunda; nerilen yntemle dřk sıkıřtırma oranlarında yz resimlerinin direkt olarak sıkıřtırılmasına yakın MSE elde edilirken, yksek sıkıřtırma oranlarında ise grsel olarak daha iyi sonular elde edilmiřtir. Bunun yanında izelge 6.1.6.'dan da grleceđi gibi SPIHT yntemi kullanılarak yapılan yksek sıkıřtırmada nerilen yntemle grsel olarak daha iyi sonular elde edilmesinin yanında daha dřk ortalama karesel hataya ulařılmaktadır.

Kaynak olarak kullanılan yz resimlerinin farklı ortamlarda oluřturulmuř olması, resimlerde farklı ıřık yođunluklarına sebep olmaktadır. Yz resimlerindeki bu ıřık yođunluk farkının tamamıyla dengelenememesi alıřmanın performansının dřmesine bir etkendir. Ayrıca kullanılan yz resimlerinin yine farklı ortamlarda oluřturulmasından kaynaklanan lek ve rotasyon farklılıđı da performansın dřmesine bir etkendir. Btn bu etkilerin giderilmesi ileriki arařtırmalar iin nemli bir ařama olacaktır.

Grsel bilgi geri kazanımı bilgi teknolojisinde yeni bir konu oluřturmaktadır. Bu alıřmada dřk hızlı bitlerde daha iyi sıkıřtırma elde etmenin yanında veritabanından sorguya iliřkin grntler de geri elde edilebilmektedir. Sadece organ blgelerinin VQ numaraları kullanılarak yapılan bir n tarama ile sorgulama cevabına aday organ resimleri tespit edildikten sonra, bu resimlerin arasından arka fon (yzn geri kalanı) karřılařtırması ile en iyi uyum yapan resim bulunmaktadır. Bylece bu alıřmanın sorgulamaya uygunluđu sađlanmaktadır.

8. KAYNAKLAR

- [1] RABBANI, M. ve JONSE, P.W., *Digital Image Compression Techniques*, Bellingham, WA: SPIE (1991).
- [2] GUTTA, S. ve WECHSLER H., *Face recognition using hybrid classifiers*, *Pattern Rec.*, **30(4)**, (1997).
- [3] WU, J.K. ve. NARASHIMHALU, A.D, *Identifying faces using multiple retrievals*, *IEEE Multimedia*, (yaz 1994).
- [4] BACH, J.R., PAUL, S, ve JAIN, R., *A VIMS for the interactive retrieval of faces*, *IEEE Trans. Knowledge and Data Eng.*, **5(4)**, 619-628, (1993).
- [5] RABIEE, H., KASHYAP, R.L. ve S.R. Safavian, *Multiresolution segmentation based image coding with hierarchical data structures*, *IEEE Int.'l Conf. on ASSP (ICASSP'96)*, Atlanta, GA, (Mayıs 1996).
- [6] CAMPBELL, N.W. MACKEOWN, W.P.J., THOMAS, B.T. ve TROSCIANKO, T. *Interpreting image databases by region classification*, *Pattern Recog.*, **30(4)**, 555-563, (1997).
- [7] YOO, H.W., JUNG, S.H., D-S Jang, ve NA, Y.K., *Extraction of major object features using VQ clustering for content-based image retrieval*, *Pattern Recognition*, **35**, 1115-1126, (2002).
- [8] CHANG, Y., CHOI, I., CHOI, J., KIM, M., ve RAGHAVAN, V.V., *Conceptual retrieval based on feature clustering of documents*, *Proc. ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, Tampere, Finlandiya, (Ağustos 2002).
- [9] LINDE, Y., BUZO, A. ve GRAY, R., *An algorithm for vector quantizer design*, *IEEE Trans. on Communications*, **COM-28**, 84-95, (Ocak 1980).
- [10] ZHAO, Y. ve MALAH, D., *Improved Segmentation and Extrapolation for Block-Based Shape-Adaptive Image Coding*, *Vision Interface (VI'2000)*, Kanada, (Mayıs 2000).
- [11] RAJAN, D. ve CHAUDHURI, S., *Generalized interpolation and its application in super-resolution imaging*, *Image and Vision Computing* **19**, 957-969, (2001).

- [12] HADLOCK, C. H., *Mathematical modeling in the environment*, USA: The Mathematical Association of America, (1998)
- [13] GERSHO, R. ve GRAY, M., *Vector Quantization and Signal compression*, Kluwer Academic Publishers, (1992)
- [14] http-1 : <http://www.data-compression.com/vq.html>
- [15] CHAUDHURI, D., MURTHY, C.A. ve CHAUDHURI, B.B., *A modified metric to compute distance*, Pattern Recognition, 7 No.25, (1992).
- [16] MEILA, M. ve IIECKERMAN, D., *An Experimental Comparison of Several Clustering and Initialization Methods*, Technical Report, Microsoft Research, MSR-TR-98-06, (1998).
- [17] GANTHI, V., RAMAKRISHNAN, R., GEHRKE, J., POWELL, A. ve FRENCH, J., *Clustering large datasets in arbitrary metric spaces*, Proc. Int.'l Conf. on Data Engineering, (1999).
- [18] CHEN, J.Y., BOUMAN, C.A., ve DALTON, J.C., *Hierarchical Browsing and Search of Large Image Databases*, IEEE Trans. on Image Processing, 9(3), 442-455, (Mart 2000).
- [19] SAID, A. ve PEARLMAN, W., *New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*, IEEE Trans. Circuits and Systems for Video Tech., 6(3), 243-250, (Haziran 1996).
- [20] WEINBERGER, M., SEROUSSI, G. ve SAPIRO, G., *LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm*, Proc. Data Compression Conference, Snowbird, Utah, (Nisan 1-3, 1996).
- [21] YANG, M.H., KRIEGMAN, D.J., AHUJA, N., *Detecting Faces in Images: A Survey*, IEEE Trans. Pattern Analysis and Machine Intelligence, 24(1), 34-58 (Ocak 2002)
- [22] HSU, R.L., ABDEL-MOTTALEB, M., ve JAIN, A.K.n, *Face Detection in Color Images*, IEEE Trans. Pattern Analysis and Machine Intelligence, 24(5), 696-706 (Mayıs 2002)
- [23] DEL-BIMBO, A., *Visual Information Retrieval*, Morgan Kaufmann Publishers, (1999)
- [24] DUDA, R.O., HART, P.E., ve STORK, D.G., *Pattern Classification*, John Wiley & Sons Inc. (2001)

- [25] YÜCEL, H., *Vesikalık Resimlerde Yüz Sınırları ve Sağ Göz Pozisyonu Tespiti*, Yüksek Lisans Tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, Türkiye (2002)
- [26] JAIN, R., KASTURI, R. ve SCHUNK, B.G., *Machine Vision*, McGraw-Hill (1995)

9. EKLER

EK-1 Öklit Uzaklığı ve Ortalama

$$D = \frac{1}{M} \sum_{k=1}^M d(x_k, y)$$

$$d(x, y) = d_e(x, y)$$

$$\frac{\partial D}{\partial y} = \frac{1}{M} \sum_{k=1}^M \frac{\partial d_e(x_k, y)}{\partial y} = 0 \Rightarrow \sum_{k=1}^M \frac{\partial d_e(x_k, y)}{\partial y} = 0$$

$$\Rightarrow \frac{\partial d_e(x, y)}{\partial y} = \frac{\partial [\|x-y\|^2]}{\partial y} = \frac{\partial [(x-y)^T(x-y)]}{\partial y} = \frac{\partial [x^T x - x^T y - y^T x + y^T y]}{\partial y}$$

$$\frac{\partial}{\partial w} (p^T f) = \left(\frac{\partial f}{\partial w} \right) p \text{ eşitliği bilindiğinden}$$

$$\frac{\partial [x^T x - x^T y - y^T x + y^T y]}{\partial y} = -2x + 2y \text{ olur.}$$

$$\Rightarrow \sum_{k=1}^M \frac{\partial d_e(x_k, y)}{\partial y} = (-2x_1 + 2y) + (-2x_2 + 2y) + \dots + (-2x_M + 2y) = 0$$

$$\Rightarrow M \cdot y - (x_1 + x_2 + \dots + x_M) = 0$$

$$\Rightarrow y = \frac{1}{M} (x_1 + x_2 + \dots + x_M) = \frac{1}{M} \sum_{k=1}^M x_k = \text{mean}\{x_k\}$$

$$\Rightarrow C = \text{mean}\{x\}$$

EK-2 City-block Uzaklığı ve Ortanca

$$D = \frac{1}{M} \sum_{k=1}^M d(x_k, y)$$

$$d(x, y) = d_{cb}(x, y) = \sum_{i=1}^n |x_i - y_i|$$

$$\frac{\partial D}{\partial y} = 0 \Rightarrow \frac{1}{M} \sum_{k=1}^M \frac{\partial d_{cb}(x_k, y)}{\partial y} = 0$$

$$\Rightarrow \sum_{k=1}^M \frac{\partial \left[\sum_{i=1}^n |x_i - y_i| \right]}{\partial y} = \sum_{k=1}^M \sum_{i=1}^n \frac{\partial |x_{ki} - y_i|}{\partial y_i} = 0$$

$$\frac{\partial |x_{ki} - y_i|}{\partial y_i} = \begin{cases} 1 \leftarrow y_i > x_{ki} \\ 0 \leftarrow y_i = x_{ki} \\ -1 \leftarrow y_i < x_{ki} \end{cases} \text{ olduğundan}$$

$$\sum_{k=1}^M \sum_{i=1}^n \frac{\partial |x_{ki} - y_i|}{\partial y_i} = 0$$

eşitliğin sağlanması için ifadede aynı sayıda (-1) ve (+1) 'li terimin bulunması gerekir.

$$\underbrace{(-1-1-1-\dots-1)}_{N \text{ adet}} + \underbrace{(1+1+1+\dots+1)}_{N \text{ adet}} = 0 \quad \underbrace{(-1-1-1-\dots-1)}_{N \text{ adet}} + \underbrace{(1+1+1-\dots+1)}_{N \text{ adet}} = 0$$

$$y_i = \{x_k \mid P(x_i < x_k) = P(x_i > x_k), \forall j\}$$

Bu da matematiksel anlam olarak Ortanca'ya (Median) karşılık gelir.

Sonuç olarak optimum kod vektörü C:

$$C_n = \underset{k=1}{\overset{M}{\text{median}}}\{x_{ki}\} \quad i = 1, 2, \dots, n$$

EK-3 Hamming Uzaklığı

Hamming uzaklığı'nda data vektörü ve kod vektör bileşenlerinin eşit olması durumunda aradaki uzaklık minimum olmakta dolayısıyla Kod vektör bileşenleri ile data vektörleri bileşenleri ne kadar çok çakışırsa Bozulma o kadar küçük çıkacaktır. Bu durumda Kod vektörünün bileşenleri yani piksel değerleri resim matrisimizin sütunlarında en fazla rastlanan piksel değerlerinden seçildiğinde optimum bozulmaya yaklaşmış olunur. Gri seviye bir resmin bloklara ayrılmış matris hali T ile gösterilsin. Bu durumda optimum kod vektörünün k'nci bileşeni; T matrisinin k'nci sütununda en çok hangi gri seviye

değerine sahip eleman varsa, bu gri seviye değerine eşittir. Şu şekilde formüle edilebilir:

$$y_i(k) = \text{mod}(x_i(k))$$

$$= \{x_i(k) \mid P(x_i(k)) > P(x_j(k)), \forall j\}$$

EK-4 Mahalanobis Uzaklığı

$$d_M(x, y) = (x - y)C_x^{-1}(x - y)^T$$

$$\frac{\partial d_M(x, y)}{\partial y} = C_x^{-1}[-(x - y)^T + (-1)^T(x - y)] = C_x^{-1}[-x^T + y^T - x^T + y^T] = C_x^{-1}[2y^T - 2x^T]$$

$$D = \frac{1}{M} \sum_{k=1}^M d(x_k, y)$$

$$\frac{\partial D}{\partial y} = 0$$

$$\Rightarrow \frac{\partial d(x_1, y)}{\partial y} + \frac{\partial d(x_2, y)}{\partial y} + \dots + \frac{\partial d(x_M, y)}{\partial y} = 0$$

$$\Rightarrow 2C_{x1}^{-1}[y^T - x_1^T] + 2C_{x2}^{-1}[y^T - x_2^T] + \dots + 2C_{xM}^{-1}[y^T - x_M^T] = 0$$

$$y^T [C_{x1}^{-1} + C_{x2}^{-1} + \dots + C_{xM}^{-1}] = C_{x1}^{-1}x_1^T + C_{x2}^{-1}x_2^T + \dots + C_{xM}^{-1}x_M^T$$

$$\Rightarrow y^T = \frac{C_{x1}^{-1}x_1^T + C_{x2}^{-1}x_2^T + \dots + C_{xM}^{-1}x_M^T}{C_{x1}^{-1} + C_{x2}^{-1} + \dots + C_{xM}^{-1}}$$

$$\Rightarrow y_i = \left[\frac{\sum_{j=1}^M C_{xj}^{-1}x_j^T}{\sum_{j=1}^M C_{xj}^{-1}} \right]^T$$

olarak bulunur.

EK-5 SPIHT (Set Partitioning in Hierarchical Trees)

Dalgacık tabanlı sıfır-ağaç bir kodlayıcıdır, aşağıdaki özellikleri taşıyan kodlayıcı programları bulunmaktadır :

PROGRAM ADI		ENTROPİ- KODLAMA METODU	PİRAMİT DÖNÜŞÜMÜ
KODLAYICI	KODÇÖZÜCÜ		
codetree	Dectree	aritmetik	dalgacık (kayıplı)
fastcode	fastdect	yok	dalgacık (kayıplı)
progcode	progdec	aritmetik	S+P (kayıpsız)
codecolr	decdec	aritmetik	KL & dalgacık

En çok sıkıştırma 'codetree/dectree' programları ile elde edilmektedir. 'progcode/progdec' programları 'codetree/dectree' programlarına benzerliğinin yanında kayıplı kodlama olanağı da sunmaktadır. İlk üç kodlayıcı-kodçözücü tek renkli (siyah-beyaz) görüntüler için, dördüncü ise renkli görüntüler için yapılmıştır. Bu çalışmada codetree-dectree ve progcode-progdec programları kullanılmıştır. SPIHT kodlayıcı-kodçözücü programları ortalama karesel hatayı (MSE) en aza indirmeye yönelik olarak tasarlanmıştır.

EK-6 LOCO-I (LOW COMPLEXITY LOSSLESS COMPRESSION FOR IMAGES)

LOCO-I programları öngörülü kodlama prensibi ile çalışmaktadır. LOCO-I aşağıdaki programlardan oluşur:

- locoe - 8 bit kayıpsız kodlayıcı
- locod - 8 bit kayıpsız kodçözücü
- nlocoe - 8 bit kayıplı kodlayıcı
- nlocod - 8 bit kayıplı kodçözücü
- locol6e - 16 bit kayıpsız kodlayıcı

loco16d - 16 bit kayıpsız kodçözücü

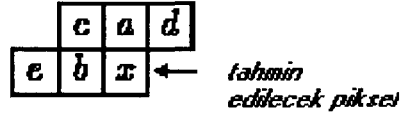
nloco16e - 16 bit kayıplı kodlayıcı

nloco16d - 16 bit kayıplı kodçözücü

"8 bit" renk bileşeni 8 bite kadar olan resimlerde kullanılmaktadır.

"16 bit" renk bileşeni 9'dan 16 bite kadar olan resimlerde kullanılmaktadır.

Algoritma basit bir sabit bağlam modelini baz almaktadır. Model, adaptif bir Golomb-Rice kod ile gerçekleşen Huffman kodları ile verimli bir performans elde edebilmek için ayarlanmıştır. LOCO-I Huffman kodlamanın basitliği ile bağlam modellerinin sıkıştırma potansiyelini birleştirmektedir. Algoritma, temel köşe bulma yeteneğine sahip doğrusal olmayan bir öngörücü kullanmaktadır.



$$\hat{x}_{i+1} \triangleq \begin{cases} \min(a, b) & \text{eğer } c \geq \max(a, b) \text{ ise} \\ \max(a, b) & \text{eğer } c \leq \min(a, b) \text{ ise} \\ a + b - c & \text{diğer durumlarda} \end{cases}$$