

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**AKTİF GÖRÜNÜM MODELİ KULLANARAK
YÜZ TANIMA**

Çağlar TIRKAZ

FBE BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Yrd. Doç. Dr. Songül ALBAYRAK (Yıldız Teknik Üniversitesi)

İSTANBUL, 2008

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AKTİF GÖRÜNÜM MODELİ KULLANARAK
YÜZ TANIMA

Bilg. Müh. Çağlar TIRKAZ

FBE Bilgisayar Mühendisliği Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Songül ALBAYRAK

Prof. Dr. A. Coşkun Şenmez

Prof. Dr. Aytül Ergil

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iii
KISALTMA LİSTESİ.....	iv
ŞEKİL LİSTESİ.....	v
ÇİZELGE LİSTESİ.....	vii
ÖNSÖZ	viii
ÖZET.....	ix
ABSTRACT	x
1 GİRİŞ.....	1
2 ŞEKİL VE DOKU MODELLERİ VE AKTİF GÖRÜNÜM MODELİ	5
2.1 İstatistiksel Şekil Modelleri.....	5
2.1.1 Genel Bakış.....	5
2.1.2 İmajların İşaretlenmesi ve Uygun İşaretlerin Yerleri	5
2.1.3 Şekillerin Hizalanması	6
2.1.4 Şekiller Arasındaki Varyasyonların Ele Alınması.....	8
2.1.5 Hizalama İşleminde Kaynaklanan Lineerlik Kaybının Düzeltilmesi	10
2.2 İstatistiksel Doku Modelleri.....	11
2.2.1 Genel Bakış.....	11
2.2.2 İmaj Çarpıtılması	11
2.2.3 Görüntülerin Normalleştirilmesi.....	15
2.2.4 Doku Varyasyonunun Ele Alınması	16
2.2.5 Doku İşleminde Kullanılan Hızlı PCA	16
2.3 Birleştirilmiş Görünüm Modeli.....	17
2.3.1 Genel Bakış.....	17
2.3.2 Şekil ve Doku Modellerinin Birleştirilmesi.....	17
2.3.3 PCA Sonucunda Kullanılacak Eigen Vektör Sayısının Belirlenmesi.....	19
2.4 Aktif Görünüm Modeli	20
2.4.1 Genel Bakış.....	20
2.4.2 Model Parametrelerinin Nasıl Değiştirileceğinin Çevrimdışı Öğrenilmesi.....	20
2.4.3 İteratif Şekilde Modeli Rötüşlamak.....	27
2.5 Özet	28

3	AKTİF GÖRÜNÜM MODELİ KULLANARAK YÜZ TANIMA	29
4	TEMEL AGM ÜZERİNE YAPILAN EKLEMELER	41
4.1	Genel Bakış	41
4.2	Şekli Daha İyi İfade Etmek	41
4.3	Doku Özelliğini Arttırmak	41
4.4	Aktif Şekil Modeli Benzeri Sınır AGM	43
4.5	Sınırlı AGM Araması	43
4.5.1	Model Eşlemesi	44
4.5.2	Temel AGM Formülasyonu	44
4.5.3	MAP Formülasyonu	45
4.5.4	Noktaların Yerlerinin Önceden Belirtilmesi	46
4.5.5	Deneyler	47
4.6	AGM İlkendirme	50
4.7	Bakış Açısına Dayalı Görünüm Modeli	51
4.7.1	Eğitim Verisi	54
4.7.2	Pozu Tahmin Etmek	55
4.8	Direk Görünüm Modeli	56
4.9	Ters Kompozisyon AGM	57
4.10	Optimum Performans İçin Modelin Yakınsamasını Ayarlamak	60
4.11	Kuvvetli İstatistik Uygulamak	61
5	SONUÇLAR	64
5.1	Tartışma	64
5.1.1	AGM'nin Güçlü Olduğu Noktalar	64
5.1.2	AGM Metodunun Zayıf Kaldığı Noktalar	65
6	KAYNAKLAR	66
7	İNTERNET KAYNAKLARI	70
8	ÖZGEÇMİŞ	71

SİMGE LİSTESİ

k : Euclid boyutu sayısı. (3 boyutlu ise $k=3$)

I : Birim vektör ya da bir imaj

m : Şekil içerisindeki doku örneği sayısı

n : Şekil üzerinde nokta sayısı

N : Eğitim verisindeki şekil sayısı

$\sum$: Kovaryans matrisi

Λ : Eigen değerlerinden oluşan bir diyagonal matris

Φ : Eigen vektörlerinin kolon değerleri olduğu matris

λ_i : i 'inci sıradaki Eigen değeri

ϕ_i : i 'inci sıradaki Eigen vektörü

E : İmaj yakınsaması sırasındaki enerji modeli

$\mathcal{G}$: Radyan cinsinden 2 boyutta şekle verilen rotasyon

KISALTMA LİSTESİ

AGM: Aktif Görünüm Modeli

Birim vektörler: $\mathbf{1} = [1 \dots 1]^T$

Birim matrisler: $I = \begin{bmatrix} 1 & & \dots & & 0 \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 1 \end{bmatrix}$

Matrisler büyük harflerle ifade edilecektir: $M = \begin{bmatrix} x & y \\ z & d \end{bmatrix}$

Nokta çarpımı: $a.b = \sum_n a_n b_n$

Vektörler: $\mathbf{v} = [a, b, c]$

Vektör fonksiyonları: $f(\mathbf{v}) = \mathbf{v} + \mathbf{v}$

ŞEKİL LİSTESİ

Şekil 2-1: İyi işaretler yüksek eğimin ve T kesişimlerinin olduğu yerlerdir	6
Şekil 2-2 Şekil üzerindeki noktalar arasındaki değişim.....	8
Şekil 2-3 İki boyuttaki vektörler kümesine PCA uygulanması. p ana eksenini göstermektedir. Her hangi bir x noktası bu eksen üzerinde bu noktaya en yakın x' noktasıyla ifade edilebilir	10
Şekil 2-4 Farklı en boy oranlarındaki dikdörtgenlerin hizalanması işlemi.....	11
Şekil 2-5 Düzlem üzerindeki 10 noktanın Delaunay üçgenlere ayrılması. (Wikipedia, 2008)	13
Şekil 2-6 İki üçgen sınırından geçen çizgiler çarpıtma işlemi sırasında kırılabilir.....	14
Şekil 2-7 Bilineer interpolasyon sırasında ε için kullanılan değer ε 'nin 4 komşu pikseline bakılarak hesaplanır	15
Şekil 3-1 Eğitim setinde bir kişiye ait görüntüler üzerine 58 referans nokta yerleştirilmiş örnekler	29
3-2 Eğitim setinde bir kişiye ait görüntüler üzerine 112 referans nokta yerleştirilmiş örnekler	30
Şekil 3-3 Bir kişiye ait 5. ve 6. görüntüler.....	31
Şekil 3-4 Bir kişi için yapılan 4 testte kullanılan ilklendirme görüntüleri.....	31
Şekil 3-5 Yakınsama sonucunda bulunan en uygun görüntü.....	31
Şekil 4-1 Delaunay üçgenlere ayırma metodundan sonra oluşan istenmeyen üçgenlerin yok edilmesi	41
Şekil 4-2 Eğer şekil homojen yapıdaysa sadece doku bilgisinden sonuç üretildiği için normalden daha küçük sonuçlar üretilebilir.....	42
Şekil 4-3 (a) 150 adet işaretle elde edilmiş sonuçlar. (b) Yapay noktaların eklenmesiyle 2x150=300 işaret ile elde edilmiş sonuçlar.....	43
Şekil 4-4 Aktif Şekil Modeli benzer AGM'de sadece sınırlardaki doku bilgisi kullanılacak iç noktalar ise bir delik gibi görünecektir	43
Şekil 4-5 İki göz merkezinin varyansının hataya olan etkisi	48
Şekil 4-6 İki göz merkezinin varyansının doku hatasına etkisi	48
Şekil 4-7 Model parametrelerinde bir ön sınırlama kullanıldığında göz merkezlerinin varyansının hata üzerine etkisi.....	49
Şekil 4-8 Model parametrelerinde bir ön sınırlama kullanıldığında göz merkezlerinin varyansının doku hatası üzerine etkisi	49
Şekil 4-9 Sınırlanan nokta sayısı ile şekil hatası arasındaki ilişki	50
Şekil 4-10 Sınırlanan nokta sayısı ile doku hatası arasındaki ilişki.....	50
Şekil 4-11 Modeli eğitmek için kullanılan eğitim setinden örnekler. İlk kolonda profilden, ikinci kolonda yarım profilden, son kolonda ise karşıdan çekilmiş görüntüler yer almaktadır	53
Şekil 4-12 Fotoğraf çekilirken aynı anda profilden fotoğrafın da elde edilmesi için kullanılan ayna	53
Şekil 4-13 Yüz modelinin ilk iki parametresinin yüz modeli üzerindeki etkisi	55
Şekil 4-14 Eğitim seti üzerinde gerçek açının tahmin edilen açıyla karşılaştırılması	56
Şekil 4-15 Üç yüz modelinde bakış açısının değiştirilmesi sonucunda oluşan görüntüler	56

Şekil 4-16 236 görüntüden oluşan bir film için yakınsama sonucunda oluşan RMS hataları. Ters kompozisyon lineer regresyondan çok daha iyi sonuçlar üretmiştir.....	59
Şekil 4-17: Sol tarafta temel AGM ile bir el röntgeninde kemikler bulunmaya çalışılmıştır. Sağ taraftaki görüntüde ise hem benzetimli tavlama hem de Lorentzian normu kullanılmıştır. Sonuç olarak hata 7.0 pikselden 2.4 piksele kadar gerilemiştir.....	63

ÇİZELGE LİSTESİ

Çizelge 3.1 Kullanılan algoritmalar, test resimleri ve modeller ile elde edilen tanıma başarıları	34
Çizelge 3.2 1. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti.....	35
Çizelge 3.3 2. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti.....	36
Çizelge 3.4 3. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti.....	36
Çizelge 3.5 4. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti.....	37
Çizelge 3.6 5. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti.....	37

Çizelge 3.7 6. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özetі.....	38
Çizelge 3.8 Değişik parametrelerle yapılan testlerin sonuçları.....	38
Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.	

ÖNSÖZ

Tezin yazılışı sırasında benden desteklerini eksik etmeyen aileme sevgilerimi sunuyorum. Ayrıca tezin yazımı sırasında görüntülerin işaretlenmesi işleminde bana yardım eden arkadaşım Onur Karagöz'e de teşekkürlerimi iletiyorum.

Bu süreçte tezimle en az benim ilgilendiğim kadar yakından ilgilenen tez Hocam Yrd. Doç. Dr. Songül Albayrak'a teşekkür etmek istiyorum.

ÖZET

Bu tezin yazım amacı istatistiksel şekil ve doku modellerini inceleyerek bunlar hakkında detaylı bilgi vermek, bu modelleri kullanarak kompakt bir model oluşturan Cootes vs. tarafından geliştirilen Aktif Görünüm Modelini (AGM) incelemek ve son olarak da oluşturulan model yardımıyla yüz tanıma performansını değerlendirmektir.

AGM'nin teorik arka planını detaylı olarak açıklanmış ve AGM üzerine yapılabilecek olan uzantılar hakkında bilgi verilmiştir. AGM'nin temel avantajı deneyler sırasında doku ve şekle ait bilginin önceden öğrenilmesidir. Bu bilginin kullanılmasıyla oluşturulan model, başarılı bir şekilde görüntüler içinde bir objeyi tespit edebilmektedir.

AGM kullanılarak yapılan yüz tanıma işleminde değişik poz ve ışık koşulları altında cesaret verici sonuçlara ulaşılmıştır ve bu sonuçlar detaylı olarak açıklanmıştır.

Anahtar Kelimeler: Şekil Değiştirebilen Şablon Modelleri, Ana bileşenler analizi, Şekil ve Doku Analizi, Esnek Obje Bulunması, Aktif Görünüm Modeli, Yüz Tanıma

ABSTRACT

This thesis gives a detailed explanation of the statistical models of shape and texture, the Active Appearance Models (AAM) developed by Cootes et. al, supplies several extensions to basic AAM for better performance and finally uses AAM for face recognition.

The primary advantage of AAM is that it uses a priori knowledge of shape and texture and combines this knowledge to form a compact model. This compact model then can be used for various purposes including image segmentation and object detection.

The details of face recognition and the results obtained using AAM are given. The recognition accuracy is encouraging in images under various lighting and pose variations.

Keywords: Deformable Template Models, Principal Component Analysis, Shape and Texture Analysis, Non-rigid Object Detection, Active Appearance Models, Face Recognition

1 GİRİŞ

Makine görüşünün en büyük amacı görüntüleri anlayabilmektir yani sadece görüntülerin içindeki yapıların bulunması değil bu yapıların neyi ifade ettiğinin bulunabilmesidir. Bu işlem, dünyadaki modellerin kullanılmasını, bu modellerin tanımlanmasını ve işaretlenmesini içerir. Model temelinde yapılan görüş, her ne kadar insanlar tarafından yapılan görünümü sabit cisimlerde başarılı olsa da, karmaşık ve değişken yapıdaki cisimler, mesela yüz veya insanın iç organları ve tıbbi görüntüler, için başarısız olmaktadır. Buradaki temel sorun bu cisimler arasındaki değişkenliktir. Başarılı olabilmesi için bir model kesin olabilmelidir, yani sadece kabul edilebilir modellenen cisim ya da cisimleri temsil edebilmelidir. Hem doğal varyasyona izin verip hem de böyle bir model çıkarabilmek ise zordur. Son gelişmeler, bu zorluğun üstesinden gelmek için modellenecek objenin şekil ve görünüm özelliklerini kullanarak istatistiksel modeller ışığında görüntüleri yorumlayabilmektedir.

Sınıf içi değişimleri öngörerek görüntüleri yorumlamaya çalışan model temelli bu metotlara değişken şablon modelleri (*deformable template models*) denir. Değişken şablon modellerine örnek olarak verebileceğimiz metotlardan bir tanesi Aktif Görünüm Modeli'dir (*Active Appearance Model*). Tezin kalan kısmı bu metodu temel alacak ve açıklamaya çalışacaktır.

Model temelli metotlar, görüntüde bulunması planlanan obje sınıfına eldeki verileri kullanarak, yapısal karmaşıklıkların, gürültünün ve kayıp verinin yol açtığı karmaşıklığı çözmeye yardımcı olabilir ve yapının doğru şekilde işaretlenmesini sağlayabilir. Aktif görünüm modeli (AGM), yapıların tahmini şekillerini, bu şekiller üzerindeki belli başlı noktaları ve noktalar arasındaki ilişkileri ve yapının gri seviye görünümünü kullanarak sadece kabul edilebilir yapılarla kendini sınırlayabilir. Aktif görünüm modelinin bir diğer ilginç tarafı ise yeteri kadar örneği kapsayacak şekilde eğitildiğinde yeni modellerin sentezlenmesinin mümkün olmasıdır. Mesela bir yüz modeli yaratıldığında her hangi bir kişiye ait farklı ifadelerde gerçekçi yüz modelleri yaratmak mümkündür. Böyle bir model kullanıldığında, görüntüyü yorumlamak aslında bir eşleştirme problemidir. Modeli oluşturan parametreler yakınsanmak istenen gerçek objeye göre değiştirilerek objeye en yakın yeni sentezlenmiş yapı oluşturulur.

Aktif görünüm modeli Cootes ve arkadaşları tarafından (T. F. Cootes, 1998) 1998 yılında geliştirilmiştir. Bu metot, aşağıdan yukarıya doğru bir stratejiyi benzeyen metotların aksine

yukarıdan aşağıya bir metottur. Aşağıdan yukarıya metotlar görüntüleri alt seviyede inceler ve yerel olarak bulunması gereken noktaları ya da nokta kümelerini bulmaya çalışır. Global bir model olmadığında bu metotlar hataya açıktır. Aktif görünüm modeli ise önceden yaratılmış olan modeli kullanarak aranan yapıya en benzeyen yapıyı ortaya çıkarmaya çalışır. Tezde bu metodun kullanılmasının sebepleri (T.F. Cootes, 2004):

- Metodun birçok alanda kullanılabilir olması. Aynı algoritma birçok alanda farklı eğitim verisiyle kullanılabilir.
- Modeller eğitim verisinden farklı ama büyük varyasyonlara izin vermeden yeni yapıların oluşturulmasında kullanılabilir.
- İşaretleme işlemini insanlar yaptığı için uzman yardımıyla insanların bilgisinden faydalanılabilir.

Bu tezin amacı ise:

- AGM metodunu açıklamak,
- AGM metodunu kullanan bir sistem geliştirmek,
- AGM metodunun başarısını ölçmektir.

Değişken şablon modelleri kullanarak görüntüleri yorumlayan çalışmalar son yıllarda giderek artmaktadır.

En basit şekilde bir cisim modellemek istersek, cismin tipik bir örneğini referans görüntü olarak kullanıp yeni bir görüntüyle bu referans görüntüsünü bir ilişileşim metoduyla karşılaştırabiliriz. Eğer referans görüntüsündeki noktaların yerleri işaretlenmişse yeni görüntüde bu noktalara karşılık gelen yerlere yakınsayabiliriz. Ancak eğer yeni gelen görüntü ile referans görüntü arasında doku ya da şekil bilgisinde değişiklikler olursa bu metodun başarısı sınırlı olacaktır.

Bu değişikliklerle başa çıkmak için önerilen yöntemlerden biri modeli elle işlemektir (*hand craft*). Mesela Yuille vs. (A. L. Yuille, 1992) yüzde gözleri bulmak için geliştirdikleri modelde elips, ark ve çemberleri kullanmışlardır. Bu sistem efektif olmasına rağmen yeni bir özellik bulunmak istendiğinde bunun için yeni bir çözüm bulunması gerekmektedir.

Staib ve Duncan (Duncan, 1992) ise tıbbi görüntülerdeki şekilleri kapalı eğrilerin fourier açıklayıcılarını kullanarak temsil etmişlerdir. Seçilen katsayılar eğrinin karmaşıklığını belirlemektedir. Eğer katsayılar üzerinde belirli limitler getirilirse oluşacak olan şekli de sınırlamak mümkündür. Fakat böyle bir yol izlenirse her şekil (mesela açık sınırlar) ifade edilemez.

Kass vs. (M. Kass, 1988) Aktif Kontür Modelleri (*Active Contour Models*) ya da yılanlar (*snakes*) adını verdikleri yöntemi oluşturmuşlardır. Bunlar enerjiyi minimize eden eğrilerdir. Enerjiyi eğrilerin pürüzsüzlüğü ve imaj üzerindeki özellik noktalarına olan yakınlık belirler. Bu metot her ne kadar şekli önceden belli olmayan cisimlerin ana hatlarını bulmak için ideal olsa da şekli daha önceden bilinen cisimler için ideal metot değildir.

Poggio vs. (Poggio, 1996) bir cismin örnek bakış açılarından yeni bir açıyla aynı cismi sentezlemeyi başarmışlardır. Daha önce karşılaşılmamış bir açıda gelen yeni görüntüye modeli yerleştirmek için stokastik bir arama metodunu kullanmışlardır. Bu metod yavaş olmasına rağmen sentezlenen görüntüler kalitelidir. Cootes vs. (Taylor, 1994) gri seviye yüzeyin üç boyutlu modelini tanımlamışlar ve şekil ve görünüm olarak sentezleyebilmişlerdir ancak yeni bir görüntüyle model arasında nasıl eşleştirme yapılabileceğine dair bir algoritma vermemişlerdir. Nastar vs. (C. Nastar, 1996) bununla bağlantılı olarak gri seviye yüzeyin üç boyutlu modelini oluşturmuşlar ve hem istatistiksel hem de fiziksel değişiklik elemanlarını eklemişlerdir. Bu çalışmada bir arama algoritması da verilmesine rağmen bu metod çok iyi bir ilklendirmeye gerek duymaktadır.

Covell (Covell, 1996) oluşturulan bir eigen özellik modelinin parametrelerinin şekil model noktalarını doğru yerlere doğru sürebileceğini göstermiştir. AGM bu fikrin bir uzantısı olarak düşünülebilir. Black ve Yacoob (Yacoob, 1995) imajların el ile işaretlenmiş modellerini kullanarak yüz üzerindeki özellikleri tesbit etmişler ama yüzü bütün bir model olarak kullanmamışlardır. AGM bu fikrin genelleştirilmiş hali olarak düşünülebilir. AGM’de imajlar arasındaki, her bir model parametresinde değişikliğe yol açan, değişimler öğrenilir ve model tahmininin değişmesine yol açar. Cootes vs. (T. F. Cootes, 1995) bir eğitim setinden şekil değişikliklerinin öğrenildiği Aktif Şekil Modelleri (*Active Shape Model*) metodunu geliştirmişlerdir. Aktif şekil modelleri metodu sadece şekil bilgisini kullanırken AGM buna ek olarak doku bilgisini de kullanmaktadır.

AGM'nin geliştirilmesine paralel olarak Sclaroff ve Isidoro (Isidoro, 1998) AGM metoduna oldukça benzeyen Aktif Damla (*Active Blob*) metodunu geliştirmişlerdir. Aktif Damla, şekil ve doku bilgisini tek bir prototip imaj kullanarak sonlu eleman modeliyle (*Finite Element Model*) alan, gerçek zamanlı bir takip tekniğidir. Bu teknik AGM'ye takip sistemini imaj farklılıkları üstüne kurmasıyla, imaj hataları arasındaki ilişkiyi öğrenmesiyle ve çevrimdışı olarak parametrelili dengelemesiyle benzemektedir. İki metod arasındaki en önemli fark ise Aktif Damla metodunun tek bir örnek üstünden statik dokuyu değiştirirken AGM'nin bir eğitim seti üzerinden üzerinden işlem yapmasıdır.

2 ŞEKİL VE DOKU MODELLERİ VE AKTİF GÖRÜNÜM MODELİ

2.1 İstatistiksel Şekil Modelleri

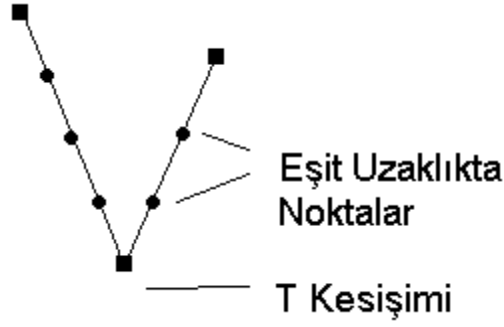
2.1.1 Genel Bakış

Bir objenin şekli her hangi bir boyutta n adet nokta kümesinden oluşur. Genel olarak bu nokta kümesi 2 ya da 3 boyuttadır. 3 boyutlu nokta kümesine örnek olarak 3 boyutlu uzayda noktalar ya da 2 boyutta noktalara zaman boyutunun eklenmesi (mesela bir görüntü sekansı) örnek olarak gösterilebilir. Benzer şekilde 2 boyutlu nokta kümesine örnek olarak 2 boyutlu uzayda noktalar ya da 1 boyutlu uzayda noktalara zaman ekseninin eklenmesi örnek olarak gösterilebilir. Bir nokta kümesini şekil olarak tanımlayabilmemiz için bu nokta kümesindeki elemanlara Euclid dönüşümleri uygulandığımızda özelliklerini kaybetmemeleri gerekir. Bu dönüşümler yer değiştirme, büyütme ya da küçültme ve döndürme operasyonlarıdır. (Mardia, 1998)

AGM'nin amacı hem yeni şekilleri analiz etmek hem de eğitim kümesinde bulunan veriye benzeyen modeller sentezlemektir. Eğitim kümesindeki şekil verisi genel olarak el ile işaretlenmiş imajlar kullanılarak elde edilir.

2.1.2 İmajların İşaretlenmesi ve Uygun İşaretlerin Yerleri

Öncelikle uygun işaretler elimizdeki her şekil üzerinde yer almalıdır. Yani, bir şekil üzerinde işaretlenen nokta ile başka bir şekil üzerinde işaretlenen nokta arasında bağlantı olmalıdır. Uygun işaretlerin yerleri genel olarak şekillerin sınırlarında yüksek eğimin olduğu yerler ve noktaların T şeklinde birleştiği yerlerdir. Ayrıca şekli daha iyi temsil edebilmek için bu noktaların arasına eşit uzaklıkla noktalar seçebiliriz. (Şekil 2.1)



Şekil 2-1: İyi işaretler yüksek eğimin ve T kesişimlerinin olduğu yerlerdir

Eğer işaretleme yapılır ve k boyutta n adet nokta işaretlenirse bu noktalar nk eleman sayısına sahip bir vektörle ifade edilebilirler. Mesela iki boyutta düşünürsek n tane işaretlenen noktayı, $[(x_i, y_i)]$, tek bir örnek üzerinde aşağıdaki gibi, bir x vektörü olarak ifade edebiliriz:

$$x = (x_1, \dots, x_n, y_1, \dots, y_n)^T \quad (2.1)$$

Eğer elimizde s adet eğitim örneği varsa, her eğitim örneği için böyle bir x vektörü yaratabiliriz.

2.1.3 Şekillerin Hizalanması

İstatistiksel olarak bu vektörler üzerinde çalışmadan önce bu vektörlerin hepsinin belirli bir algoritmayla hizalanması gereklidir. Bu hizalama işlemi için genel olarak kullanılan metot Procrustes Analizidir (Goodall, 1991). Bu metot sayesinde bütün görüntülerin ortalama görüntüye olan uzaklıkları toplamı minimize edilir ($D = \sum |x_i - \bar{x}|$). Şekiller arasındaki Procrustes uzaklığını hesaplamak için 4 aşamadan geçilir.

1. İki şeklin ağırlık merkezi hesaplanır.
2. İki şekil eşit büyüklükte olacak şekilde ölçeklenir.
3. İki şekil ağırlık merkezlerine göre hizalanır.
4. İki şekil rotasyonlarına göre hizalanır.

x_1 ve x_2 şekilleri arasındaki Procrustes uzaklığının karesi aşağıdaki gibidir:

$$P_d^2 = \sum_{j=1}^n [(x_{j1} - x_{j2})^2 + (y_{j1} - y_{j2})^2] \quad (2.2)$$

Bir şeklin ağırlık merkezi aşağıdaki gibi hesaplanabilir:

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{j=1}^n x_j, \frac{1}{n} \sum_{j=1}^n y_j \right) \quad (2.3)$$

Her şekli eşit büyüklükte olacak şekilde ölçeklemek için her şeklin büyüklüğü normalize edilir ($|x|=1$). Bunun için öncelikle şeklin büyüklüğü hesaplanır ve her nokta bu büyüklüğe bölünür:

$$S = \sqrt{\sum_{j=1}^n (x_j^2 + y_j^2)} \quad (2.4)$$

İki şeklin rotasyonunu hizalamak için *singular value decomposition* (SVD) (Bookstein, 1997) tekniği uygulanabilir. Bu metodun adımları şöyledir:

1. Boyutu ve merkezi hizalanmış olan iki şekil vektörü x_1 ve x_2 $n \times k$ vektör olarak yazılır (2 boyutta $k=2$ 'dir).
2. $x_1^T x_2$ 'nin SVD'si hesaplanır. ($UDV^T = U(x_1^T x_2)V$)
3. SVD sonucunda x_1 'i x_2 üzerine oturtmak için döndürmemiz gereken açı VU^T 'dur. 2 boyutta düşünürsek: $VU^T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$

Eğitim setindeki bütün şekilleri hizalamak için ise aşağıdaki gibi bir algoritma kullanılabilir:

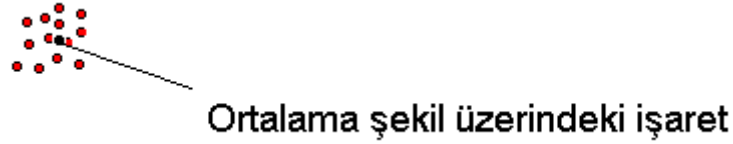
1. Eğitim setindeki ilk görüntü ortalama görüntü olarak kabul edilir.
2. Eğitim setindeki bütün görüntüler ortalama görüntüye hizalanır.
3. Hizalanmış tüm görüntüler kullanılarak tekrar ortalama şekil hesaplanır.
4. Eğer ortalama şekil belirli bir seviyeden fazla değiştiyse tekrar 2. Adıma dönülür.

5. 4. adımda ortalama şekil hesaplanırken (s elimizdeki şekil sayısı ise) aşağıdaki gibi bir formül kullanılabilir:

$$\bar{x} = \frac{1}{s} \sum_{i=1}^s x_i \quad (2.5)$$

2.1.4 Şekiller Arasındaki Varyasyonların Ele Alınması

Şekiller hizalandıktan ve ortalama şekil bulunduktan sonra her bir şekli ortalama şekil üzerine koyup incelersek, ortalama şekil üzerindeki her bir işaret etrafına dağılmış noktalar olduğunu görürüz. Bunun sebebi şekiller arasındaki varyasyondur. (Şekil 2.2)



Şekil 2-2 Şekil üzerindeki noktalar arasındaki değişim

Eğer bu noktaların dağılımını modelleyebilirsek, eğitim setine benzer yeni örnekler yaratabilir ve yeni şekilleri inceleyip bunların doğru şekiller olup olmadıklarına karar verebiliriz.

Problemi basitleştirmek için verinin boyutunu nd 'den daha düşük ve kolay işlem yapılabilir bir boyuta getirmeliyiz. Bunun için kullanabileceğimiz yöntem *Principal Component Analysis*(PCA)'dır (Nielsen, 1994). PCA bir nokta kümesinin içindeki temel aksislerin hesaplanmasına ve her hangi bir noktaya nd 'den daha düşük boyutta bir model kullanarak yakınsamamızı sağlar. PCA metodunun adımları aşağıdaki gibidir:

1. Verinin ortalaması hesaplanır,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^n x_i \quad (2.6)$$

2. Buradan kovaryans matrisinin maksimum olabilirlik (*maximum likelihood*) tahmini aşağıdaki gibidir:

$$\sum_s = \frac{1}{N} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (2.7)$$

3. Kovaryans matrisinin Eigen vektörleri, Φ_i , ve bunlara karşılık gelen Eigen değerleri, λ_i , hesaplanır. (Hesaplama sırasında Eigen değerleri $\lambda_i > \lambda_{i+1}$ olacak şekilde sıralanır)

Eğer Φ en büyük t adet Eigen değerine karşılık gelen Eigen vektörleri barındırıyorsa, eğitim setindeki herhangi bir x vektörüne aşağıdaki gibi yakınsayabiliriz:

$$x \approx \bar{x} + \Phi b$$

$$\Phi = (\Phi_1 | \Phi_2 | \dots | \Phi_t) \text{ ve } b = \Phi^T (x - \bar{x}) \quad (2.8)$$

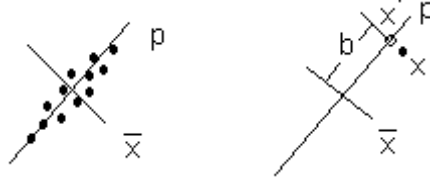
b vektörü biçim değiştirebilen (*deformable*) modelin parametrelerini tanımlar. b vektörünün elemanlarını değiştirerek yukarıdaki formülü kullanarak şekli değiştirebiliriz. Eğer b vektöründeki i . elemanın eğitim verisindeki varyansı σ_i^2 ise $\pm 3\sigma_i$ limitlerinde bu parametrede değişiklik yaparsak eğitim verisindeki şekillere benzeyen yeni şekiller üretebiliriz.

Bundan sonraki aşamada kaç tane Eigen vektör kullanarak şekli temsil edeceğimiz sorusunu cevaplamamız gerekir. Bunun için eğitim setindeki verinin belirli bir oranını (mesela %95) temsil edecek kadar Eigen vektör kullanmamız yeterlidir. Geriye kalan küçük varyasyonu gürültü olarak kabul edebiliriz. Eğer λ_i 'yi eğitim setindeki verinin kovaryans matrisinin Eigen değerleri olarak farz edersek, her bir Eigen değeri eğitim setinin değere karşılık gelen Eigen vektör yönünde ortalama çevresindeki varyansını verir. Eğitim setindeki toplam varyans ise tüm Eigen değerlerinin toplamına eşittir ($V_T = \sum \lambda_i$). Buna göre aşağıdaki formül yardımıyla en büyük t Eigen değerini kullanabiliriz:

$$\sum_{i=1}^t \lambda_i \geq f_v V_T, \text{ burada } f_v \text{ kapsamak istediğimiz oranı temsil etmektedir.}$$

Yaptığımız işlemi grafiksel olarak göstermek istersek, aşağıda 2 boyuttaki noktalara PCA uygulandığında sonucu görebiliriz. Bu durumda herhangi bir nokta ortalamadan geçen ana eksen üzerinden yakınsanabilir. Eğer b , aksis üzerinde yakınsanacak noktaya en yakın noktanın ortalama değere uzaklığı olarak kabul edilirse, $x \approx \bar{x} + bp$. Böylece 2 boyutlu veriye

sadece b parametresini kullanarak tek boyutta yakınsamak mümkün olabilmektedir. (Şekil 2.3)



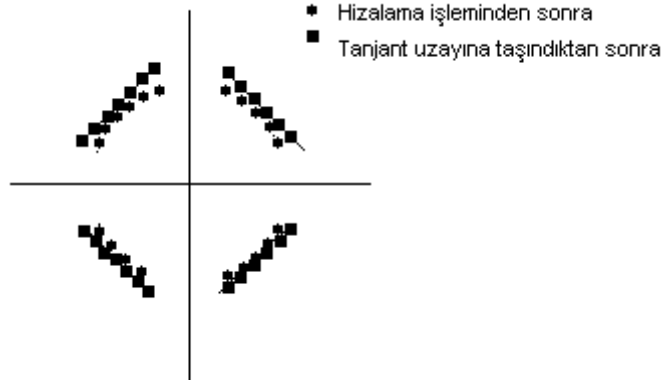
Şekil 2-3 İki boyuttaki vektörler kümesine PCA uygulanması. p ana eksen göstermektedir. Her hangi bir x noktası bu eksen üzerinde bu noktaya en yakın x' noktasıyla ifade edilebilir

Sonuç olarak, PCA metodunu eğitim setini tanımlamak için kullanarak, ortalama şekli hızlı, uygun, güvenilir ve kontrollü bir şekilde değiştirmemiz mümkündür. Böylece şekiller arasındaki varyasyon sıkıştırılmış bir şekil tanımıyla modellenmiş olur. Ayrıca PCA, yeni bir şekli eğitim verisindeki başka bir şekille karşılaştırmak için b parametresini kullanarak kolay bir yol sunmaktadır.

2.1.5 Hizalama İşleminde Kaynaklanan Lineerlik Kaybının Düzeltilmesi

Hizalama işleminde, tüm şekillerin birim büyüklüğe getirilirken $1/S(x)$ ile ölçeklenmesinden kaynaklanan doğrusal olmayan bir varyasyon oluşur. Örnek olarak değişik büyüklüklerdeki (lineer bir değişim) dikdörtgenlerin hizalama işlemini ele alabiliriz. Ölçekleme kısıtı tüm dikdörtgenlerin köşelerinin orijin merkezli bir çember etrafında olmasını sağlar. Yani büyüklükteki lineer bir değişim nokta pozisyonlarında lineer olmayan bir değişime yol açmaktadır. Eğer noktaların bir doğru üzerinde olmasını saplayabilirsek PCA işleminin daha başarılı olmasını sağlayabiliriz.

Bunun için her şekil hizalama işleminden sonra ortalama şeklin *tanjant uzayına* (*tangent space*) dönüştürülür. Bir x_i vektörünün tanjant uzayı x_i 'den geçen ve x_i 'ye dik olan vektörlerin oluşturduğu hiper uzaydır. Bu tanımdan $(x_i - \bar{x}) \cdot x_i = 0$ veya eğer $x_i = 1$ ise $x \cdot x_i = 1$ eşitliklerini sağlayan her x vektörü bu uzayın içerisinde. Aşağıdaki şekilde hem oluşan lineer olmayan yapı hem de tanjant uzayı kullanarak oluşturulan lineerlik gösterilmiştir. Tanjant uzayına şekilleri taşımak için yapılan işlem ise hizalama işleminden sonra her bir şekli $1/(x \cdot \bar{x})$ ile ölçeklemektir. (Şekil 2.4) (T.F. Cootes, 2004)



Şekil 2-4 Farklı en boy oranlarındaki dikdörtgenlerin hizalanması işlemi

2.2 İstatistiksel Doku Modelleri

2.2.1 Genel Bakış

Sadece şekil bilgisiyle tam bir görünüm modeli oluşturulamaz. Sadece piksel bilgisinin yeri değil ayrıca bu piksellerin aldığı değerler de önemlidir ve işleme katılmalıdır.

Şekil bilgisi üzerinde işlem yapılırken verilen girdi el ile işaretlenmiş noktalardı ancak doku bilgisi üzerinde işlem yaparken işaretler arasındaki doku bilgisini alacak bir fonksiyona ihtiyaç duyulmaktadır. Burada kullanılacak olan metot *Delaunay* üçgenlere ayırma (*triangulation*) metodudur. Bu metot doğrudan Eigen vektör parçalaması yapsak oluşacak sahte doku varyasyonunu yok eder. Yani eğitim setinden doku bilgisini almak için öncelikle her görüntü bir referans görüntüye (ortalama görüntü) *çarpıtılır* (*warp*) ve daha sonra örneklenir. Bu aşamadan sonra şekil örneğinde olduğu gibi PCA metodu kullanılarak doku bilgisi temsil edilir.

2.2.2 İmaj Çarpıtılması

İmaj çarpıtılması bir imajın bir başka imaja şekil değiştirmesi olarak düşünülebilir ($I \in \mathbb{R}^k \rightarrow I^* \in \mathbb{R}^k$). Mesela bir imaj üzerinde 2 boyutta yapılan bir kaydırma işlemi çarpıtılma işlemi olarak görülebilir.

AGM işaret temeline dayandığı için kullanacağımız metot herhangi bir nokta kümesini $\{x_1, \dots, x_n\}$ başka bir nokta kümesine $\{x'_1, \dots, x'_n\}$ eşleyebilmelidir. Burada her bir nokta $x = [x, y]^T$ olarak ifade edilebilir.

$$f(x_i) = x'_i \forall i = 1 \dots n \quad (2.9)$$

Böyle bir fonksiyon yardımıyla I imajı üzerindeki her piksel değerini yeni bir I' görüntüsü üzerine eşleyebiliriz. Pratikte, boşluklardan ve ara değer problemlerinden kurtulmak için, ters fonksiyon f' bulmak daha iyidir. Ters fonksiyonu kullanarak hedefteki çarpıtılmış olan I' görüntüsü üzerindeki her bir pikselin I üzerinde nerden geldiği bulunabilir ve doldurulabilir. Genel olarak $f^{-1} \neq f'$, ama yine de iyi bir yakınsamadır.

En kolay imaj çarpıtma fonksiyonu f 'in yerel bir alanda lineer olduğunu ve diğer alanlarda 0 değerini aldığını kabul etmektir.

Mesela tek boyutlu durumda düşünersek (her bir x değeri bir doğru üzerinde), her bir kontrol noktasının artan değerde sıralandığını varsayalım ($x_i < x_{i+1}$). Elimizdeki f fonksiyonunun x_i ile x_{i+1} arasındaki bir x noktasını x'_i ile x'_{i+1} arasındaki bir noktaya eşlemesini sağlamak istiyoruz. f fonksiyonunu aşağıdaki gibi bir toplam olarak parçalayabiliriz,

$$f(x) = \sum_{i=1}^n f_i(x) x'_i, n \text{ adet sürekli } f_i \text{ fonksiyonu aşağıdaki formülü sağlar,}$$

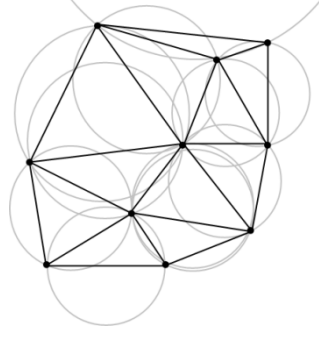
$$f_i(x_j) = \begin{cases} 1 & \rightarrow i = j \\ 0 & \rightarrow i \neq j \end{cases} \quad (2.10)$$

Bu durumda tek boyuttaki f_i fonksiyonunu aşağıdaki gibi tanımlamalıyız:

$$f_i(x) = \begin{cases} (x - x_i) / (x_{i+1} - x_i) & \text{eger } x \in [x_i, x_{i+1}] \text{ ve } i < n \\ (x - x_i) / (x_i - x_{i-1}) & \text{eger } x \in [x_{i-1}, x_i] \text{ ve } i > 1 \\ 0 & \text{diğer durumlarda} \end{cases} \quad (2.11)$$

Bu fonksiyonu kullanarak $[x_1, x_n]$ arasındaki noktalar çarpıtılabilir (T.F. Cootes, 2004).

İki boyutta düşünersek, bir üçgenlere bölme metodu kullanabiliriz. Delaunay üçgenlere ayırma metodu bir nokta kümesini Delaunay özelliğine sahip bir üçgen ağıyla bağlar. Delaunay özelliğine göre eğer üç nokta birbirine bağlanacaksa bu üç noktadan geçen çemberin içinde bir başka nokta olmamalıdır. Aşağıda 10 nokta Delaunay metoduyla üçgenlere ayrılmıştır. (Şekil 2.5)



Şekil 2-5 Düzlem üzerindeki 10 noktanın Delaunay üçgenlere ayrılması. (Wikipedia, 2008)

Noktalar üçgenlerle eşleştikten sonra üçgenlerin içindeki herhangi bir noktayı I' üzerindeki yerine eşleyebiliriz. x_1 , x_2 ve x_3 böyle bir üçgenin köşeleri olsun. Bu üçgenin içindeki herhangi bir nokta,

$$x = x_1 + \beta(x_2 - x_1) + \gamma(x_3 - x_1) = \alpha x_1 + \beta x_2 + \gamma x_3, \quad (2.12)$$

şeklinde yazılabilir. Burada $\alpha = 1 - (\beta + \gamma)$ yani $\alpha + \beta + \gamma = 1$ 'dir. x noktasının üçgenin içinde olması için, $1 \leq \alpha, \beta, \gamma \leq 1$ 'dir. Böylece lineer fonksiyonumuzda x noktası,

$$x' = f(x) = \alpha x'_1 + \beta x'_2 + \gamma x'_3 \quad (2.13)$$

'e eşlenir.

α, β, γ 'nın hesaplanması lineer 2 denklemin çözümünden ibarettir:

$$\alpha = 1 - (\beta + \gamma),$$

$$\beta = \frac{yx_3 - x_1y - x_3y_1 - y_3x + x_1y_3 + xy_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2}, \quad (2.14)$$

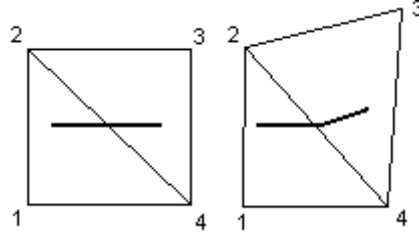
$$\gamma = \frac{xy_2 - xy_1 - x_1y_2 - x_2y + x_2y_1 + x_1y}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2}$$

Çarpıtılmış imaj yaratmak için I' üzerindeki her bir x' pikselini seçip, hangi üçgene ait olduğunu bulduktan sonra α, β, γ parametrelerini hesaplarız. Daha sonra bu parametreleri

kullanarak orijinal imaj üzerindeki bu noktaya karşılık gelen noktayı buluruz. Burada bulduğumuz değeri x' üzerine kopyalarız. Pseudo kod olarak yazarsak:

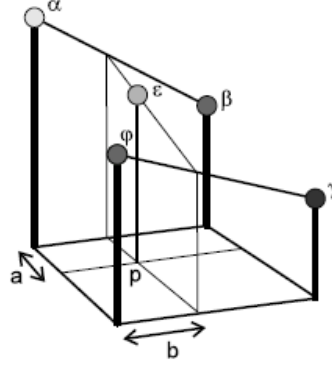
1. $\{x'_1, \dots, x'_n\}$ içindeki her piksel için $x' = [x', y']^T$
 - a. x' 'nin içinde bulunduğu üçgen, t' , bulunur
 - b. t' içindeki α, β, γ hesaplanır (2.14)
 - c. t içindeki pozisyon hesaplanır (2.13)
 - d. Orijinal görüntüdeki piksel değeri kopyalanır. $I'(x') = I(f(x))$

Bu metot her ne kadar süreklilik gösteren bir şekilde tüm noktaları eşlese de pürüzsüz değildir. İki üçgen arasında sınırdan geçen düz çizgiler bu metot kullanıldığında aşağıdaki görüntüde görüldüğü gibi kırılabilir. (Şekil 2.6)



Şekil 2-6 İki üçgen sınırından geçen çizgiler çarpıtma işlemi sırasında kırılabilir

Çarpıtılmış imaj yaratılırken her bir x' pikseli için I üzerinde her zaman bir tam sayı değeri karşılığı çıkmayacaktır. Bunun için bir interpolasyon yapmak gereklidir. Kullanılan interpolasyon tekniği bilineer interpolasyondur, yani bulunan değeri etrafındaki 4 piksele bakarak x' pikselinin değerinin ne olacağına karar verilir. Grafikselsel olarak aşağıdaki ε noktası için formülasyon (Şekil 2.7), $\varepsilon = a(b\gamma + (1-b)\phi) + (1-a)(b\beta + (1-b)\alpha)$, şeklindedir.



Şekil 2-7 Bilineer interpolasyon sırasında ε için kullanılan değer ε 'nin 4 komşu pikseline bakılarak hesaplanır

2.2.3 Görüntülerin Normalleştirilmesi

Global ışık varyasyonunun etkisini minimize etmek için, eğitim setindeki örnekler α ile ölçeklenir ve β ile kaydırılır,

$$g_{norm} = (g_{im} - \beta) / \alpha \quad (2.15)$$

α ve β katsayıları normalleştirilmiş ortalama yüze en iyi şekilde uyacak biçimde seçilir. Doku vektöründeki m adet piksel ortalaması sıfır:

$$\bar{g}_{zeroMean} = \bar{g} - \gamma 1, \quad \gamma = \frac{1}{m} \sum_{i=1}^m g_i = \frac{1}{m} \bar{g} 1 \quad (2.16)$$

ve varyansı birim büyüklükteki ortalama doku vektörüne hizalanır,

$$\bar{g} = \frac{1}{\sigma} \bar{g}_{zeroMean}, \quad \sigma^2 = \frac{1}{m} \sum_{i=1}^m (g_i - \bar{g})^2 \quad (2.17)$$

Burada $\sigma^2 = \frac{1}{m} \sum_{i=1}^m g_i^2$ 'dir çünkü $\bar{g}_{zeroMean}$ 'in ortalama değeri sıfırdır.

Böylece α ve β 'yı aşağıdaki şekilde yazabiliriz:

$$\alpha = g_{im} \cdot \bar{g}, \quad \beta = (g_{im} \cdot 1) / m$$

Pseudo kod olarak doku normalizasyonu aşağıdaki gibidir (Stegmann, 2000):

1. Bütün doku vektörlerinin ortalaması hesaplanır, $\hat{\bar{g}}$

2. $\hat{g}$ normalleştirilir (sıfır ortalama ve birim varyans)
3. Bütün doku vektörleri, g_{im} , için
 - a. $\alpha = g_{im} \cdot \hat{g}$, hesaplanır
 - b. $\beta = (g_{im} \cdot 1) / m$, hesaplanır
 - c. $(g_{im} - \beta) / \alpha$ formülüyle doku vektörü normalleştirilir
4. Eğer ortalama doku değıştiyse 1. Adıma geri dönölür.

2.2.4 Doku Varyasyonunun Ele Alınması

Şekiller arasındaki varyasyonu kullanırken uyguladığımız PCA metodunu da aynen doku varyasyonunu ele alırken de kullanabiliriz.

N adet normalize edilmiş doku vektörünün ortalama değeri:

$$\bar{g} = \frac{1}{N} \sum_{i=1}^N g_i \text{ 'dir.} \quad (2.18)$$

Bu durumda kovaryans matrisini yazarsak:

$$\sum_g = \frac{1}{N} \sum_{i=1}^N (g_i - \bar{g})(g_i - \bar{g})^T \quad (2.19)$$

Böylece m boyutlu doku uzayının ana eksenleri kovaryans matrisinin Eigen vektörler, Φ_g , tarafından temsil edilebilir.

$\sum_g \Phi_g = \Phi_g \lambda_g$, λ_g Eigen vektörlere karşılık gelen Eigen değeri'dir. Böylece yeni bir doku vektörü ortalama dokunun değıştirilmesiyle aşğıdaki gibi oluşturulabilir:

$g = \bar{g} + \Phi_g b_g$, b_g doku modeli parametreleridir.

2.2.5 Doku İşleminde Kullanılan Hızlı PCA

Doku vektörleri üzerinde işlem yaparken her zaman için piksel sayısından daha az sayıda eğitim örneğı olduğunu varsayabiliriz.

Eğer s adet n boyutlu vektöre, x_i , PCA uygulamak istersek ($s < n$), kovaryans matrisinin boyutu $n \times n$ olacaktır ki eğer n büyük bir rakam ise bu işlem uzun sürebilir. Ancak bunun yerine Eigen vektör ve değerlerini veriden çıkarılan $s \times s$ ' lik veri küçük bir matris üzerinden de hesaplayabiliriz. Eigen vektörlere parçalama işlemi kübik karmaşıklıkta bir işlem olduğu için PCA'yı bu metotla yapmak büyük bir getiri sağlayabilir. Bunun için öncelikle her bir veri vektöründen ortalama vektör çıkarılır ve D adındaki bir matrise yerleştirilir:

$$D = ((x_1 - \bar{x}) | \dots | (x_s - \bar{x})) \quad (2.20)$$

Kovaryans matrisi böylece aşağıdaki gibi yazılabilir,

$$S = \frac{1}{s} D D^T \quad (2.21)$$

Eğer $s \times s$ 'lik T matrisini tanımlarsak,

$$T = \frac{1}{s} D^T D \quad (2.22)$$

e_i 'yi T 'nin azalan şekilde sıralanmış Eigen değerlerine, λ_i , karşılık gelen s Eigen vektörü olarak kabul edelim. Buradan bulunan s adet vektörün, De_i , S 'in Eigen vektörlerinin tamamı ve vektörlere karşılık gelen değerlerin de S 'in Eigen değerleri olduğu ve geriye kalan tüm Eigen vektörlerin de değerinin sıfır olduğu gösterilebilir. Bu işlemden sonra ortaya çıkan vektörler, De_i , birim büyüklükte olmadıkları için normalize edilirler ve Eigen vektörlere parçalama işlemi tamamlanır.

2.3 Birleştirilmiş Görünüm Modeli

2.3.1 Genel Bakış

Bu bölümde hem şekil hem de doku modellerini birleştirerek nasıl tek bir görünüm modeli oluşturulacağı anlatılacaktır.

2.3.2 Şekil ve Doku Modellerinin Birleştirilmesi

Önceki bölümlerde açıkladığımız şekilde her bir örneğin şekil ve dokusunu sırasıyla b_s ve b_g parametre vektörleriyle ifade edebiliriz. Şekil ve doku değişimleri arasında oluşabilecek

olası ilgileşimi önlemek ve modeli daha kompakt hale getirmek için üçüncü bir PCA işlemi uygulanmaktadır. Her bir örnek için, W_s 'in her bir şekil parametresinin ağırlığını içeren diyagonal bir matris olduğu varsayılırsa, aşağıdaki şekilde bir parametre vektörü oluşturulabilir:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s \Phi_s^T (x - \bar{x}) \\ \Phi_g^T (g - \bar{g}) \end{pmatrix} \quad (2.23)$$

Burada W_s 'in görevi şekil ve piksel değerleri arasındaki birim farklılığını gidermektir ve bu matris içindeki değerlerin nasıl seçileceği 2.3.2.(1)'de anlatılacaktır.

Bu vektörler üzerinde bir PCA işlemi uygularsak,

$$b = \Phi_c c \quad (2.24)$$

şeklinde bir model oluşturabiliriz. Burada c birleştirilmiş model parametrelerini ifade ederken, Φ_c ise PCA sonucunda ortaya çıkan Eigen vektör kümesini ifade etmektedir. Hem şekil hem doku modeli parametrelerinin ortalaması sıfır olduğu için c 'nin de ortalama değeri sıfırdır.

Bu durumda hem doku hem şekil bilgisini içeren tam bir model aşağıdaki gibi ifade edilebilir:

$$x = \bar{x} + \Phi_s W_s^{-1} \Phi_{c,s} c \quad (2.25)$$

$$g = \bar{g} + \Phi_g \Phi_{c,g} c \quad (2.26)$$

Burada:

$$\Phi_c = \begin{pmatrix} \Phi_{c,s} \\ \Phi_{c,g} \end{pmatrix} \quad (2.27)$$

Model oluşturulurken oluşan Φ_c vektörünün rankı maksimum elimizdeki eğitim setindeki örnek sayısının değerini alabilir.

Eğer yukarıdaki eşitlikleri daha da özetlemek istersek:

$$\begin{aligned} x &= \bar{x} + Q_s c \\ g &= \bar{g} + Q_g c \end{aligned} \quad (2.28)$$

$$\begin{aligned} Q_s &= \Phi_s W_s^{-1} \Phi_{c,s} \\ Q_g &= \Phi_g \Phi_{c,g} \end{aligned} \quad (2.29)$$

Şeklinde yazabiliriz.

Bu model oluşturulurken şekil ve doku vektörleri birleştirilip tek bir vektör üzerinden PCA işlemini yapmak da mümkündür. Ayırık olarak PCA işlemi yapılmasının sebebini tarihsel olarak görebiliriz. AGM, Cootes vs. tarafından geliştirilen aktif şekil modelinden türediği için ve bu modelde doku bilgisi olmadığı için farklı PCA'lar yapıldığını düşünebiliriz. Ayrıca bu şekilde cisimleri şekil ve doku bilgilerinden ayrı ayrı farklılaştırmak mümkündür.

2.3.2.1 Şekil Parametre Ağırlıklarının Bulunması

b_s içerisindeki elemanlar uzaklık cinsinden, b_g içerisindeki elemanlar ise piksel değerleri cinsindendir bu yüzden iki vektör doğrudan karşılaştırılmaz.

W_s 'i bulmak için kullanılan bir yöntem Cootes vs. (T.F. Cootes, 2004) tarafından önerilmiştir.

Eğer r^2 toplam doku model varyansının toplam şekil varyansına bölümü dersek:

$$W_s = rI \begin{pmatrix} r & . & . & 0 \\ . & . & . & . \\ . & . & . & . \\ 0 & . & . & r \end{pmatrix} \quad (2.30)$$

$$r^2 = \frac{\sum \lambda_{g_i}}{\sum \lambda_{s_i}} \quad (2.31)$$

2.3.3 PCA Sonucunda Kullanılacak Eigen Vektör Sayısının Belirlenmesi

Daha önce yaptığımız gibi varyansın belirli bir yüzdesini sağlayacak şekilde Eigen vektörleri seçebilir ve kalan kısmı gürültü olarak varsayabiliriz. Bu durumda eğer birleştirilmiş modelin %x kadarını kapsamak istersek:

$$\sum_{i=1}^t \lambda_i > \frac{x}{100} \sum_{i=1}^{2n} \lambda_i \quad (2.32)$$

denklemini sağlayacak şekilde t adet Eigen değeri seçmemiz yeterlidir.

2.4 Aktif Görünüm Modeli

2.4.1 Genel Bakış

Bu bölümde hem şekil hem doku bilgisi kullanılarak oluşturulacak olan AGM anlatılacaktır. AGM metoduyla hedef bir görüntüye en yakın sentetik görüntüleri üretmek mümkün olacaktır.

Hedef görüntüye yakınsama, sentezlediğimiz görüntü ile hedef görüntü arasındaki farklılığı minimize etmeye çalıştığımız bir optimizasyon problemidir. Bu problem için bir uzaklık vektörü (δI) tanımlanabilir:

$$\delta I = I_{im} - I_{model} \quad (2.33)$$

Burada I_{im} görüntü üzerindeki gri seviye renklerin olduğu bir vektörü, I_{model} ise model parametreleri (c) kullanılarak sentezlenen görüntünün gri seviye renklerinin olduğu vektördür.

Yapılan optimizasyona göre, bu uzaklık vektörü model parametrelerini değiştirerek minimize edilmelidir. Her ne kadar PCA kullanılarak model sıkıştırılmış olsa da görünüm modeli birçok parametreye sahip olabileceğinden ilk bakışta bu zor bir problem gibi görünmektedir. Ancak modeli yeni bir görüntüyle eşleştirdiğimiz her bir deneme benzer bir optimizasyon problemidir. Bu sınıftaki problemlerin nasıl çözüleceğini öğrenebiliriz. Daha önce öğrendiğimiz bu bilgiyi de, hedef görüntüye yakınsama sırasında model parametrelerinin nasıl değişeceğini bulmak için kullanabilir ve gerçek zamanlı çalışan bir sistem tasarlayabiliriz. Anahtar nokta δI 'nin model parametrelerinin daha iyi bir yakınsama yapmak için nasıl değiştirilmesi gerektiği bilgisini çıkarabilmesidir. (T.F. Cootes, 2004)

2.4.2 Model Parametrelerinin Nasıl Değiştirileceğinin Çevrimdışı Öğrenilmesi

Model parametreleriyle uzaklık vektörü arasında kurabileceğimiz en basit ilişki lineerdir:

$$\delta c = R \delta I \quad (2.34)$$

Cootes vs. çalışmalarında bu ilişkinin AGM için yeterli olduğunu göstermiştir. (T.F. Cootes, 2004) (T. F. Cootes, 1998) (T. F. Cootes, 1998) (T.F. Cootes, 1999)

(2.28)'daki denklemi kullanırsak görünüm modelinin doku ve şekil bilgisini belirleyen model parametresi aşağıdaki gibidir:

$$\begin{aligned} x &= \bar{x} + Q_s c \\ g &= \bar{g} + Q_g c \end{aligned} \quad (2.35)$$

Görüntü üzerindeki bir şekil, noktalara, $x: X = S_t(x)$, belirli bir benzerlik şekil değiştirme uygulanarak oluşturulabilir. Benzerlik şekil değiştirme büyütme, s , kaydırma, (t_x, t_y) , ve çevirme, θ , işlemlerinden oluşur. AGM için pozda küçük değişiklikleri yaratabileceğimiz bir poz vektörü kullanmalıyız. Bu sayede küçük poz değişikliklerini tahmin edebileceğimiz bir lineer regresyon modeli oluşturabiliriz. Lineerliğin korunması için bu parametreler:

$$\begin{aligned} t &= (s_x, s_y, t_x, t_y)^T \\ s_x &= (s \cos \theta - 1) : \text{Büyütme verotasyon} \\ s_y &= s \sin \theta : \text{Büyütme verotasyon} \\ t_x &: x \text{ yönünde kaydırma} \\ t_y &: y \text{ yönünde kaydırma} \end{aligned} \quad (2.36)$$

şeklinde ifade edilir.

Böylece poz parametre vektörü, $t = (s_x, s_y, t_x, t_y)$, birim şekil değiştirme için sıfır değerindedir ve $S_{t+\partial t}(x) \approx S_t(S_{\partial t}(x))$ değerini alır.

Görüntü üzerindeki doku piksel değerleri büyütme ve kaydırma işlemiyle oluşturulur, $g_{im} = T_u(g) + (u_1 + 1)g_{im} + u_2 1$. Burada u şekil değiştirme parametrelerinin tutulduğu bir vektördür, $u=0$ için birim şekil değiştirmesidir ve $T_{u+\partial u}(g) \approx T_u(T_{\partial u}(g))$ değerini alır.

(2.34)'teki R değerini bulabilmek için lineer regresyon modeli kullanılarak bir dizi deney yapılacaktır. Her bir deneyde parametreler küçük bir miktarda değiştirilecek ve model ile imaj arasındaki fark hesaplanacaktır. Burada değiştirilecek olan parametreler poz, t , ve model parametreleridir, c .

Pseudo-kod olarak j 'inci deney şöyle yazılabilir (Stegmann, 2000)

Model parametrelerini (poz ve (veya) model) belli bir miktarda değiştir.

1. c 'yi güncelle: $c = \delta c + c_0$.
2. t 'yi güncelle: $t = \delta t + t_0$.
3. Yeni doku, g_m , ve yeni şekil, x , ile yeni bir model örneği yarat.
4. Şekli, x , verilen poza, t , eşleyerek imaj koordinatlarında şekli, x_{im} , bul.
5. x_{im} 'e karşılık gelen doku vektörünü, g_{im} , sentezle.
6. g_{im} üzerindeki doku örneğini normalize ederek g_i 'yi hesapla.
7. Normalize edilmiş doku farklılığını, $\delta g = g_i - g_m$, hesapla.
8. δt ve δc 'yi deney matrisleri $\mathbf{T}$ ve $\mathbf{C}$ 'nin j 'inci kolonları üzerine yaz.
9. δg 'yi $\mathbf{G}$ matrisinin j 'inci kolonuna yaz.

$\mathbf{T}$, $\mathbf{C}$ ve $\mathbf{G}$ matrislerini ve bunlar arasındaki ilişkileri kullanarak:

$$\mathbf{C} = \mathbf{R}_c \mathbf{G}, \quad \mathbf{T} = \mathbf{R}_t \mathbf{G} \quad (2.37)$$

piksel farklılıkları ile poz/model parametrelerinin değişimleri arasında bir lineer regresyon modeli oluşturulabilir:

$$\delta c = \mathbf{R}_c \delta g \quad (2.38)$$

$$\delta t = \mathbf{R}_t \delta g \quad (2.39)$$

Deneyler yapılırken kullanılması gereken değişim sayısı, lineerlik varsayımı tuttuğu sürece parametre uzayını kaplayacak kadar olmalıdır. Her bir model parametresinde yapılacak değişim miktarı 0.5 standart sapma kadardır. Poz parametrelerinde büyütme işlemi için %10, kaydırma işleminde x ve y ekseninde ± 3 piksel kadardır. (T.F. Cootes, 2004)

2.4.2.1 Lineer Regresyon

Çok değişkenli lineer regresyon, iki küme arasında lineer bir ilişki olduğu durumlarda deneylere dayalı öngörü yapmak için kullanılır. AGM metodunda, $\mathbf{G}$ matrisinde tutulan

normalize edilmiş doku farklılıkları, δg , ile V matrisinde tutulan model değişiklik parametreleri, δc ve δt , arasında lineer bir ilişki olduğunu düşünebiliriz. (Stegmann, 2000)

Bu ilişkinin hesaplanması için s adet deney yapıldığını düşünürsek her bir deneyde t model parametreleri belirli bir miktarda değiştirilirler ve bu değişikliklere karşılık gelen m^* piksel değişikliği hesaplanır. Matris olarak aşağıdaki gibi bir görünüm oluşur:

$$\begin{bmatrix} \begin{bmatrix} v_{1_1} \\ \vdots \\ v_{1_t} \end{bmatrix} & \begin{bmatrix} v_{s_1} \\ \vdots \\ v_{s_t} \end{bmatrix} \end{bmatrix} = R \begin{bmatrix} \begin{bmatrix} g_{1_1} \\ \vdots \\ g_{1_m} \end{bmatrix} & \begin{bmatrix} g_{s_1} \\ \vdots \\ g_{s_m} \end{bmatrix} \end{bmatrix} \quad (2.40)$$

Ya da kısaca:

$$V = RG \quad (2.41)$$

Böylece V txs'lik bir matriskten, R txm'lik bir matris, G ise mxs'lik bir matristir^{**}. Ancak bu işlemi doğrudan çözmek işlemsel olarak çok zor olacağından, G matrisinin boyutu G içerisindeki varyasyonun çoğunu kapsayan k-boyutlu^{***} bir altkümeyle eşlenecektir. Bu işleme *pricipal component regression* denir.

k eğer doku farklılıklarının çarpımının, $G^T G$, en büyük k Eigen değerine, $\{\lambda_1 \dots \lambda_k\}$, karşılık gelen Φ_k ile ifade edilen Eigen vektörleri ise ve $\Lambda_k \text{diag}(\lambda_1 \dots \lambda_k)$ 'ya eşit dersek aşağıdaki gibi bir eşitlik kurabiliriz:

$$G^T G \Phi_k = \Phi_k \Lambda_k \quad (2.42)$$

Buradan aşağıda görülebileceği gibi Eigen vektörlerin transpozu G matrisinin kolonlarının k boyuta izdüşümüdür:

$$B_k = \Phi_k^T = (\Lambda_k^{-1} \Phi_k^T G^T) G \quad (2.43)$$

* Burada m model içerisindeki doku örneği sayısıdır.

** AGM için kullandığımız lineer regresyon işleminde, $m \geq s > t$ olduğunu kabul edebiliriz.

*** $k \geq t$ olmalıdır çünkü $k < t$ olursa R, V uzayını kapsayamaz

B_k 'nın satırları ortogonal olduğu için aşağıdaki şekilde B_k ve transpozunun çarpımı birim matristir:

$$B_k B_k^T = I \quad (2.44)$$

Eğer (2.41)'daki gibi bir denklemi düşünürsek aynı V 'yi kullanarak:

$$V = R_k' B_k \Rightarrow R_k G = R_k' B_k \quad (2.45)$$

Eğer R_k' (2.45) için bir çözümse, (2.43)'ü kullanırsak:

$$R_k = R_k' (\Lambda_k^{-1} \Phi_k^T G^T) \quad (2.46)$$

(2.44)'ü kullanır ve (2.45) denklemini yeniden düzenlersek:

$$R_k' = V B_k^T \quad (2.47)$$

Bundan sonra (2.46)'yı (2.47)'ye uygularsak:

$$R_k = V B_k^T (\Lambda_k^{-1} \Phi_k^T G^T) \quad (2.48)$$

Bu aşamadan sonra tek karar vermemiz gereken k 'nın alacağı değerdir. $\mathbf{R}$, tüm $\mathbf{V}$ uzayını kaplamak zorunda olduğundan ve $G^T G$, $s \times s$ 'lik bir matris olduğundan k üzerinde $t \leq k \leq s$ şeklinde bir limit vardır.

k için en iyi değerini bulmak: k 'yı hesaplamak için tüm olasılıklar denenebilir. Bu şekilde çalışacak bir sistemde $\mathbf{V}$ 'den her seferinde bir kolon çıkarılacak, $\mathbf{G}$ 'den bu kolona karşılık gelen kolon çıkarılacak ve R_k hesaplanacaktır. R_k , v_j ve x_j 'in tahmin edilmesinde oluşacak olan hatanın hesaplanmasında kullanılacaktır. Böyle bir model doğru çalışacak olmasına rağmen çok fazla sayıda işlem gerektirdiği için uygun değildir.

k 'nın uygun deęerinin hesaplanmasında kullanılabilircek verimli bir yöntem ise B_k 'nın kolonlarını sırayla çıkarmaktır. Eęer B_k 'nın j . kolonunu çıkardığımız halini $B_{k/j}$ olarak gösterirsek* (2.44)'ü de kullanarak aşağıdaki eşitliği bulabiliriz:

$$B_{k/j} B_{k/j}^T = B_k B_k^T - b_{kj} b_{kj}^T = I - b_{kj} b_{kj}^T \quad (2.49)$$

Buradan aşağıdaki şekilde bir denklem çıkarılabilir (T.F. Cootes, 2004):

$$(I - b_{kj} b_{kj}^T)^{-1} = (I + f_{kj} b_{kj} b_{kj}^T) \quad (2.50)$$

$$f_{kj} = \begin{cases} \frac{1}{1 - \alpha_{kj}} & |1 - \alpha_{kj}| \geq \varepsilon \\ 0 & |1 - \alpha_{kj}| < \varepsilon \end{cases} \quad (2.51)$$

Yukarıdaki denklemlerde $\alpha_{kj} = |b_{kj}|^2$, dir. ε ise hassasiyete baęlı bir deęerdir.

$B_{k/j}$ 'ye benzer şekilde V matrisinin j . kolonunu çıkardığımız halini $V_{/j}$ olarak gösterebiliriz.

Eęer $R'_{k/j}$ aşağıdaki denklem için bir çözümse:

$$V_{/j} = R'_{k/j} B_{k/j} \quad (2.52)$$

O zaman (2.49), (2.50) ve (2.52)'yi kullanarak:

$$\begin{aligned} R'_{k/j} B_{k/j} &= V_{/j} \\ R'_{k/j} B_{k/j} B_{k/j}^T &= V_{/j} B_{k/j}^T \\ R'_{k/j} &= V_{/j} B_{k/j}^T (B_{k/j} B_{k/j}^T)^{-1} \\ R'_{k/j} &= V_{/j} B_{k/j}^T (I - b_{kj} b_{kj}^T)^{-1} \\ R'_{k/j} &= V_{/j} B_{k/j}^T (I + f_{kj} b_{kj} b_{kj}^T) \end{aligned} \quad (2.53)$$

* $B_{k/j}$ matrisinin büyüklüğü $k \times (s-1)$ 'dir.

Buradan b_{kj} 'yi kullanarak v_j 'yi tahmin ettiğimizde oluşacak hata aşağıdaki şekildedir:

$$\delta v_{kj} = R'_{k/j} b_{kj} - v_j \quad (2.54)$$

Bu denklemde (2.53)'ü kullanırsak:

$$\begin{aligned} \delta v_{kj} &= V_{/j} B_{k/j}^T (I + f_{kj} b_{kj} b_{kj}^T) b_{kj} - v_j \\ &= V_{/j} B_{k/j}^T (b_{kj} + b_{kj} \alpha_{kj}^T) - v_j \\ &= f_{kj} V_{/j} B_{k/j}^T b_{kj} - v_j \end{aligned} \quad (2.55)$$

j. elemanla, $B_{k/j}^T b_{kj} = B_k^T b_{kj}$ olduğu için denklemi daha da düzenlersek:

$$\begin{aligned} \delta v_{kj} &= f_{kj} (VB_k^T b_{kj} - \alpha_{kj} v_j) - v_j \\ &= \frac{1}{1 - \alpha_{kj}} (VB_k^T b_{kj} - v_j) \quad \begin{cases} |1 - \alpha_{kj}| \geq \varepsilon \\ |1 - \alpha_{kj}| < \varepsilon \end{cases} \end{aligned} \quad (2.56)$$

Böylece k adet mod kullandığımızda oluşacak olan hatayı hesaplamak için aşağıdaki denklemi hesaplamalıyız:

$$E_k = \sum_{j=1}^s |\delta v_{kj}|^2 \quad (2.57)$$

Böylece k mod kullanmak için (2.57)'deki toplama işlemini minimize etmek gereklidir. E_k 'yı hesaplamanın bir yolu:

$$\delta V = VB_k^T B_k - V \quad (2.58)$$

Eğer δv_j , δV 'nin j. kolonunu ifade ediyorsa - E_k şöyle bulunabilir:

$$E_k = \sum_{j=1}^s f_{kj}^2 |\delta v_j|^2 \quad \begin{cases} |1 - \alpha_{kj}| \geq \varepsilon \\ |1 - \alpha_{kj}| < \varepsilon \end{cases} \quad (2.59)$$

2.4.3 İteratif Şekilde Modeli Rötüşlamak

Bu bölümde, lineer regresyonun modelin en uygun şekle sokulması sürecinde nasıl kullanılacağıının detayları açıklanacaktır.

Modeli iyileştirmek için yapmamız gereken düzeltmeyi bulacağımız yolu belirledikten sonra, iteratif bir şekilde optimizasyon problemini çözeceğimiz bir algoritma geliştirebiliriz. Bunun için öncelikle model, belli poz ve model parametreleriyle, c_0 , görüntünün üzerinde ilklendirilecek ve modele dayalı normalize edilmiş bir görüntü örneği, g_0 , oluşturulacaktır. Bu görüntü örneğine ve model örneğine bağlı olarak, bir uzaklık vektörü, δg , hesaplanacak ve bu vektör de, model parametrelerinin modelin görüntü üzerine daha iyi oturması için nasıl değişmesi gerektiğinin bulunması için kullanılacaktır. (Stegmann, 2000)

Pseudo-kod olarak yazarsak algoritmanın adımları aşağıdaki gibidir (Stegmann, 2000):

1. Bir k vektörünü $k = [1.5 \quad 0.5 \quad 0.25 \quad 0.125 \quad 0.0625]^T$ olacak şekilde ilklendir.
2. Parametre tahmin sınırları içerisinde modeli ilklendir.
 - a. Model üzerinden normalize edilmiş doku vektörünü oluştur, g_m .
 - b. İmajı, x_{im} , şekil modeli üzerinden sentezle, g_{im} .
 - c. g_{im} 'i normalize et ve g_i 'yi oluştur.
 - d. Hata vektörünü hesapla, $\delta g_0 = g_i - g_m$.
 - e. Hatayı hesapla, $E_0 = |\delta g_0|$.
 - f. Pozdaki yer değiştirmeyi hesapla, $\delta t = R_t \delta g_0$.
 - g. Model parametrelerindeki yer değiştirmeyi hesapla, $\delta c = R_t \delta g_0$.
 - h. $i=1$
 - i. Model parametrelerini güncelle, $c = c - k_i \delta c$.
 - ii. δt şekil değiştirmesine uyacak şekilde şekli değiştir.

- iii. a-e arasındaki adımları kullanarak yeni bir hata hesapla, E_i .
 - i. Eğer $E_i > E_0$ ise $i = i + 1$ olarak eşitle ve h'ye geri dön.
 - j. Oluşan hatayı kaydet $E = E_i$
3. Eğer son iterasyonda E üzerinde herhangi bir gelişme olmadıysa işlemi bitir. Eğer gelişme varsa a'ya geri dön.

Algoritmanın sonlanması için gereken koşul, doğru bir modelin oluşturulması değil lineer regresyonun artık daha fazla performansı geliştirememesidir.

2.5 Özet

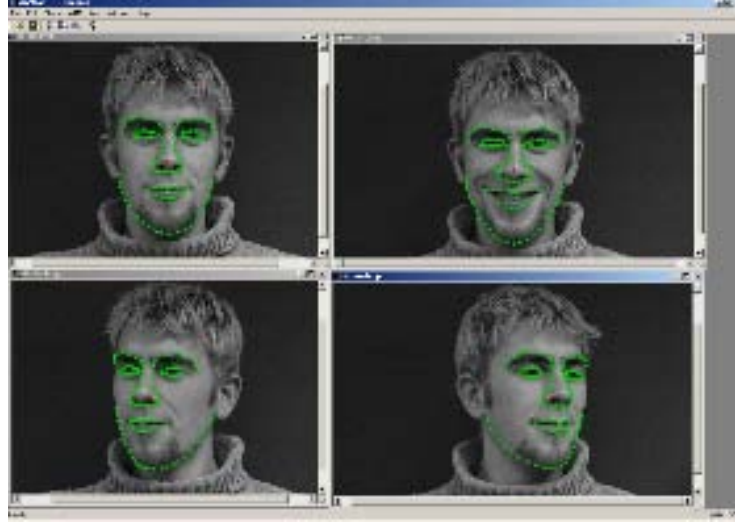
Bu bölümle beraber temel AGM algoritmasını tamamlamış oluyoruz. Şekil ve doku modellerinin PCA metoduyla nasıl çıkarılacağını ve bu iki modelin nasıl tek bir modelde birleştirileceğini. Bundan sonra da program çalışmadan önce nasıl lineer regresyon yardımıyla arama işlemini optimize edeceğimiz görmüş olduk.

3 AKTİF GÖRÜNÜM MODELİ KULLANARAK YÜZ TANIMA

AGM metodunun yeni yüzleri sentezleyebilmek konusundaki başarısını ölçmek için Stegmann'ın (M. B. Stegmann, 2003) geliştirdiği ara yüz, AAM-API, kullanılmıştır. Eğitim seti için AAM-API sitesindeki (Stegmann) veri kümesindeki 40 kişiye ait görüntüler kullanılmıştır. Testler yapılırken üzerinde 112 ve 58 adet işaret bulunan görüntüler kullanılmıştır. Bir kişinin işaret sayısına göre farklı görüntüleri Şekil 3-1 ve Şekil 3-2'de gösterilmektedir.



Şekil 3-1 Eğitim setinde bir kişiye ait görüntüler üzerine 58 referans nokta yerleştirilmiş örnekler



3-2 Eğitim setinde bir kişiye ait görüntüler üzerine 112 referans nokta yerleştirilmiş örnekler

Kişilerin görüntülerini açıklamak gerekirse 1. görüntü kişinin karşıdan çekilmiş herhangi bir yüz ifadesi olmayan görüntüsüdür. 2. görüntü ise kişinin karşıdan çekilmiş ve kişi gülümserken çekilmiş görüntüsüdür. 3. görüntü kişi başını sağa doğru çevirmişken yüz ifadesi olmadan çekilmiş görüntüsüdür. 4. görüntü kişi başını sola çevirmişken herhangi bir yüz ifadesi olmadan çekilmiş görüntüsüdür. 5. görüntü kişinin karşıdan yüz ifadesi olmadan ama farklı ışıklandırma ile çekilmiş görüntüsüdür. 6. görüntü kişinin yüz ifadesi ve başının rotasyonu rastgele olarak çekilmiş görüntüsüdür. Bu görüntüler kullanılarak 6 tane model oluşturulmuştur. Bu modeller şöyle sıralanabilir:

1. Eğitim setinde 1., 2., 3. ve 4. görüntüler ve 58 adet işaret ile oluşturulmuş model
2. Eğitim setinde 1., 2., 3., 4. ve 5. görüntüler ve 58 adet işaret ile oluşturulmuş model
3. Eğitim setinde 1., 2., 3., 4. ve 6. görüntüler ve 58 adet işaret ile oluşturulmuş model
4. Eğitim setinde 1., 2., 3. ve 4. görüntüler ve 112 adet işaret ile oluşturulmuş model
5. Eğitim setinde 1., 2., 3. 4. ve 5. görüntüler ve 112 adet işaret ile oluşturulmuş model
6. Eğitim setinde 1., 2., 3. 4. ve 6. görüntüler ve 112 adet işaret ile oluşturulmuş model

Modeller oluşturulurken, 1. modelde: 55 birleştirilmiş model, 18 şekil modeli ve 98 doku modeli parametresi; 2. modelde: 55 birleştirilmiş model, 21 şekil modeli ve 106 doku modeli parametresi; 3. modelde: 63 birleştirilmiş model, 20 şekil modeli ve 117 doku modeli parametresi; 4. modelde: 56 birleştirilmiş model, 18 şekil modeli ve 98 doku modeli parametresi; 5. modelde: 55 birleştirilmiş model, 21 şekil modeli ve 107 doku modeli parametresi; 6. modelde ise: 45 birleştirilmiş model, 21 şekil modeli ve 94 doku modeli parametresi kullanılmıştır. Parametreler model içindeki varyasyonun %95'ini temsil edecek şekilde seçilmiştir.

Test için 1. ve 4. modelde 5 ve 6. görüntüler, 2. ve 5. modelde 6. görüntü, 3. ve 6. modelde ise 5. görüntü kullanılmıştır. Bir kişiye ait 5. ve 6. görüntüler Şekil 3.3'de gösterilmektedir.



Şekil 3-3 Bir kişiye ait 5. ve 6. görüntüler

Her bir model için her bir test görüntüsü üzerinde 4 deney yapılmıştır. Bu dört deney ilklendirme yapılırken test görüntüsünün ağırlık merkezinin sağına, soluna, üstüne ve altına gelecek şekilde ortalama görüntünün oturtulmasıyla oluşturulmuştur. Şekil 3.4’de bir test kişisi için 1. model kullanılarak yapılan testlerde 5. test görüntüsü için yapılan 4 testin ilklendirme pozisyonları gösterilmektedir.

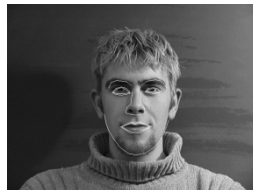


Şekil 3-4 Bir kişi için yapılan 4 testte kullanılan ilklendirme görüntüleri

Yapılan testler sonucunda tüm test görüntüleri için her bir deneyde bu görüntüye eğitim setindeki en yakın görüntü bulunmaya çalışılmıştır. Bu işlem için önce her bir görüntüye en yakın görüntü sentezlenmiştir. Böyle bir sentezleme işlemi sırasında sentezlenen görüntünün nasıl değiştiğini “01-5m_exp00.avi” ye bakarak görebiliriz. Bu süreçte öncelikle ortalama görüntü test görüntüsünün üzerine oturtulmuş. Daha sonra ise yakınsama sırasında parametre değerleri değiştirilerek her bir iterasyonda test görüntüsüne en yakın görüntü sentezlenmeye çalışılmıştır.

01-5m_exp00.avi

Yakınsama işleminden sonra ortaya çıkan en uygun görüntülerden birine örnek olarak Şekil 3.5’e bakabiliriz.



Şekil 3-5 Yakınsama sonucunda bulunan en uygun görüntü.

Yakınsama sonucunda sentezlenen görüntünün birleştirilmiş model parametreleri hesaplanmış ve eğitim setindeki her bir kişiye ait görüntüleri sentezlemek için kullanılan parametrelerle

karşılaştırılmıştır. Bu karşılaştırma yapılırken daha önce hesaplanmış olan bir kişiye ait eğitim setindeki görüntüleri sentezlemek için kullanılan parametreler kullanılmıştır. Bir kişiye ait görüntüleri sentezlemek için kullanılan parametreler aynı sınıfın elemanları olarak görülmüştür. Böylece elimizdeki 40 kişiye ait 40 sınıf oluşturulmuştur. Bu sınıfları birbirinden olabildiğince ayırmak için lineer diskriminant analiz (LDA) (*Linear Discriminant Analysis*) kullanılmıştır. LDA kullanarak temel olarak başarmaya çalıştığımız farklı sınıflara ait noktaları başka bir düzleme birbirlerinden en ayrı olacak şekilde iz düşürmektir. Bunun için her bir sınıftaki her bir nokta $y_i = V^T x_i$ olacak şekilde değiştirilmiştir. Burada V iz düşüm matrisidir. Eğer n_i , i sınıfına ait görüntülerin sayısını, c , toplam sınıf sayısını, μ_i , i sınıfına ait noktaların ortalama değerini ve μ bütün sınıflara ait noktaların ortalama değerini ifade ederse:

$$\mu_i = \frac{1}{n_i} \sum_{x \in \text{class}_i} x \quad (3.1)$$

$$\mu = \frac{1}{n} \sum_{x \in \text{TrainingSet}} x_i \quad (3.2)$$

Buradan sınıf içi Kovaryans matrisi, S_W , aşağıdaki şekilde hesaplanır:

$$S_W = \sum_{i=1}^c \sum_{x_k \in \text{class}_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad (3.3)$$

Sınıflar arası Kovaryans matrisi, S_B , de aşağıdaki gibi hesaplanır:

$$S_B = \sum_{i=1}^c n_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (3.4)$$

Projeksiyon matrisi V ise aşağıdaki problemin çözümleridir:

$$S_B V = \lambda S_W V \quad (3.5)$$

V 'nin bulunması için $S_W^{-1} S_B$ 'nin Eigen vektörleri hesaplanır. V bulunduktan sonra eğitim setindeki yüzleri sentezlemek için kullanılan her bir parametre vektörü V matrisi kullanılarak bu uzaya geçirilir. Bu uzaya yüz tanıma uzayı adını verelim.

Bir test görüntüsü geldiğinde, bu görüntüyü sentezlemek için kullanılan parametrelere c dersek, bu görüntünün hangi kişiye ait olduğunun bulunması için öncelikle yukarıda hesaplanan V vektörü ile bu parametreler yüz tanıma uzayına geçirilir. Yüz tanıma uzayında bir noktanın hangi sınıfa daha yakın olduğunun bulunması için Mahalanobis uzaklığı kullanılmıştır. Metrik olarak Mahalanobis uzaklığı kullanılmasının en büyük sebebi, bu metriğin sınıflar arası farklılık (Kimlik) etkisinin arttırılmasını sağlaması ve sınıf için

farklılığın (Poz, ışıklandırma, yüz ifadesi) etkilerini bastırmasıdır. (G.J. Edwards, 1998) Böylece herhangi bir test örneğinin elimizdeki bir sınıfa olan uzaklığı hesaplanırken:

$$D_i = (V^T * c - \bar{c}_i)^T C^{-1} (V^T * c - \bar{c}_i) \quad (3.6)$$

şeklindeki formül kullanılmıştır. Bu formülde $\bar{c}_i$ bir kişiye ait görüntüleri sentezlemek için kullanılan birleştirilmiş model parametrelerinin yüz tanıma uzayındaki ortalama değerini, C^{-1} ise tüm eğitim kümesindeki görüntüleri sentezlerken hesaplanan birleştirilmiş model parametrelerinin yüz tanıma uzayındaki kovaryans matrisinin tersini temsil etmektedir. Ters kovaryans matrisi hesaplanırken öncelikle tüm eğitim setindeki model parametrelerinin ortalama değeri hesaplanır. Bu ortalama değer yüz tanıma uzayına geçirilir ve yüz tanıma uzayındaki ortalama değer hesaplanır. Daha sonra her bir eğitim görüntüsünü sentezlerken kullanılan birleştirilmiş model parametreleri yüz tanıma uzayına geçirilir ve bu değerlerden bulunan yüz tanıma uzayındaki ortalama çıkarılır. Elde edilen vektör transposu ile çarpılarak bir matris elde edilir. Bu matrisler toplandıktan sonra toplam eğitim setindeki eleman sayısına bölünür ve tersi alınarak ters kovaryans matrisi elde edilmiş olur. Eğer pseudo kod olarak yazmak istersek:

1. Kullanılan birleştirilmiş model parametre vektörü uzunluğunda bir vektör v oluştur ve bu vektörün içindeki tüm elemanları sıfıra eşitle
2. Eğitim setindeki her bir örnek için
 - a. Bu görüntüyü sentezlemek için kullanılan birleştirilmiş model parametrelerini v vektörüne ekle
3. v Vektörünü eğitim setindeki örnek sayısına bölüp ortalama parametre vektörünü oluştur.
4. Ortalama parametre vektörünü yüz tanıma uzayına geçir $v = V^T v$
5. Kovaryans matrisini C oluştur ve bu matrisi sıfıra eşitle. C matrisinin büyüklüğü (v vektörünün uzunluğu x v vektörünün uzunluğu) 'dur.
6. Eğitim setindeki her bir örnek için
 - a. Bu görüntüyü sentezlemek için kullanılan birleştirilmiş model parametrelerini, c , yüz tanıma uzayına geçirdikten sonra ortalama parametre vektörünü v çıkar ve bir d vektörü elde et.
 - b. C matrisine dd^T matrisini ekle.
7. Kovaryans matrisini elimizdeki eğitim setindeki örnek sayısına böl
8. Ters Kovaryans matrisini C matrisinin tersini alarak hesapla $C^{-1} = \text{Invert}(C)$

Yukarıda açıklandığı şekilde her bir test görüntüsü aktif görünüm modeli kullanılarak sentezlenmiş ve sentezlenirken kullanılan parametrelerin yüz tanıma uzayına geçirilmesi ve bu uzayda bulunan noktaya en yakın sınıf merkezinin hesaplanması yardımıyla tanıma gerçekleştirilmiştir. LDA'nın sınıflandırmadaki başarısını diğer sınıflandırma metotlarıyla karşılaştırmak için WEKA (Frank, 2005) içerisindeki Çok Katmanlı Geriye Yaslanma

Algoritması (MPA) (*Multi-layer perceptron analysis*) ve Sıralı Minimal Optimizasyon Algoritması (SMO) (*Sequential Minimal Optimization*) kullanılmıştır.

1. model kullanılarak ve LDA metoduyla yapılan 320 deneyden 239'unda doğru sonuca ulaşılabilmişken 81 görüntüde ise tanıma hatalı olmuştur. Toplam başarı oranı %74.6875 olarak hesaplanmıştır. Bu testlerde 5. görüntülere özel olarak 160 deneyde başarılı olan 114 deneye karşılık 46 deney başarısız olmuş ve başarı oranı %71.25 olarak hesaplanmıştır. 6. görüntülerde ise yapılan 160 deneyin 125'si başarılı olurken 35 deney başarısız olmuş ve bu görüntüler için başarı oranı %78.125 olarak hesaplanmıştır. SMO metodu kullanılarak yapılan 320 deneyden 228'i başarılı olurken 92 deneyde sonuç yanlış olmuştur. Toplam başarı oranı %71.25 olarak hesaplanmıştır. Bu testlerde 5. görüntülere özel olarak yapılan 160 deneyden 113'ü başarılı olabilmışken 47 deneyde sınıflandırma hatalı olmuş ve başarı oranı %70.625 olarak hesaplanmıştır. 6. görüntülere özel olarak yapılan 160 deneyden 115'inde başarılı sonuca ulaşılabilmişken 45 deneyde sınıflandırma hatalı olmuş ve başarı oranı %71.875 olarak hesaplanmıştır. MPA kullanılarak yapılan 320 deneyde ise toplam 226 başarılı sonuca ulaşılabilmişken 94 deneyde sınıflandırma hatalı olmuş ve başarı oranı %70.625 olarak hesaplanmıştır. Bu testlerde 5. görüntülere özel olarak yapılan 160 deneyde 110 deney başarılı olmuşken 50 deneyde sınıflandırma hatalı olmuş ve başarı oranı %68.75 olarak hesaplanmıştır. 6. görüntülere özel olarak yapılan 160 deneyden 116'sında başarılı sonuca ulaşılabilmişken 44 deneyde sınıflandırma hatalı olmuş ve başarı oranı %72.5 olarak hesaplanmıştır.

2. model kullanılarak ve LDA metoduyla yapılan 160 deneyden 122'si başarılı olurken 38 deneyde sınıflandırma hatalı olmuş ve başarı oranı %76.25 olarak hesaplanmıştır. SMO metodu kullanılarak yapılan testlerde 160 deneyden 116'sı başarılı olurken 44 deneyde sınıflandırma hatalı olmuş ve başarı oranı %72.5 olarak hesaplanmıştır. MLP metodu kullanılarak yapılan 160 deneyde ise 113 başarılı sonuca ulaşılabilmişken 47 deney başarısız olmuştur ve başarı oranı %70.625 olarak hesaplanmıştır.

3. model kullanılarak ve LDA metoduyla yapılan 160 deneyden 121'i başarılı olurken 39 deneyde sınıflandırma hatalı olmuş ve başarı oranı %75.625 olarak hesaplanmıştır. SMO metodu kullanılarak yapılan testlerde 160 deneyden 112'si başarılı olurken 48 deneyde sınıflandırma hatalı olmuş ve başarı oranı %70 olarak hesaplanmıştır. MLP metodu kullanılarak yapılan 160 deneyde ise 122 başarılı sonuca ulaşılabilmişken 38 deney başarısız olmuştur ve başarı oranı %76.25 olarak hesaplanmıştır.

4. model kullanılarak ve LDA metoduyla yapılan 320 deneyden 189'unda doğru sonuca ulaşılabilmişken 131 görüntüde ise tanıma hatalı olmuştur. Toplam başarı oranı %59.0625 olarak hesaplanmıştır. Bu testlerde 5. görüntülere özel olarak 160 deneyde başarılı olan 71 deneye karşılık 89 deney başarısız olmuş ve başarı oranı %44.375 olarak hesaplanmıştır. 6. görüntülerde ise yapılan 160 deneyin 118'si başarılı olurken 42 deney başarısız olmuş ve bu görüntüler için başarı oranı %73.75 olarak hesaplanmıştır. SMO metodu kullanılarak yapılan 320 deneyden 200'ü başarılı olurken 120 deneyde sonuç yanlış olmuştur. Toplam başarı oranı

%62.5 olarak hesaplanmıştır. Bu testlerde 5. görüntülere özel olarak yapılan 160 deneyden 76'sı başarılı olabilmışken 84 deneyde sınıflandırma hatalı olmuş ve başarı oranı %47.5 olarak hesaplanmıştır. 6. görüntülere özel olarak yapılan 160 deneyden 124'ünde başarılı sonuca ulaşılabilmişken 36 deneyde sınıflandırma hatalı olmuş ve başarı oranı %77.5 olarak hesaplanmıştır. MPA kullanılarak yapılan 320 deneyde ise toplam 198 başarılı sonuca ulaşılabilmişken 122 deneyde sınıflandırma hatalı olmuş ve başarı oranı %61.875 olarak hesaplanmıştır. Bu testlerde 5. görüntülere özel olarak yapılan 160 deneyde 84 deney başarılı olmuşken 76 deneyde sınıflandırma hatalı olmuş ve başarı oranı %52.5 olarak hesaplanmıştır. 6. görüntülere özel olarak yapılan 160 deneyden 114'ünde başarılı sonuca ulaşılabilmişken 46 deneyde sınıflandırma hatalı olmuş ve başarı oranı %71.25 olarak hesaplanmıştır.

5. model kullanılarak ve LDA metoduyla yapılan 160 deneyden 133'ü başarılı olurken 27 görüntüde sınıflandırma hatalı olmuş ve başarı oranı %83.125 olarak hesaplanmıştır. SMO metodu kullanılarak yapılan testlerde 160 deneyden 135'i başarılı olurken 25 deneyde sınıflandırma hatalı olmuş ve başarı oranı %84.375 olarak hesaplanmıştır. MLP metodu kullanılarak yapılan 160 deneyde ise 138 başarılı sonuca ulaşılabilmişken 22 deney başarısız olmuştur ve başarı oranı %86.25 olarak hesaplanmıştır.

6. model kullanılarak ve LDA metoduyla yapılan 160 deneyden 126'sı başarılı olurken 34 deneyde sınıflandırma hatalı olmuş ve başarı oranı %78.75 olarak hesaplanmıştır. SMO metodu kullanılarak yapılan testlerde 160 deneyden 127'si başarılı olurken 33 deneyde sınıflandırma hatalı olmuş ve başarı oranı %79.375 olarak hesaplanmıştır. MLP metodu kullanılarak yapılan 160 deneyde ise 149 başarılı sonuca ulaşılabilmişken 11 deney başarısız olmuştur ve başarı oranı %93.125 olarak hesaplanmıştır.

Tanıma başarısı, modeller, sınıflandırma algoritması ve test görüntüleri arasındaki ilişki Çizelge 3.1 de açıklanmıştır.

Çizelge 3.1 Kullanılan algoritmalar, test resimleri ve modeller ile elde edilen tanıma başarıları

	5.görüntü LDA	6.görüntü LDA	5.görüntü SMO	6.görüntü SMO	5.görüntü MPA	6.görüntü MPA
1. model	%71.25	%78.125	%70.625	%71.875	%68.75	%72.5
2. model		%76.25		%72.5		%70.625
3. model	%75.625		%70		%76.25	
4. model	%44.375	%73.75	%47.5	%77.5	%52.5	%71.25

5. model		%83.125		%84.375		%86.25
6. model	%78.75		%79.375		%93.125	

Her bir test görüntüsü için sentezlenen görüntüler ile elle işaretlenmiş gerçek görüntüler arasındaki uzaklığın detayları hesaplanmıştır. Buna göre noktalar arası uzaklık, elle işaretlenmiş görüntülerdeki noktalar ile sentezlenen görüntülerde bunlara karşılık bulunan noktalar arasındaki uzaklığı açıklamaktadır. Bu uzaklığın ortalama değeri sentezlenen görüntüdeki her bir nokta ile test görüntüsündeki gerçek noktaların arasındaki farkın hesaplandıktan sonra toplanıp nokta sayısına bölünmesiyle elde edilmektedir. Noktalar arasındaki uzaklığın standart hatası, her bir noktada oluşan hatalardan oluşturulan bir vektörün standart sapmasının elimizdeki nokta sayısının kareköküne bölünmesiyle elde edilmiştir. Noktalar arasındaki uzaklığın medyan değeri, her bir noktada oluşan hatalar sıralandığında ortadaki değer bulunarak hesaplanmıştır. Noktalar arasındaki uzaklığın minimumu, oluşan en küçük hatayı maksimumu ise en büyük hatayı temsil etmektedir. Noktaların eğrilere uzaklığı, sentezlenen görüntüde bulunan şekil ile elle işaretlenmiş görüntülerdeki şekiller arasındaki uzaklığı temsil etmektedir. Bu uzaklığın hesaplanması için sentezlenen görüntüdeki her bir bağlı nokta yolunun gerçek görüntüde bu yola karşılık gelen nokta yoluna uzaklığı hesaplanarak bulunmuştur. Noktaların eğrilere uzaklığındaki ortalama hata, bu uzaklığın her bir test görüntüsünde hesaplandıktan sonra ortalamasının alınmasıyla bulunmuştur. Noktaların eğrilere uzaklığındaki standart hata, hata vektörünün toplam uzunluğunun kareköküne bölünmesiyle elde edilmiştir. Medyan hata, hata vektöründe hatalar sıralandığında ortada yer alan değerdir. Minimum ve maksimum ise noktaların eğrilere uzaklığındaki minimum ve maksimum hatayı temsil etmektedir. Çakışma, sentezlenen görüntüdeki her bir noktanın gerçekte elle işaretlenmiş modelin içinde olup olmamasına göre hesaplanmaktadır. Eğer sentezlenen görüntüdeki en küçük x noktasına $\min X$, en büyük x noktasına $\max X$, en küçük y noktasına $\min Y$, en büyük y noktasına $\max Y$ dersek her bir test görüntüsündeki çakışma aşağıdaki gibi hesaplanabilir:

1. $TP = 0, FP = 0, FN = 0$
2. i $\min X$ 'den $\max X$ 'e kadar
 - a. j $\min Y$ 'den $\max Y$ 'ye kadar
 - i. Eğer (i,j) noktası hem gerçek görüntüdeki şeklin hem de sentezlenen şeklin içindeyse $TP = TP + 1$
 - ii. Eğer (i,j) noktası sentezlenen şeklin içindeyse ama gerçek şeklin içinde değilse $FP = FP + 1$
 - iii. Eğer (i,j) noktası sentezlenen şeklin içinde değilse ama gerçek şeklin içindeyse $FN = FN + 1$
3. $\text{Çakışma} = TP / (TP + FN + FP)$

Çakışmanın ortalama değeri, her bir görüntüdeki çakışmanın ortalama değeri alınarak, medyan değeri çakışmalar bir vektörde sıralandığında ortadaki değer alınmasıyla, standart hata çakışmalar arasındaki hatalar bir vektöre dizildiğinde bu hata vektörünün standart sapmasının vektör uzunluğunun kareköküne bölünmesiyle hesaplanmıştır. Minimum çakışma sentezlenen görüntü ile gerçek görüntü arasındaki minimum çakışmayı maksimum ise maksimum çakışmayı göstermektedir. Mahalanobis uzaklığı, sentezlenen görüntü ile gerçek görüntü arasındaki Mahalanobis uzaklığının hesaplanmasıyla elde edilmiştir. Çizelge 3.2, 3.3, 3.4, 3.5, 3.6 ve 3.7’de her bir model için hesaplanan detaylar gösterilmektedir.

Çizelge 3.2 1. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti

Özet	Ortalama	Standart Hata	Medyan	Minimum	Maksimum
Noktalar arası uzaklık	5.89	0.10	5.65	2.43	14.73
Noktaların eğrilere uzaklığı	3.06	0.06	2.82	1.27	7.67
Çakışma	0.74	0.01	0.75	0.16	0.89
Mahalanobis uzaklığı	60.60	1.61	54.37	17.87	199.53

Çizelge 3.3 2. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti

Özet	Ortalama	Standart Hata	Medyan	Minimum	Maksimum
Noktalar arası uzaklık	6.78	0.26	5.82	3.08	26.80

Noktaların eğrilere uzaklığı	3.50	0.17	2.74	1.33	17.82
Çakışma	0.70	0.01	0.72	0.01	0.89
Mahalanobis uzaklığı	72.01	4.13	55.21	22.47	390.75

Çizelge 3.4 3. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti

Özet	Ortalama	Standart Hata	Medyan	Minimum	Maksimum
Noktalar arası uzaklık	5.83	0.12	5.96	2.36	9.81
Noktaların eğrilere uzaklığı	3.16	0.07	3.16	1.42	5.18
Çakışma	0.72	0.01	0.72	0.39	0.88
Mahalanobis uzaklığı	61.77	1.12	58.95	33.72	101.07

Çizelge 3.5 4. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti

Özet	Ortalama	Standart Hata	Medyan	Minimum	Maksimum
------	----------	---------------	--------	---------	----------

Noktalar arası uzaklık	5.93	0.18	5.37	3.05	36.94
Noktaların eğrilere uzaklığı	3.16	0.11	2.87	1.27	25.03
Çakışma	0.75	0.01	0.76	0.01	0.90
Mahalanobis uzaklığı	66.43	1.70	61.35	20.47	227.05

Çizelge 3.6 5. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti

Özet	Ortalama	Standart Hata	Medyan	Minimum	Maksimum
Noktalar arası uzaklık	6.30	0.38	5.14	2.89	40.56
Noktaların eğrilere uzaklığı	3.20	0.24	2.43	1.16	22.64
Çakışma	0.75	0.01	0.79	0.00	0.91
Mahalanobis uzaklığı	65.71	3.89	51.07	22.80	439.90

Çizelge 3.7 6. model kullanılarak sentezlenen görüntülerin test resimlerine olan uzaklıklarının özeti

Özet	Ortalama	Standart Hata	Medyan	Minimum	Maksimum
Noktalar arası uzaklık	5.20	0.11	4.96	2.92	8.78
Noktaların eğrilere uzaklığı	2.42	0.05	2.21	1.35	4.61
Çakışma	0.78	0.01	0.79	0.51	0.90
Mahalanobis uzaklığı	39.69	0.62	39.06	23.55	68.56

Tanıma işlemi test amaçlı olarak LDA metoduyla, sadece şekil ve doku parametreleri kullanılarak da gerçekleştirilmiş ve birleştirilmiş model parametreleriyle yapılan tanıma işlemiyle karşılaştırılmıştır. Elde edilen verilere göre yüz tanıma sisteminde doku parametrelerinin daha etkili olduğu ortaya çıkmıştır. Ayrıca genel olarak birleştirilmiş model parametrelerinin tanıma için daha başarılı olduğu gözlemlenmiştir. Sadece 4. modelde birleştirilmiş model parametreleriyle yapılan tanıma kötü sonuçlar üretmiştir.

Çizelge 3.8 Değişik parametrelerle yapılan testlerin sonuçları

	1. model	2. model	3. model	4. model	5. model	6. model
Birleştirilmiş	%74.6875	%76.25	%75.625	%59.0625	%83.125	%78.75
Doku	%68.4375	%70	%78.125	%70.3125	%82.5	%72.5
Şekil	%49.6875	%46.25	%38.75	%40	%55.625	%53.75

4 TEMEL AGM ÜZERİNE YAPILAN EKLEMELER

4.1 Genel Bakış

Bu bölümde temel AGM metodu üzerine yapılabilecek eklemelerle AGM metodunun nasıl daha güvenilir ve hızlı çalışabileceği üzerinde durulacaktır. Bu konuda yapılan çalışmalara Cootes vs. (T.F. Cootes, 2004) (Taylor, 1999), Walker vs. (K.N. Walker, 2000), Wolstenholme vs. (Taylor, 1999) örnek gösterilebilir.

4.2 Şekli Daha İyi İfade Etmek

Temel AGM, görüntü çarpıtılma (*warping*) işleminde Delaunay üçgenlere ayırma metodunu kullanır. Bu işlem sonucunda şekil noktalarının oluşturduğu bir üçgen ağı meydana gelir. Ancak eğer üzerinde çalıştığımız şekil konveks değil de konkav ise bu işlem sonucunda gereksiz üçgenler meydana gelip gürültü oluşturabilir. Gürültüyü gidermek için yapılabilecek kolay bir metod, üçgenlere ayırma metodundan sonra her bir üçgen üzerinde dolaşp üçgenin ağırlık merkezinin şekil poligonunun içinde olup olmadığını kontrol etmektir.

Bir noktanın poligonun içinde olup olmadığını kontrol etmek için o noktadan başlayan ve sonsuza giden bir doğrunun poligonu kaç kere kestiğine bakılabilir. Eğer tek sayıda kesiyorsa nokta poligonun içinde çift sayıda kesiyorsa nokta poligonun dışındadır. (Stegmann, 2000) Bu işleme örnek olarak Şekil 4.1'e bakabiliriz:

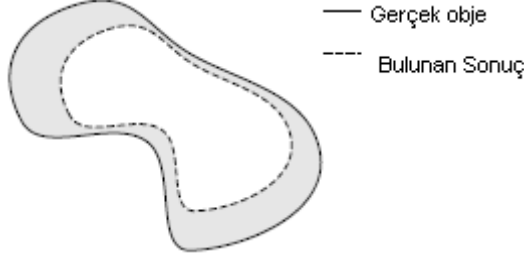


Şekil 4-1 Delaunay üçgenlere ayırma metodundan sonra oluşan istenmeyen üçgenlerin yok edilmesi

4.3 Doku Özelliğini Arttırmak

Eğer yüzey olarak homojen yapıdaki (mesela kemikler) objeleri AGM ye sokarsak AGM sadece doku bilgisine dayalı sonuç üreteceğinden beklediğimizden daha küçük bir yapıyı sonuç olarak üretebilir (Şekil 4.2). Bu yüzden bu tip durumlarda şekle komşu sanal noktalar

yaratılıp bu noktalar AGM'ye verilebilir. Böylece dokuda daha fazla farklılık olacak ve bu durumdan kaçılacaktır.

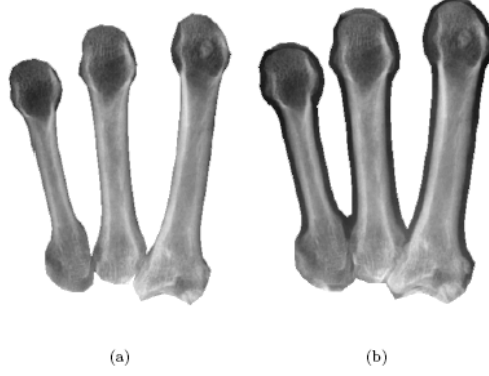


Şekil 4-2 Eğer şekil homojen yapıdaysa sadece doku bilgisinden sonuç üretildiği için normalden daha küçük sonuçlar üretilebilir

Doku PCA'sı bu işlemten sonra farklılaşacaktır ancak şekil PCA'sının eski değerlerini koruması, eski noktalarla ilişkili noktaların eklenmesiyle giderilebilir. Bunun için eski noktalara dik şeklin ortalama büyüklüğüne göre noktaların yaratılması bir sonuçtur. Pseudo-kod olarak yazarsak (Stegmann, 2000):

1. Yapay bir uzaklığı şeklin ortalama büyüklüğüne göre belirle, $d_{mean} = n$.
2. Her bir şeklin ortalama büyüklüğünü hesapla, $s_{mean} = \frac{1}{n} \sum_{i=1}^n S(x_i)$.
3. Her şekil için [1..n]
 - a. Her dış nokta kümesi için
 - i. Her model noktası için
 1. Noktaya dikmeyi hesapla
 2. $d = d_{mean} \frac{S(x_i)}{s_{mean}}$
 3. Dikme üzerinde noktaya d piksel uzakta yeni bir nokta ekle
 - ii. Bitir
 - b. Bitir
4. Bitir

Kemiklerde bu uygulamanın sonuçlarını Şekil 4.3’de görebiliriz. Komşu noktaların eklenmesiyle el tarak kemiklerinde daha başarılı sonuçlar elde edilmiştir.



Şekil 4-3 (a) 150 adet işaretle elde edilmiş sonuçlar. (b) Yapay noktaların eklenmesiyle $2 \times 150 = 300$ işaret ile elde edilmiş sonuçlar

4.4 Aktif Şekil Modeli Benzeri Sınır AGM

Eğer AGM’yi kullandığımız görüntülerin içinde önceden belirlenmemiş bir doku varyasyonu varsa ve bu doku varyasyonu gürültüye neden oluyorsa sadece sınır bilgilerini kullanan bir AGM geliştirmek mümkündür. Bu işlem için bundan önceki bölümde anlatılan gibi bir dış nokta kümesine ek olarak aynı şekilde iç noktalar kümesi eklemek yeterlidir. Böylece sadece sınırdaki doku bilgisi kullanılacak ve şeklin iç kısmı ise AGM’ye girdi olarak girmeyecektir. Şekil 4.4’de durumu daha net olarak görebiliriz:



Şekil 4-4 Aktif Şekil Modeli benzer AGM’de sadece sınırlardaki doku bilgisi kullanılacak iç noktalar ise bir delik gibi görünecektir

4.5 Sınırlı AGM Araması

Birçok uygulamada AGM tek başına yeterli değildir. Eşleştirme işlemi için uygun bir ilklendirmeye ihtiyaç duyulur ve sınırlandırılmadığında AGM her zaman doğru sonuca ulaşamayabilir. Görünüm modeli, şekil ve doku bilgisini barındıran bir modeli sağlar.

İklendirmeye ve sınırlandırmaya ilişkin akla gelen ilk fikir ya manüel olarak ya da bir özellik detektörü yardımıyla bazı şekil noktalarının bulunmasıdır. Mesela bir yüz modeli eşlemesi işleminde gözlerin ortalama yerinin bir kullanıcı ya da bir özellik detektörü yardımıyla hesaplanması kullanışlı olabilir.

Bu bölümde AGM'nin orijinal regresyon analizi yöntemi kullanılan eşleme metodunun nasıl bir istatistiksel yapıya dönüştürülebileceği açıklanacaktır. Bu sayede elde bulunan olasılıklar ve sınırlandırmalar kullanılabilecektir. Sınırlandırmalar sayesinde bir ya da birden fazla noktanın yeri belirli bir varyansla sabitlenebilecektir. Bu sayede detektör ya da kullanıcı yardımıyla bazı noktaların belirlenmesi işlemi sağlıklı bir şekilde yapılabilecektir. (T.F. Cootes, 2004)

4.5.1 Model Eşlemesi

Model eşlemesi sentezlenen model görüntü ile hedef görüntü arasındaki uzaklığı minimize etmeye çalıştığımız bir optimizasyon sürecidir. 2.4.3'de açıkladığımız şekilde, imajdaki, $\mathbf{X}$, model noktalarını tanımlayan görünüm model parametreleri, $\mathbf{c}$, ve şekil değiştirme parametreleri, $\mathbf{t}$, model tarafından oluşturulan imaj örneğinin şeklini verir. Kullanılan parametrelerle oluşturulan imajın başarısını test etmek için, $\mathbf{X}$ tarafından ifade edilen piksellerden bir doku vektörü, $\mathbf{g}_{im}$, elde edilir. Bu vektör doku modeline çevrilir, $\mathbf{g}_s = T^{-1}(\mathbf{g}_{im})$. Modelin dokusu ise $\mathbf{g}_m = \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c}$ ile ifade edilir. Bu örnek için model ve imaj arasındaki uzaklık vektörü ise:

$$\mathbf{r}(p) = \mathbf{g}_s - \mathbf{g}_m \quad (4.1)$$

Bu denklemde $\mathbf{p}$ model parametrelerini barındıran bir vektördür, $\mathbf{p}^T = (\mathbf{c}^T \mid \mathbf{t}^T \mid \mathbf{u}^T)$.

4.5.2 Temel AGM Formülasyonu

Temel AGM formülasyonunu oluşturan fark vektörünün karesini minimize etmeye çalışmaktadır, $E_1(p) = \mathbf{r}^T \mathbf{r}$. İteratif olarak bu fark minimize edilirken mevcut farkın $\mathbf{r}$ olduğunu varsayalım. Bir sonraki adımda $\delta \mathbf{p}$ güncellemesini yaparak $E_1(p + \delta \mathbf{p})$ minimize edilmeye çalışılmaktadır. Eğer (4.1) için birinci derece Taylor açılımını yazarsak, ij 'inci elemanı $\frac{dr_i}{dp_j}$ olan $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ matrisini kullanarak, $\delta \mathbf{p}$ 'nin aşağıdaki denklemi sağlaması gerektiğini gösterebiliriz:

$$r(p + \delta p) = r(p) + \frac{\partial r}{\partial p} \delta p \quad (4.2)$$

$$\frac{\partial r}{\partial p} \delta p = -r \quad (4.3)$$

Böylece:

$$R_1 = \left(\frac{\partial r}{\partial p} \right)^T \frac{\partial r}{\partial p} \quad (4.4)$$

dersek:

$$\delta p = -Rr(p) \quad (4.5)$$

diyebiliriz. (T.F. Cootes, 2004)

Standart bir optimizasyon durumunda $\frac{\partial r}{\partial p}$ her bir adımda hesaplanacaktır. Ancak, normalize edilmiş bir referans çerçevesinde olduğumuz kabul edildiği için bu değer yaklaşık olarak sabittir. Böylece eğitim setinden $\frac{\partial r}{\partial p}$ yaklaşık olarak hesaplanabilir ve ortalaması bulunabilir.

4.5.3 MAP Formülasyonu

Farkın doğrudan karesini minimize etmek yerine, model eşleştirmesini bir olasılık yapısına yerleştirebiliriz. Bir Maksimum aposteriori (MAP) formülasyonunda aşağıdaki olasılığı maksimize etmeye çalışmalıyız:

$$p(model | data) \propto p(data | model)p(model) \quad (4.6)$$

Eğer, farklılık hataları varyansı σ_r^2 olan uniform normal (*gaussian*) olarak kabul edilir ve model parametreleri diyagonal kovaryansı S_p^2 olan bir normal (*gaussian*) olarak kabul edilirse, (4.6) denklemini maksimize etmek aşağıdaki denklemi minimize etmeye eşdeğerdir:

$$E_2(p) = \sigma_r^{-2} r^T r + p^T (S_p^{-1}) p \quad (4.7)$$

Eğer aşağıdaki gibi bir vektör oluşturursak:

$$y = \begin{pmatrix} \sigma_r^{-1} r \\ S_p^{-0.5} p \end{pmatrix} \quad (4.8)$$

O zaman $E_2 = y^T y$ 'dir ve:

$$y(p + \delta p) = y(p) + \begin{pmatrix} \sigma_r^{-1} \frac{\partial r}{\partial p} \\ S_p^{-0.5} \end{pmatrix} \delta p \quad (4.9)$$

Bu durumda uygulanacak en optimum güncelleme değeri aşağıdaki denklemin çözümüdür:

$$\begin{pmatrix} \sigma_r^{-1} \frac{\partial r}{\partial p} \\ S_p^{-1} \end{pmatrix} \delta p = - \begin{pmatrix} \sigma_r^{-1} r(p) \\ S_p^{-0.5} p \end{pmatrix} \quad (4.10)$$

Ya da başka bir şekilde:

$$\delta p = -(R_2 r + K_2 p) \quad (4.11)$$

(4.11) denkleminde R_2 ve K_2 eğitim sırasında önceden hesaplanabilir. Bu durumda güncelleme adımı sadece matris çarpımından ve görüntü sentezinden oluşacaktır. (T.F. Coates, 2004)

4.5.4 Noktaların Yerlerinin Önceden Belirtilmesi

Görüntüde, X_0 , bazı noktalar hakkında önceden bir tahminimiz ve bu noktalar için de bir kovaryans değerimiz, S_X , olduğunu varsayalım. Bilinmeyen noktaların X_0 içindeki değerleri sıfır ve varyansları sonsuz olacaktır. Bunun sonucunda bu noktalar için ters kovaryans, S_X^{-1} , matrisinde değerleri sıfır olacaktır. $d(p) = (X - X_0)$, hedef noktaların şu anki noktalardan uzaklıklarını gösteren bir vektör olsun. Ne kadar iyi yaklaşıldığının bir göstergesi olarak aşağıdaki denklemi kullanabiliriz:

$$E_3(p) = \sigma_r^{-2} r^T r + p^T (S_p^{-1}) p + d^T S_X^{-1} d \quad (4.12)$$

Bu durumda eğer,

$$\begin{aligned}
A_1 &= (\sigma_r^{-2} \frac{\partial r^T}{\partial p} \frac{\partial r}{\partial p} + S_p^{-1} + \frac{\partial d^T}{\partial p} S_x^{-1} \frac{\partial d}{\partial p}) \\
a_1 &= (\sigma_r^{-2} \frac{\partial r^T}{\partial p} r(p) + S_p^{-1} + \frac{\partial d^T}{\partial p} S_x^{-1} d)
\end{aligned} \tag{4.13}$$

dersek, güncelleme adımı aşağıdaki denklemin bir çözümü olmalıdır:

$$A_1 \delta p = -a_1 \tag{4.14}$$

$\frac{\partial d}{\partial p}$ hesaplanırken hem global transformasyon hem de şekil değişiklikleri hesaba katılmalıdır.

Eğer S_t t parametreleriyle global bir transformasyon uyguluyorsa, imaj düzleminde model noktalarının yerleri:

$$X = S_t(\bar{x} + Q_s c) \tag{4.15}$$

şeklinde gösterilebilir.

Böylece,

$$\frac{\partial d}{\partial c} = \frac{\partial S_t(x)}{\partial x} Q_s, \quad \frac{\partial d}{\partial t} = \frac{\partial S_t(x)}{\partial t} \tag{4.16}$$

dersek $\frac{\partial d}{\partial p} = \left(\frac{\partial d}{\partial c} \mid \frac{\partial d}{\partial t} \right)$ diyebiliriz.

Bunlar güncelleme adımlarını bulmak için (4.13) ve (4.14) denklemlerine yerleştirilebilir. (T.F. Cootes, 2004)

4.5.5 Deneyler

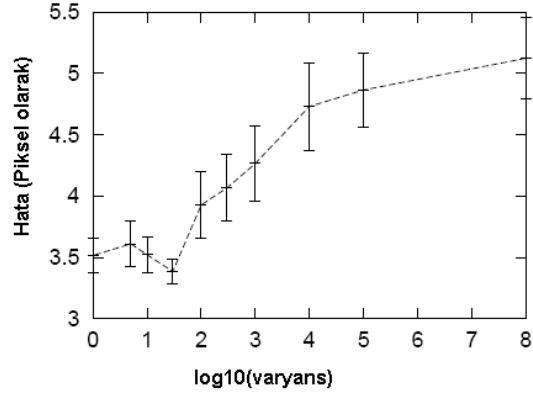
Sınırlı AGM aramasının başarı üzerine etkilerini araştırmak için Cootes vs. 100 görüntülük bir eğitim ve 100 görüntülük bir test kümesi kullanarak M2VTS yüz veritabanında deneyler gerçekleştirmişlerdir. Şekil bilgisi için 68 nokta, doku bilgisi için ise 5000 gri seviye piksel kullanmışlardır.

4.5.5.1 Nokta Sınırlaması

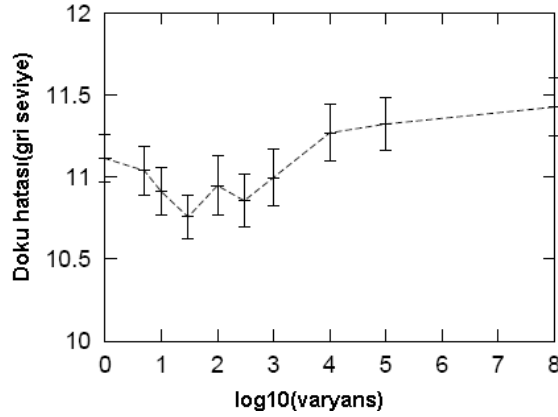
Cootes vs. (T.F. Cootes, 2004) noktaları sınırlamanın etkilerini test etmek için, göz merkezlerinin belirli bir varyansla bilindiğini varsaymıştır. Daha sonra modeli olması

gerekenden 10 piksel kadar x ve y ekseninde oynatmış ve sınırlı bir arama başlatmıştır. 100 adet test görüntüsü üzerinde 9 arama gerçekleştirmiştir ve ortaya çıkan hatayı hesaplamıştır.*

Şekil (4.5)'de değişen varyansın hataya olan etkiyi gösterilmektedir. Şekil (4.6)'da ise aynı değişimin doku hatası üzerine olan etkisi gösterilmektedir.



Şekil 4-5 İki göz merkezinin varyansının hataya olan etkisi



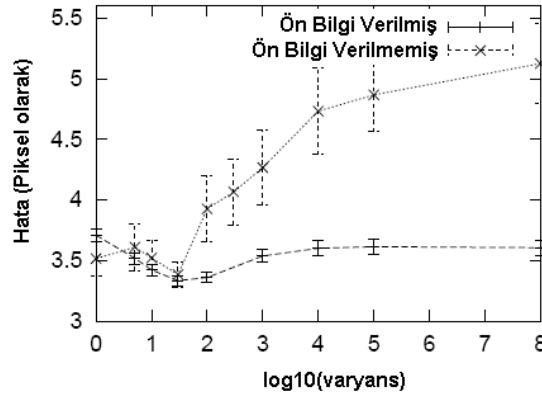
Şekil 4-6 İki göz merkezinin varyansının doku hatasına etkisi

Cootes vs. (T.F. Cootes, 2004) yaptıkları bu deneyler sonucunda standart sapmanın en uygun değerini 7 piksel olarak bulmuşlardır. Göz noktalarını çok katı standartlar ve düşük varyansla sabitlemek ise modelin doğru şekilde eşleşmesine engel olmuştur.

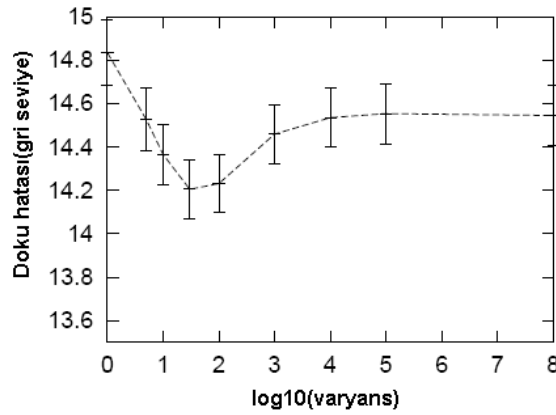
* Hatalar RMS (Root Mean Squared) olarak: Bulunan noktalar ile el ile işaretlenen noktalar arasındaki uzaklığın karelerinin toplamının karekökü olarak belirtilmiştir.

4.5.5.2 Parametre Sınırlaması

Cootes vs. (T.F. Cootes, 2004) aynı deneyi bu sefer model parametreleri üzerine bir normal(*gaussian*) ön bilgi vererek ve (4.12) denklemini kullanılarak tekrarlamıştır. Şekil 4.7 ve 4.8 görüldüğü gibi pozisyon bilgisinde daha iyi sonuç elde edilmesine rağmen doku hatası artmıştır.



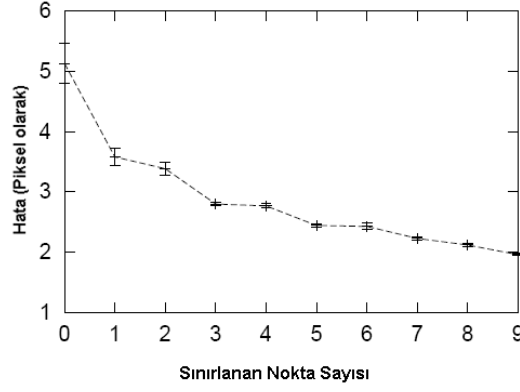
Şekil 4-7 Model parametrelerinde bir ön sınırlama kullanıldığında göz merkezlerinin varyansının hata üzerine etkisi



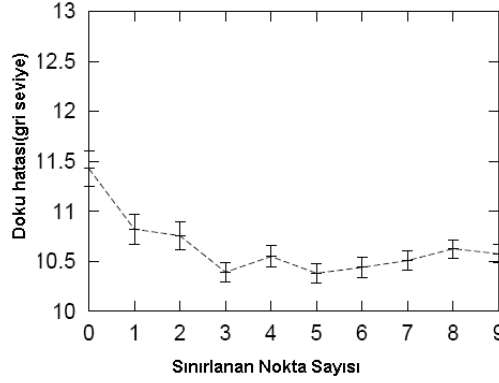
Şekil 4-8 Model parametrelerinde bir ön sınırlama kullanıldığında göz merkezlerinin varyansının doku hatası üzerine etkisi

4.5.5.3 Nokta Sayısını Değiştirmek

Cootes vs. (T.F. Cootes, 2004) aynı deneyi sınırlanan nokta sayısını değiştirerek tekrarlamıştır. Önce göz merkezleri sonra ağız, çene ve yüzün kenarları sınırlanmıştır. Daha fazla nokta eklendikçe noktaların uzaklığı üzerinde daha başarılı sonuçlar alınmıştır. Doku hatasında ise noktalar eklendikçe bir atış gözlemlenmiştir. Bunun sebebi noktaları doğrulayabilmek için dokudan ödün verilmesidir. Eklenen noktalara karşılık doku ve uzaklık hataları arasındaki ilişkiyi Şekil 4.9 ve 4.10'da görebiliriz.



Şekil 4-9 Sınırlanan nokta sayısı ile şekil hatası arasındaki ilişki



Şekil 4-10 Sınırlanan nokta sayısı ile doku hatası arasındaki ilişki

4.6 AGM İlkendirme

AGM iyi bir ilkendirmeye ihtiyaç duyar. AGM'nin temsil ettiği obje sınıfının üyelerine benzer yeni görüntüler sentezleyebilmesi ilkendirme işleminde kullanılabilir. Bu eklemede AGM'nin içsel bir özeliği olan en uygundan belirli bir uzaklıkta AGM'nin en uygun sonuca ulaşabilme özelliği kullanılacaktır.

Öncelikle arama parametreleri belirlenecek ve belirli aralıklarla bu parametrelerin ara noktalarındaki parametreler için sentezleme işlemi yapılacaktır. Böylece elimizde AGM'nin ilk başta sentezleyebildiği görüntülerden oluşan bir aday seti bulunacaktır. Bu set üzerinden AGM başlatılıp birkaç adım (m) çalışmasına izin verilecek ve uzaklık not edilecektir. Daha sonra bu setin içindeki en başarılı (hedef görüntüye en yakın) n adet örnek seçilecek ve ilk başta çalışan AGM optimizasyonundan daha fazla adım (k) AGM çalıştırılacaktır. n adet örnekten en başarılısı ilkendirme işleminde kullanılacaktır. Pseudo-kod olarak yazarsak: (Stegmann, 2000)

1. m 'yi uygun küçük bir rakama eşitle ($m=3$)
2. İçinde n adet aday tutabilecek büyüklükte bir aday seti, $\{c\}$, oluştur.
3. Her bir parametre için uygulamaya özel arama sınırlarını belirle. ($-\sigma_1 \leq c_1 \leq \sigma_1$ vs.)
4. Sınırlar içerisinde her bir parametre uzayını kapsayacak şekilde (lineer regresyonun izin verdiği kadar uzak aralıklarla parametrelerin maksimum ve minimum aralıkları arasına yeni noktalar ekleyerek) arama örnekleri oluştur. $V=\{v_1, \dots, v_r\}$
5. V içindeki her bir vektör için
 - a. AGM optimizasyonu yap (m adım)
 - b. Sonucu hesapla, $E_{fit} = |\delta g^2|$
 - c. Eğer $E_{fit} < \max_{E_{fit}} \{c\}$, vektörü $\{c\}$ 'ye ekle ve $\max_{E_{fit}} \{c\}$ 'yi $\{c\}$ 'den çıkar.
6. $\{c\}$ içindeki her bir eleman için
 - a. AGM optimizasyonu yap (k adım, $k>m$)
 - b. Sonucu hesapla, $E_{fit} = |\delta g^2|$
 - c. E_{fit} 'i güncelle
7. $\{c\}$ içindeki E_{fit} 'i en düşük olan elemanı ilklendirme için kullan.

4.7 Bakış Açısına Dayalı Görünüm Modeli

Bakış açısı değiştikçe iki boyutlu bir objenin görünümü de büyük ölçüde değişir. Günlük hayattaki böyle durumlarla başa çıkabilmek için bu çeşitliliği ifade edebilen modeller geliştirilmesi zorunludur. Örnek olarak vermek gerekirse çoğu yüz tanıma ve takip etme sistemi yüzün önden çekilmiş fotoğrafının ver olduğunu kabul eder ve bakış açısı değiştikçe başarı oranları düşer. Bununla başa çıkabilmek için üç genel metot vardır:

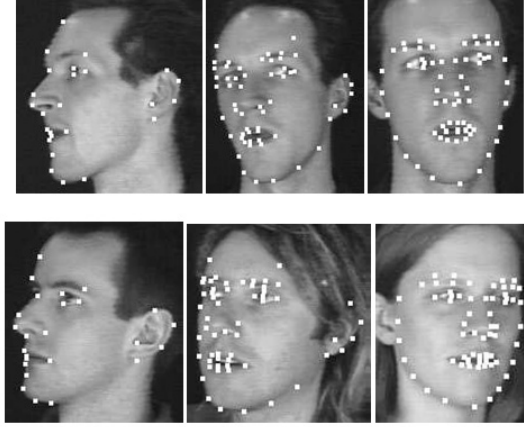
- a) Üç boyutlu bir model kullanmak. (Blanz, 1998) (F. Pighin, 1999) (Yuxiao Hu, 2004) (Sami Romdhani, 2006)

- b) 2D modele lineer olmayan özellikler katmak. (Nayar, 1995) (S. Romdhani, 2000) (S. Romdhani, 1999)
- c) Farklı bakış açılarını temsil eden birden fazla model kullanmak. (Pentland, 1994) (Kruger, 1997)

Bu çalışmada Cootes vs.'nin de (T.F. Cootes, 2004) üzerinde durduğu üçüncü metod anlatılacaktır. Buna göre belirli bir bakış açısını ve değişik bakış açılarının modelleri arasındaki korelasyonu temsil eden bir şekil ve görünüm modeli oluşturulacaktır.

Bakış açısı değiştikçe görüntüler üzerindeki işaretlerden bazıları görünmez hale gelebilir ve böylece modelin bir ön koşulu oluşmayabilir. 180 derecelik rotasyonla başa çıkabilmek için (sağ profilden sol profile kadar) 5 adet modele ihtiyaç vardır. Bunlar -90, -45, 0, 45 ve 90 derecelik (0 önden çekilmiş olarak) bakış açılarıyla oluşturulmuş modeller olacaktır. ± 90 ve ± 45 derecelik modeller birbirlerinin yansımaları olduğu için aslında sadece 3 adet farklı modelimiz olacaktır. Bu modelleri kullanarak kafanın pozunu, büyük değişiklikleri kabul edebilecek şekilde yüz takibini ve bir örneğin farklı açılarda yüzünü sentezleme işlemini başarabiliriz.

Her bir model değişik kişilerin verilen rotasyonlarda kafalarını döndürmesiyle çekilen görüntülerin üzerinde işaretleme yapılmasıyla oluşturulur. Böylece bu model için bir noktanın görünmemesi gibi bir durum söz konusu olamaz. Şekil 4.11 de görüldüğü gibi her bir model için farklı noktalar işaretlenmektedir.



Şekil 4-11 Modeli eğitmek için kullanılan eğitim setinden örnekler. İlk kolonda profilden, ikinci kolonda yarım profilden, son kolonda ise karşıdan çekilmiş görüntüler yer almaktadır

Her bir örnek bakış açısı uygun görünüm modeli ve modele ait parametreler, c , kullanılarak yakınsanabilir. Parametreler ve kafanın açısı arasındaki ilişki öğrenilebilir böylece hem kafanın dönüş açısı tahmin edilebilir hem de her dönüş açısı için yüzün görüntüleri sentezlenebilir. (T.F. Cootes, 2004)

AGM algoritmasını kullanarak her bir modeli yeni bir imaja hızlı bir şekilde eşleyebiliriz. Eğer pozun nasıl olduğu önceden biliniyorsa doğru modeli seçebiliriz. Eğer pozun nasıl olduğu bilinmiyorsa ise beş modelin hepsiyle arama yapıp en başarılı sonucu veren model kullanılabiliriz. Bir model seçildikten ve eşlendikten sonra kafanın dönüş açısı tahmin edilebilir, böylece yüz takip edilebilir ve yüz açısı değişirse yeni bir modele geçilebilir. Bir açısına ait modelle diğer bakış açısına ait modeller arasında açık bir ilgileşim vardır. Bu ilgileşimi öğrenmek için farklı bakış açılarından çekilmiş fotoğraflara ihtiyacımız vardır. Bunun için Cootes vs. (T.F. Cootes, 2004) deneylerinde önden fotoğraf çekildiğinde fotoğrafı çekilen insanın profilini gösterecek şekilde bir ayna yerleştirmiştir.



Şekil 4-12 Fotoğraf çekilirken aynı anda profilden fotoğrafın da elde edilmesi için kullanılan ayna

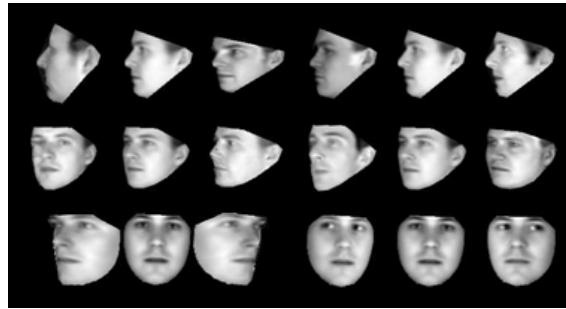
Bu görüntüleri işaretleyerek ve önden ve profilden çekilen görüntüleri eşleyerek, bunlara karşılık gelen bir parametre seti elde edilmiştir. Bu parametreler daha sonra analiz edilmiş ve hem profil hem de önden çekilmiş görüntülerin görünümelerini kontrol eden bir ikili model oluşturulmuştur. Bu model sayesinde önden çekilmiş görüntülere karşılık gelen profil görüntüsü oluşturulabilmiştir. Bu işlem her ne kadar üç boyutlu bir model kullanılarak daha efektif bir biçimde gerçekleştirilebilecekse de (Blanz, 1998) Cootes çalışmasında (T.F. Cootes, 2004) iki boyutta da başarılı sonuçlar alınabileceğini göstermiştir. Geliştirilen ikili model ayrıca AGM aramasını sınırlandırmak için (G. Edwards, 1998) kullanılabilir, böylece aynı anda önden ve profilden geliştirilen modeller iki görüntüye eşlenebilir.

4.7.1 Eğitim Verisi

Görünüm modelinin, farklı bakış açılarından yüzü nasıl temsil ettiğini görmek için insanların -180 dereceden 180 dereceye kadar olan görüntüleri Şekil 4.11’de gösterildiği şekilde çekilmiştir. Eğitim setindeki görüntüler beş gruba ayrılmıştır (sol, sol yarım, önden, sağ yarım ve sağ). İki grup arasında kalan ve hangi gruba dâhil edilmesi kararlaştırılamayan görüntüler her iki gruba da dâhil edilmişlerdir. Bu sayede yakın bakış açıları arasındaki ilişki öğrenilmeye ve bakış açıları arasında daha pürüzsüz bir geçiş sağlanmaya çalışılmıştır.

Cootes çalışmasında Şekil 4.11’de gösterildiği şekilde üç ayrı model eğitmiştir. Profil modeli üzerinde 234 işaret bulunan 15 ayrı kişinin görüntülerinden oluşturulmuştur. Yarım profil modeli 82 görüntü üzerinden, önden bakış açısını tanıyan model ise 294 görüntüden oluşturulmuştur. Görüntülerin yansımaları eğitim verisini büyütmek için kullanılmıştır.

Şekil 4.13 ilk iki görünüm model parametresini, c_1, c_2 , değiştirmenin görüntü üzerindeki etkilerini göstermektedir. Bu parametreleri değiştirmek hem şekil hem de doku üzerinde değişikliğe yol açmıştır.



$c_1 \pm 2$ standart sapma $c_2 \pm 2$ standart sapma

Şekil 4-13 Yüz modelinin ilk iki parametresinin yüz modeli üzerindeki etkisi

4.7.2 Pozu Tahmin Etmek

Model parametrelerinin bakış açısına, θ , yaklaşık olarak (4.17)'deki gibi bağlı olduğu kabul edilmiştir:

$$c = c_0 + c_x \cos(\theta) + c_y \sin(\theta) \quad (4.17)$$

(4.17) içinde c_0 , c_x ve c_y eğitim verisinden tahmin edilen vektörlerdir. Çalışmada, sadece dik eksen üzerinde dönüş, kafayı döndürmek, hesaba katılmıştır. Baş sallamak da aynı şekilde hesaba katılabilir.

Cootes çalışmasında her bir eğitim örneği için kafanın açısını, θ_i , ± 10 derecelik bir hata payıyla hesaplamayı başarmıştır. Bu işlem için öncelikle en uygun model parametreleri, c_i , hesaplanmış ve c_0 , c_x ve c_y 'yi öğrenmek için $\{c_i\}$ ile vektörler $\{(1, \cos(\theta_i), \sin(\theta_i))\}$ arasında regresyon uygulanmıştır.

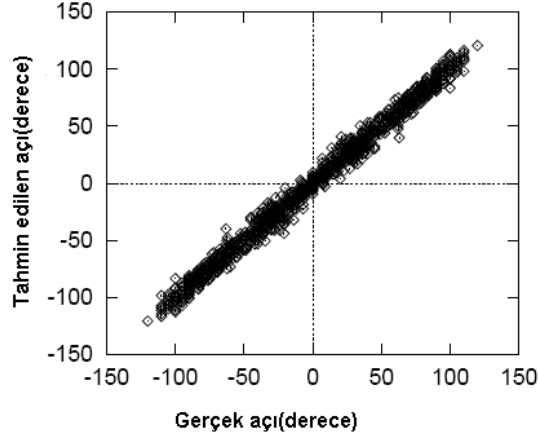
Parametreleri $\mathbf{c}$ olan yeni verilen bir örnekte kafanın açısını hesaplamak için: R_c^{-1} , $(c_x | c_y)$ matrisinin sol pseudo-tersi olsun* ve (4.18) sağlansın:

$$(x_a, y_a) = R_c^{-1}(c - c_0) \quad (4.18)$$

Bu durumda bakış açısının en yaklaşık değeri $\tan^{-1}(y_a / x_a)$ 'dır.

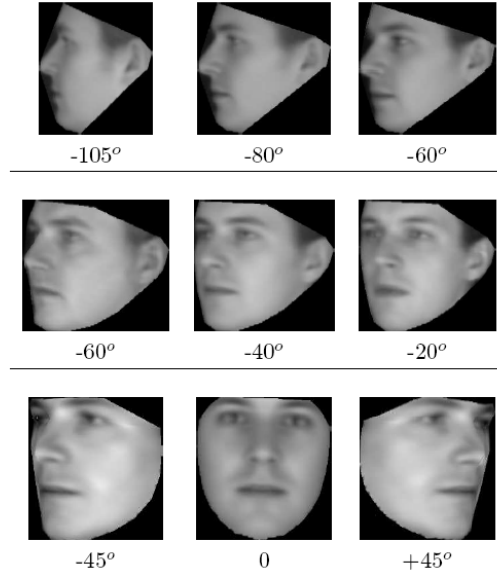
Şekil 4.14 eğitim setinde tahmin edilen açıları ve gerçek açıları her bir model için göstermektedir. Bu durumda (4.17) denklemi açığı bulmak için kabul edilebilir bir parametre modelidir.

* $R_c^{-1}(c_x | c_y) = I$



Şekil 4-14 Eğitim seti üzerinde gerçek açının tahmin edilen açıyla karşılaştırılması

Şekil 4.15’de ise (4.17) denkleminde bakış açısının, θ , değiştirilmesi sonucunda oluşturulan görüntüler gösterilmektedir.



Şekil 4-15 Üç yüz modelinde bakış açısının değiştirilmesi sonucunda oluşan görüntüler

4.8 Direk Görünüm Modeli

Hou vs. (X. Hou, 2001) yüz için doku ve şekil bileşenleri arasındaki ilişkinin çoktan bire, bir şekil birden fazla doku içerebilirken hiçbir doku birden fazla şekille ilişkili olamaz, olduğunu iddia etmişlerdir. Böylece şekil direk olarak doku bilgisinden tahmin edilebilir, bu da daha hızlı ve güvenilir arama algoritması elde etmemizi sağlayabilir.

Şekli dokudan tahmin etmek için eğitim seti örnekleri üzerinde regresyon kullanılabilir:

$$b_s = R_{t-s} b_t \quad (4.19)$$

Bu durumda arama algoritması:

1. Bir ilk pozu, $\mathbf{t}$, ve parametreleri b_s , b_t ve u varsay.
2. Modeli kullanarak imajı ve normalize edilmiş dokuyu, $\mathbf{g}$, hesapla.
3. Doku parametrelerini b_{tex} ve $\mathbf{g}'$ 'ye en uygun normalizasyonu $\mathbf{u}$ hesapla.
4. $b_s = R_{t-s} b_t$ denklemini kullanarak şekli güncelle.
5. $t = t - R_t \delta \mathbf{g}$ denklemini kullanarak pozu güncelle.

Her bir adımda en çok zaman alan imajın sentezlenmesi ve poz ve doku parametre güncellemesidir. Bu yüzden orijinal AGM'den daha hızlı çalışacaktır. Hou vs. yüz üzerine yaptıkları deneylerde arama performansının, hem hız hem de yakınsama açısından, daha iyi olduğunu göstermişlerdir.

Bu metodun varsayımı olan doku ve şekil bileşenleri arasındaki ilişkinin çoktan bire olduğu genel olarak doğru değildir. Bazı durumlarda şekil dokudan bağımsız olabilir. Mesela boyutları değişen ama rengi sabit bir dikdörtgen buna örnek olarak gösterilebilir. (T.F.Cootes, 2002)

4.9 Ters Kompozisyon AGM

Baker ve Matthews (Matthews, 2001) AGM algoritması içinde nasıl güncelleme ve hangi adımda bir minimizasyon yapıldığına göre bu algoritmanın bir imaj hizalama algoritmaları grubuna dâhil edilebileceğini göstermişlerdir.

Parametre güncellemelerinin nasıl yapıldığına göre hizalama algoritmaları iki sınıfa ayrılır. Bunlardan birincisi parametre değerlerine küçük değerler ekleme metodu (*additive*), diğeri ise bir çarpıtma fonksiyonu kullanarak parametreleri değiştirme metodudur (*compositional*).

Sabit bir doku şablonumuz, $T(x)$, olduğunu varsayalım. Bu şablonu bir hedef görüntüye, $I(x)$, hizalamaya çalışıyoruz. $x = (x, y)^T$ kaynak, $y = (x, y)^T$ hedef görüntüdeki koordinatlarını gösterebilir, $p = (p_1, \dots, p_n)^T$ parametre vektörümüz olsun, $W(x; p)$ de çarpıtma fonksiyonumuzu temsil etsin. Bu durumda hizalamanın amacı aşağıdaki denklemi minimize etmektir:

$$\sum_x [I(W(x; p)) - T(x)]^2 \quad (4.20)$$

Döngülü bir yaklaşım olduğunu varsayarsak her bir adımda eşleşmenin daha iyi olması için parametrelerin nasıl değişeceğini gösteren bir δp değeri hesaplanmalıdır. Baker ve Matthews çalışmalarında dört şekilde bunun çözülebileceğini anlatmışlardır:

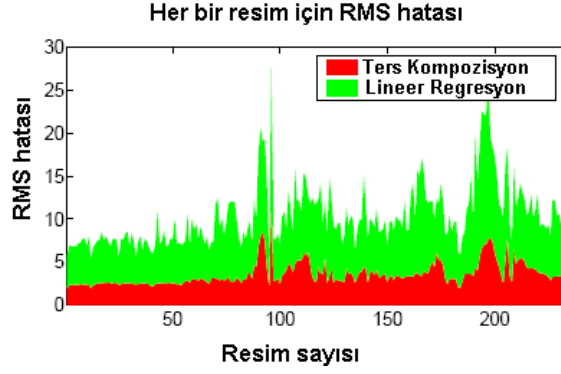
Ekleme: δp 'yi $\sum_x [I(W(x; p + \delta p)) - T(x)]^2$ fonksiyonunu minimize edecek şekilde seç.

Kompozisyon: δp 'yi $\sum_x [I(W(W(x; \delta p); p)) - T(x)]^2$ fonksiyonunu minimize edecek şekilde seç.

Ters Ekleme: δp 'yi $\sum_y \left| \frac{\partial W^{-1}}{\partial y} \right| [I(y) - T(W(y; p + \delta p)^{-1})]^2$ fonksiyonunu minimize edecek şekilde seç.

Ters Kompozisyon: δp 'yi $\sum_x [T(W(x; \delta p)) - I(W(x; p))]^2$ fonksiyonunu minimize edecek şekilde seç.

AGM'nin orijinal formülasyonu eklemelidir ama hata imajıyla parametre güncellemeleri arasında sabit bir ilişki varsayımını yapar ki bu varsayım her zaman tatmin edici değildir. Şekil 4.16 da görüldüğü gibi Baker ve Matthews çalışmalarında ters kompozisyon kullanarak AGM kadar hızlı çalışabilen ama AGM den daha iyi yakınsama yapabilen sonuçlara ulaşmışlardır.



Şekil 4-16 236 görüntüden oluşan bir film için yakınsama sonucunda oluşan RMS hataları. Ters kompozisyon lineer regresyondan çok daha iyi sonuçlar üretmiştir

Ayrıca Cootes vs. (T.F.Cootes, 2002) AGM üzerine yapılan varyasyonlar üzerine yaptığı çalışmada ters kompozisyonun parametre güncellemeleri için en şık ve matematiksel olarak doğru metot olduğunu söylemiştir.

Ters kompozisyon metodunun güncelleme adımını aşağıdaki gibi gösterebiliriz:

$$\delta p = - \sum_x H^{-1} \left[\nabla T \frac{\partial W}{\partial p} \right]^T [T(x) - I(W(x; p))] \quad (3.21)$$

$$H = \sum_x \left[\nabla T \frac{\partial W}{\partial p} \right]^T \left[\nabla T \frac{\partial W}{\partial p} \right] \quad (3.22)$$

Yukarıdaki denklemlerde Jacobian $\frac{\partial W}{\partial p}(x;0)$ 'da hesaplanmaktadır. Hessian matrisi(H) p 'den bağımsız olduğu için döngü sırasında matrisin değeri sabittir ve önceden hesaplanıp kullanılabilir.

Görünüm modelindeki gibi üçgenlere ayrılmış bir ağ kullanırken parametre güncellemesi yapmak işlemsel açıdan önemsiz değildir ama bunun yerine birinci dereceden bir yakınsama makul şekilde kullanılabilir.

Bu algoritmanın işleyişi sırasında birleştirilmiş görünüm modelinde ters kompozisyonun yararlarından faydalanmak için şekil ve dokunun birbirinden bağımsız olduğunu varsaymak gereklidir. Bu durumda parametreler yakınsanana kadar algoritmanın adımları aşağıdaki gibidir (T.F. Cootes, 2004):

1. Verilen parametrelere göre $p = (b_s^T, t^T)^T$ imajı yarat ve normalize et ve g 'yi hesapla.
2. Yapılacak güncellemeyi hesapla $\delta p = R(g - \bar{g})$.
3. Parametreleri $T_p(.) \rightarrow T_p(T_{\delta p}(.))$ olacak şekilde güncelle.
4. Döngüye devam et.

Güncelleme matrisi, R , Jacobian matrisinin doku modelinin sıfır uzayına izdüşümü olan A matrisinin pseudo tersi olarak değerlendirilebilir. Böylece i . kolonu aşağıdaki gibi hesaplanan A matrisine karşılık $R = A^T(A^T A)^{-1}$ şeklindedir.

1. Her bir eğitim örneği için en iyi şekli ve pozu hesapla p
2. Şu anki parametre i için küçük yer değiştirmeler yarat δp_j
3. Yeri değiştirilmiş parametreleri, p' , $T_{p'}(.) = T_p(T_{\delta p_j}(.))$ olacak şekilde yap (δp_j 'nin i . elemanı hariç tüm elemanları sıfırdır.)
4. Yeri değiştirilmiş olarak imajı sentezle ve normalize ederek g_s 'yi hesapla
5. Doku alt uzayındaki bileşeni dışarı çıkar, $b_g = P_g^T(g_s - \bar{g})$ dersek $\delta_{g_j} = (g_s - \bar{g}) - P_g p_g$
6. A matrisinin i . kolonunu $w_j = \exp(-0.5 \delta p_j^2 / \sigma_i^2)$ dersek $a_i = \frac{1}{\sum w_j} \sum w_j \delta g_j' / \delta p_j$ eşitliğini kullanarak yap (σ_i^2 parametre yer değiştirmelerinin varyansdır)
7. $R = A^T(A^T A)^{-1}$ şeklinde eşitle

4.10 Optimum Performans İçin Modelin Yakınsamasını Ayarlamak

AGM yakınsama işlemini bitirdiğinde en iyi sonuca yakın bir sonuca ulaşacaktır ama tam doğru pozisyona gelmemiş olabilir. En iyi sonuca ulaşamamasının en önemli nedenlerinden biri AGM araması sırasındaki optimizasyon problemlerinin eğitim setindekilere benzemesi gerektiği şeklindeki zayıf varsayımdır. Stegmann ve Fisker (Stegmann, 2000) (M. B. Stegmann, 2001) genel amaçlı bir iyileştirici yardımıyla sonucu daha da iyileştirmenin mümkün olduğunu göstermişlerdir. Bu yaklaşım AGM araması sonucunda zaten en iyiye

yakın bir sonuca geleceğimiz için mantıklıdır. Böylece optimizasyon makul bir süre içerisinde gerçekleştirilebilir. Düşünülen optimizasyon metotları:

- Düşüm(*gradient*) temelli metotlar: En dik düşüş (*Steepest descent*), eşlenik düşüm (*conjugate gradient*), Quasi-Newton
- Düşüme dayanmayan(*non-gradient*) metotlar: Desen(*pattern*) arama
- Rastgele sentezlemeye(*random-sampling*) dayalı metotlar: Benzetimli tavlama(*simulated annealing*)

Stegmann ve Fisker çalışmalarında benzetimli tavlama metodunun kullanılmasını tavsiye etmişlerdir. Bunun sebebi AGM’de kullanılan amaç işlevi $-\|\delta g\|^2$ - büyük ihtimalle konveks değildir. Bu açıdan benzetimli tavlama metodu yerel minimumlardan kaçabilirken diğer optimizasyon metotlarının yerel minimumlara yakalanma olasılığı daha fazladır.

4.11 Kuvvetli İstatistik Uygulamak

Stegmann ve Fisker (Stegmann, 2000) (M. B. Stegmann, 2001) çalışmalarında kuvvetli istatistik kullanarak AGM’nin başarısının arttırılabileceğini göstermişlerdir. AGM optimizasyonu doku farklılıklarını $-\|\delta g\|^2$ - kullanarak ilerler. Optimizasyonun kendini değerlendirme ölçüsüne benzerlik ölçüsü(*similarity measure*) adı verilmiştir. Benzerlik ölçüsü matematiksel olarak belirsiz olduğu için $\|\delta g\|^2$ benzerin ne demek olduğunu açıklar. Benzerliğin ne kadar kuvvetli olduğu dağılımın ortalamasına çok uzak olan değerlere ne karşı ne kadar dirençli olunduğuyla doğru orantılıdır.

Modeli yakınsama problemini çözmek, bir parametre kümesinin $c = [c_1, \dots, c_p]^T$ yapılan ölçümlere(mesela bir görüntü) $g = [g_1, \dots, g_m]^T$ uyumlu hale getirilmesi için değiştirildiği bir süreçtir. Bu da aslında bir minimizasyon problemidir:

$$E = \sum_{i=1}^m \rho(g_i - u(i, c), \sigma_s) = \sum_{i=1}^m \rho(e_i, \sigma_s) \quad (3.23)$$

(8.23)’de u i. ölçümdeki model yapısını döndüren bir fonksiyon, σ_s ise neyin dağılımın ortalamasına uzak olduğuna karar vermek için kullanılan bir büyütme parametresidir.

ρ gözlemler arasındaki farklılıkların ağırlığını belirler, bu fonksiyona hata normu(*error norm*) da denir. En çok kullanılan hata normu ikinci dereceden fonksiyondur:

$$\rho(e_i) = e_i^2 \quad (3.24)$$

AGM’de kullanılan hata normu ise:

$$E = \sum_{i=1}^m (g_{\text{model}} - g_{im})^2 = \sum_{i=1}^m \delta(g_i)^2 = |\delta g|^2 \quad (3.25)$$

Ancak bu norm ortalamaya uzak olan değerlere karşı çok hassastır. Bunun sebebi ikinci dereceden bir fonksiyonun hızlı büyümesi ve ortalamaya uzak olan değerlerin sonuca çok fazla katkı yapmasıdır. Bunu engellemek uygulanabilecek bir yöntem gözlemler arasındaki farka bir üst sınır koymaktır. Buna kesilmiş ikinci derece norm(*truncated quadratic norm*) adı verilir:

$$\rho(e_i, \sigma_s) = \begin{cases} e_i^2 & e_i \leq \sqrt{\sigma_s} \\ \sigma_s & e_i > \sqrt{\sigma_s} \end{cases} \quad (3.26)$$

Başka bir yöntem ise belirli bir sınırdan sonra farkın büyümesini lineer hale getirmektir. Buna Huber en küçük-en büyük kestiricisidir(*Huber’s minimax estimator*) adı verilir:

$$\rho(e_i, \sigma_s) = \begin{cases} \frac{e_i^2}{2\sigma_s} + \frac{\sigma_s}{2} & e_i \leq \sqrt{\sigma_s} \\ |e_i| & e_i > \sqrt{\sigma_s} \end{cases} \quad (3.27)$$

Kullanılan başka bir norm ise Aktif Damla metodunda da kullanılan Lorentzian kestiricisidir(*Lorentzian estimator*):

$$\rho(e_i, \sigma_s) = \log \left(1 + \frac{e_i^2}{2\sigma_s^2} \right) \quad (3.28)$$

Stegmann (Stegmann, 2000) çalışmasında işin içine varyansı da kattığı ve yüksek varyansın olduğu gözlemlere karşı daha az duyarlı olduğu için Mahalanobis uzaklığıyla ilişkili bir benzerlik ölçüsü kullanılmasını tavsiye etmiştir. Ancak $\sum$ ’nın doku farklılıklarının Kovaryans matrisi olduğu bağımlı uzaklık:

$$\rho(e_i) = (g - \bar{g}) \sum^{-1} (g - \bar{g}) \quad (3.29)$$

Yerine pikseller arasında bağımsızlığın olduğunu (Bir imaj içerisinde pikseller birbirinden bağımsız değildir ancak işlemsel açıdan (8.29)'un hesaplanması güçtür) kabul edip aşağıdaki gibi bir norm kullanılmasını uygun bulmuşlardır:

$$\rho(e_i) = \frac{e_i^2}{\sigma_i^2} \quad (3.30)$$

(8.30) denkleminde σ_i^2 i'inci pikselin varyansının en büyük olabilirlik kestirimidir.

Stegmann ve Fisker hem optimum performans için modelin yakınsamasını ayarlamak için benzetimli tavlama kullanmışlar hem de kuvvetli istatistik uygulayıp Şekil(8.17) de görüldüğü gibi temel AGM'den daha başarılı sonuçlara ulaşmışlardır.



Şekil 4-17: Sol tarafta temel AGM ile bir el röntgeninde kemikler bulunmaya çalışılmıştır. Sağ taraftaki görüntüde ise hem benzetimli tavlama hem de Lorentzian normu kullanılmıştır. Sonuç olarak hata 7.0 pikselden 2.4 piksele kadar gerilemiştir

5 SONUÇLAR

5.1 Tartışma

Bu tezin temel yazılma amacı:

- İstatistiksel şekil ve doku modellerinin ve bu modellerin arka planlarındaki teorik detayların verilmesi
- AGM'nin teorik detaylarının verilmesi
- AGM metodunu kullanan bir yüz tanıma programı yazılması ve AGM'nin yüz tanımadaki başarısının ölçülmesi
- AGM'nin üzerine yapılabilecek uzantıların tartışılması.

Birinci ve ikinci madde ile ilgili tezin akışı sırasında istatistiksel şekil ve doku modelleri hakkında detaylı bilgi verildi. AGM'nin istatistiksel şekil ve doku modellerini ve önceden elde edilmiş bilgiyi kullanarak nasıl daha kompakt bir model oluşturduğu detaylı olarak anlatıldı.

Üçüncü maddeyle ilgili olarak AGM algoritması kullanarak poz, ışık ve ifade farklılıklarına dayanıklı bir yüz tanıma mekanizması geliştirildi ve AGM'nin başarısı test edildi. Yapılan testler sonucunda AGM kullanarak başarılı sonuçlara ulaşıldı.

Son olarak ise temel AGM'nin performansını geliştirmek için literatürde önerilen metotlar ve bu metotlar kullanılarak yapılan testler anlatıldı.

Tez sırasında AGM'nin kullanılmasının sebebi AGM'nin önceden öğrenilen veriyi kullanarak eğitim setine benzeyen görüntüler sentezleyebilmesi, güvenilir ve hızlı bir metot olmasıdır. AGM bu tez sırasında yüz tanıma için kullanılmış olsa da algoritma değiştirilmeden daha farklı görüntü problemlerine de sadece eğitim kümesinin değiştirilmesiyle uyarlanabilir. Ayrıca verilen uzantılar sayesinde karşılaşılan problemlere özel koşullar kullanılarak da AGM'nin başarısını arttırmak mümkündür. AGM'nin güçlü ve zayıf kaldığı noktaları vermek gerekirse:

5.1.1 AGM'nin Güçlü Olduğu Noktalar

- AGM modeli geliştirildikten sonra, birçok farklı bilgisayarla görme probleminde farklı eğitim setleriyle kullanılabilir. Bunlara örnek olarak: Kalp, diz ve beyin MRI'larında segmentasyon işlemleri ya da yüz tanıma ya da takip etme problemleri verilebilir.
- Hem doku hem de şekil bilgisi kullanıldığından AGM verilen problemi geniş bir perspektifle ve verilen probleme odaklanarak değerlendirir.
- AGM metodu eğitim kümesiyle ilk eğitilme işleminden sonra çok hızlı bir şekilde verilen görüntü üzerinden işler.

- AGM metodu eğitim işleminden sonra eğitim kümesindeki benzeyen gerçekçi örnekler sentezleyebilir.
- AGM metodu, gürültü olan ortamlarda bile başarılı bir şekilde çalışabilir.

5.1.2 AGM Metodunun Zayıf Kaldığı Noktalar

- AGM metodu şekil değiştirme potansiyeli olan cisimler üzerinde kullanılamaz. Mesela hücreler, bulutlar veya şehirler AGM metoduyla çalışmak için uygun değildir. Eğitim kümesinde yapılan işaretleme işlemindeki noktalar kümedeki her elemanda ortak bulunmalıdır.
- AGM metodunda eğitim kümesinin oluşturulma işlemi bu metodun en uğraştırıcı bölümüdür. El ile tek tek kümedeki her eleman üzerinde işaretleme yapılmalıdır.
- AGM ancak şekil kümesindeki benzeyen örnekler yaratabildiği için eğer eğitim setindeki eleman sayısı ya da eğitim setindeki elemanlar arasındaki varyasyon az ise AGM dar bir alanda çalışır.
- AGM metodu iyi bir ikklendirmeye ihtiyaç duyar.
- Eğer test imajında bazı noktalar kapatılmışsa AGM metodunun arama metodu başarısız olacaktır.

6 KAYNAKLAR

A. L. Yuille P. W. Hallinan, and D. S. Cohen Feature extraction from faces using deformable templates [Journal] // Int. Jour. of Computer Vision. - 1992. - pp. 99–111.

Blanz T. Vetter ve V. Estimating coloured 3d face models from single images: an example based approach [Conference] // 5th European Conference on Computer Vision. - Berlin : Springer, 1998. - pp. 499-513.

Bookstein F. L. Landmark methods for forms without landmarks: localizing group differences in outline shape [Article] // Medical Image Analysis. - 1997. - pp. 225–244.

C. Nastar B. Moghaddam ve A. Pentland Generalized image matching: Statistical learning of physically-based deformations [Conference] // 4th European Conference on Computer Vision. - Cambridge, UK : [s.n.], 1996. - pp. 589-598.

Covell M. Eigen-points: Control point location using principal component analysis [Conference] // 2nd International Conference on Automatic Face and Gesture Recognition. - Killington, USA : [s.n.], 1996. - pp. 122-127.

Duncan L.H. Staib and J.S. Boundary Finding with parametrically deformable models [Article] // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 1992. - pp. 1061-1075.

F. Pighin R. Szeliski ve D. Salesin Resynthesizing facial animations through 3d model-based tracking [Conference] // 7th International Conference on Computer Vision. - 1999. - pp. 143-150.

G. Edwards C.J. Taylor ve T.F. Cootes Interpreting face images using active appearance models [Conference] // 3rd International Conference on Automatic Face and Gesture Recognition. - Japan : [s.n.], 1998. - pp. 300-305.

Goodall C. Procrustes methods in the statistical analysis of shape [Journal] // Journal of the Royal Statistical Society. - 1991. - pp. 285-339.

Isidoro S. Sclaroff ve J. Active blobs [Conference] // 6th International Conference on Computer Vision. - 1998. - pp. 1146–1153.

K.N. Walker T. F. Cootes, ve C. J. Taylor Determining correspondences for statistical models of facial appearance [Conference] // In Proc. Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition. - 2000. - pp. 271-6.

Kruger N. An algorithm for the learning of weights in discrimination functions using a-priori constraints [Conference] // IEEE transactions on Pattern Analysis and Machine Intelligence. - 1997. - pp. 764-768.

M. B. Stegmann B. K. Ersbøll, R. Larsen FAME - A Flexible Appearance Modelling Environment [Conference] // IEEE Transactions on Medical Imaging. - 2003. - Vol. 22(10). - pp. 1319-1331.

M. B. Stegmann R. Fisker ve B. K. Ersbøll Extending and applying active appearance models for automated, high precision segmentation in different image modalities [Conference] // Scandinavian Conference Image Analysis. - 2001. - pp. 90-97.

M. Kass A. Witkin, and D. Terzopoulos Snakes: Active contour models [Journal] // Int. Jour. of Computer Vision. - 1988. - pp. 321-331.

Mardia I. L. Dryden and K. V. Statistical Shape Analysis [Book]. - [s.l.] : John Wiley & Sons, 1998.

Matthews Simon Baker ve Iain Equivalence and Efficiency of Image Alignment Algorithms [Conference] // IEEE Conference on Computer Vision and Pattern Recognition. - 2001. - Vol. 1. - pp. 1090-1097.

Nayar H. Murase ve S. Learning and recognition of 3d objects from appearance [Conference] // International Journal of Computer Vision. - 1995. - pp. 5-25.

Nielsen A. A. Analysis of Regularly and Irregularly Sampled Spatial, Multivariate, and Multi-temporal Data [Book]. - PhD thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby : [s.n.], 1994.

Pentland B. Moghaddam ve A. Face Recognition using view-based and modular eigenspaces [Conference] // SPIE, volume 2277. - 1994. - pp. 12-21.

Poggio T. Ezzat ve T. Facial Analysis and synthesis using image-based models [Conference] // 2nd International Conference on Automatic Face and Gesture Recognition. - Killington, Vermont : [s.n.], 1996. - pp. 116-121.

S. Romdhani A.Psarrou ve S. Gong On utilising template and feature-based correspondence in multi-view appearance models [Conference] // 6th European Conference on Computer Vision. - [s.l.] : Springer, 2000. - pp. 799-813.

S. Romdhani S. Gong ve A. Psarrou A multi-view non-linear active shape model using kernel PCA [Conference] // 10th British Machine Vision Conference. - Nottingham : BMVA Press, 1999. - pp. 483-492.

Sami Romdhani Jeff Ho, Thomas Vetter, David J. Kriegman Face Recognition Using 3-D Models: Pose and Illumination [Conference] // In Proceedings of the IEEE. - 2006. - pp. 1977-1999.

Stegmann Mikkel Bille Active Appearance Models [Report]. - Lyngby : IMM, Technical University of Denmark, 2000.

T. F. Cootes C. J. Taylor, D. H. Cooper, ve J. Graham Active shape models - their training and application. Computer Vision and Image Understanding [Conference]. - 1995. - pp. 38–59.

T. F. Cootes G. Edwards, ve C. J. Taylor A comparative evaluation of active appearance model algorithms [Conference] // Ninth British Machine Vision Conf. - Southampton : [s.n.], 1998. - pp. 680–689.

T. F. Cootes G. J. Edwards, and C. J. Taylor Active appearance models [Conference] // European Conf. on Computer Vision. - [s.l.] : Springer, 1998. - pp. 484–498.

T.F. Cootes C.J. Taylor Statistical Models of Appearance for Computer Vision [Report]. - Manchester : [s.n.], 2004.

T.F. Cootes G. J. Edwards, ve C. J. Taylor Comparing active shape models with active appearance models [Conference] // British Machine Vision Conf. - 1999. - pp. 173–182.

T.F.Cootes P.Kittipanya-ngam Comparing Variations on the Active Appearance Model Algorithm [Conference] // BMVC2002. - 2002. - Vol. 2. - pp. 837-846.

Taylor C.B.H. Wolstenholme ve C.J. Wavelet compression of active appearance models [Conference] // In Medical Image Computing and Computer-Assisted Intervention. - 1999. - pp. 544–554.

Taylor G.J. Edwards ve T. F. Cootes ve C. J. Advances in Active Appearance Models [Conference] // Proc International Conference on Computer Vision. - 1999. - pp. 137-142.

Taylor T. F. Cootes ve C. J. Modelling object appearance using the gray-level surface [Conference] // British Machine Cision Conference. - York, England : BMVA Press, 1994. - pp. 479-488.

X. Hou S. Li, H. Zhang ve Q.Cheng Direct appearance models [Conference] // Computer Vision and Pattern Recognition Conference. - 2001. - Vol. 1. - pp. 828-833.

Yacoob M. J. Black ve Y. Recognizing facial expressions under rigid and non-rigid facial motions [Conference] // 1st International Workshop on Automatic Face and Gesture Recognition 1995. - Zurich : [s.n.], 1995. - pp. 12-17.

Yuxiao Hu Dalong Jiang, Shuicheng Yan, Lei Zhang ve Hongjiang zhang Automatic 3D Reconstruction for Face Recognition [Conference] // Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference. - 2004. - pp. 843- 848.

7 İNTERNET KAYNAKLARI

Stegmann Mikkel B. Free data sets for statistical models of shape [Online] // Active Appearance Models. - July 2008. - <http://www2.imm.dtu.dk/~aam/>.

Wikipedia Delaunay triangulation [Online] // Wikipedia. - May 11, 2008. - http://en.wikipedia.org/wiki/Delaunay_triangulation.

8 ÖZGEÇMİŞ

Doğum tarihi 13.09.1983

Doğum yeri Denizli

Lise 1998-2001 Denizli Anadolu Lisesi

Lisans 2001-2005 Sabancı Üniversitesi Bilgisayar Müh.

Yüksek Lisans 2005-2008 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Müh. Yüksek Lisans Programı

Çalıştığı kurum(lar)

2005-2008 Garanti Teknoloji