

# İÇİNDEKİLER

---

## Kullanım Kataloğu

### 1. Giriş

1.1. Giriş .....	2
------------------	---

### 2. Yöntemler

2.1. Nesne Takip Yöntemleri .....	2
2.2. Nesne Takibinde Karşılaşılan Temel Zorluklar .....	3
2.3. Nesne Takip Yöntemlerinin Karşılaştırılması .....	3
2.3.1. Nokta Tabanlı Nesne Takibi .....	4
2.3.2. Çekirdek Tabanlı Nesne Takibi .....	5
2.3.3. Silüet Takibi .....	7

### 3. Uygulama

3.1 Uygulama Kodları .....	10
----------------------------	----

### 4. Kaynakça

4.1. Kaynaklar .....	14
----------------------	----

## GİRİŞ

- Nesne takibi, “bilgisayarlı görme” alanı içerisinde önemli bir göreve sahiptir. En basit tanımıyla, nesne takibi, video kaydı içerisinde hareket etmekte olan bir nesnenin hareket yörüngesini tahmin etme problemidir.
- Nesne takibi ile ilgili literatürde doyurucu sayıda yaklaşım önerilmiştir. Önerilen bu yaklaşımların temel farkları aşağıdaki sorulara verilen yanıtlar yardımıyla ortaya çıkmaktadır.
  - ✓ Nesne takibi için hangi nesne sunumu daha uygundur?
  - ✓ Takip işleminde nesnelere ait hangi özellikler kullanılmalıdır?
  - ✓ Belirlenen nesne özellikleri nasıl modellenir?

## NESNE TAKİP YÖNTEMLERİ

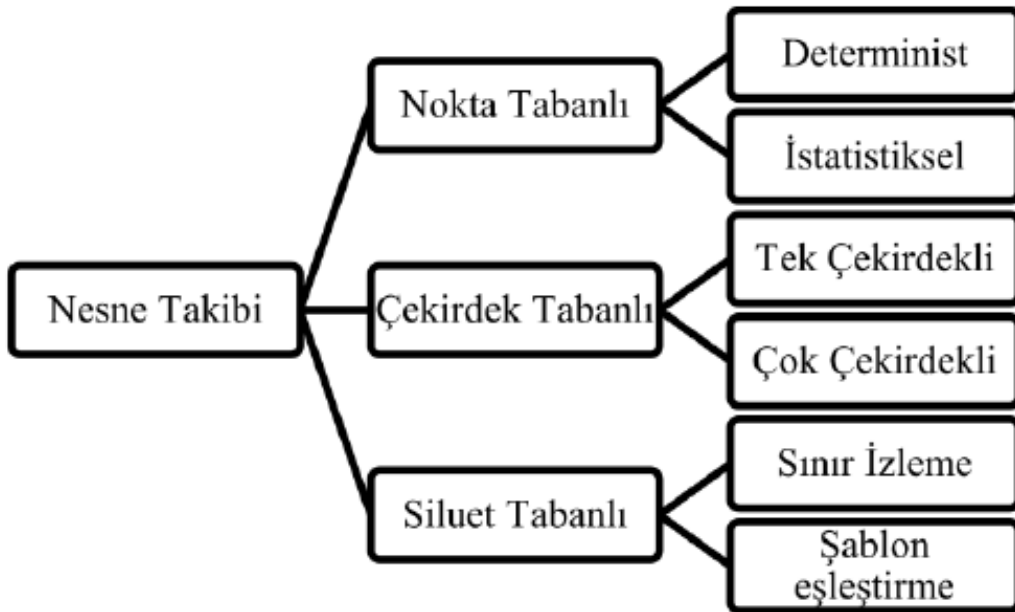
- Nesne takip yöntemleri iki temel görevi içermektedir (Forsyth vd., 2005):
  - 1) Takip edilecek nesnenin güncel imge içerisindeki konumunun tespit edilmesi (object detection, lifting)
  - 2) İmgeler boyunca konumları belirlenen nesneler arasındaki veri bağı ilişkisinin kurulması(data association)
  - 3) Nesne hareketleri veya görünümleri üzerinde belirli sınırlamalar yapılarak nesne takip problemi basitleştirilebilir. Hemen tüm nesne takip algoritmaları, nesne hareketlerinin ani değişimlere değil de yumuşak hareketlere sahip olduğunu varsayar.
  - 4) Ayrıca, nesne hareketlerinin sabit hız veya ivmeye sahip olduğu düşünülürse nesne takip işlemi oldukça basitleşir. Buna ilaveten, nesnelerin sayıları, büyüklükleri, görünümleri veya sahip oldukları fiziksel şekillerin önceden belirtilmesiyle de, nesne takip algoritmalarının karmaşıklığı azaltılır.

# NESNE TAKİBİNDE KARŞILAŞILAN TEMEL ZORLUKLAR

Nesne takibinde karşılaşılan temel zorluklar aşağıda listelendiği gibidir (Bettencourt ve Somers, 2007):

- 3 boyutlu gerçek dünya verilerinin 2 boyutlu görüntü alanına yansıtılmasıyla meydana gelen bilgi kaybı
- Görüntü üzerinde meydana gelen gürültüler
- Nesne hareketlerinin karmaşık oluşu
- Nesnelerin ayırt edilebilir fiziksel bir yapıya sahip olamayışları
- Nesne görünümünün bir kısmının veya tamamının engellenmesi (occlusion)
- Karmaşık nesne şekilleri
- Ortamdaki ışık miktarının değişimi
- Gerçek zamanlı uygulamaların gereksinimleri

## NESNE TAKİP YÖNTEMLERİNİN SINIFLANDIRILMASI



**Şekil 1. Nesne takibi yöntemlerinin sınıflandırılması (Yılmaz vd., 2006)**

Şekil 1’de listelendiği gibi yöntemlerin kullanıldığı uygulama alanları da aşağıdaki gibi listelenebilir:

- Hareket tabanlı tanımlama (Örn: Hareket yörüngesi bilinen bir tümörünün akciğer üzerindeki konumun tespiti (Berbeco vd., 2005)).
- Gizlice izleme (Örn: Şüpheli aktivitelerin veya istenmeyen hareketlerin tespit edilebilmesi için sabit bir ekran görüntüsünün izlenmesi (Haritaoglu vd., 2000)).

## 1. NOKTA TABANLI NESNE TAKİBİ

- Bu takip yönteminde takip edilen her bir nesne tek bir nokta ile ifade edilir.
- Bu yöntem ile güncel imgede takip edilecek her bir nesneye bir nokta aktarıldıktan sonra bu noktalar ile önceki imgede tespit edilen noktalar arasındaki veri bağı ilişkisinin doğru bir şekilde oluşturulması beklenir.
- Bu alandaki yöntemler genellikle iki alt kategoride incelenir:
  - Deterministik Yöntemler
  - İstatistiksel Yöntemler

### 1.1. DETERMİNİSTLİK YÖNTEMLER

- Deterministik yöntemler, ardışık imgelerde bulunan nesnelerin birbirlerine bağlanma maliyetlerini tanımlar. Nesneler arasındaki bağlanma maliyetinin tanımlanması için nesne hareketleri üzerinde aşağıda belirtilen sınırlamalar göz önüne alınır:
- **Yakınlık:** Bir imgeden diğer bir imgeye geçerken nesne pozisyonlarının önemli ölçüde değişmeyeceğini ima eder.
- **Maksimum hız:** Nesnelerin hızları üzerinde bir üst sınır değeri tanımlanır ve sadece nesnelerin etrafında dairesel bir komşuluk içerisinde kalan muhtemel nesne adaylarının benzerlik değerleri göz önüne alınır.
- **Küçük hız değişimi (yumuşak hareket):** Nesnenin hız yönünün önemli bir ölçüde değişmeyeceğini ima eder.
- **Genel hareket:** Küçük bir komşuluktaki nesnelerin hızlarının benzer olması için sınırlama uygulanır. Bu sınırlama çoklu noktalar ile sunulan nesneler için uygundur.
- **Katılık:** 3-boyutlu dünyadaki nesneler genellikle katı bir biçime sahiptir. Bundan dolayı güncel bir nesne üzerindeki herhangi iki nokta arasındaki mesafenin değişmeyeceği düşünülür.

### 1.2. İSTATİSTİKSEL YÖNTEMLER

- Video algılayıcılarından elde edilen ölçümler her zaman gürültüye sahiptir.
- Ayrıca, nesne hareketleri birtakım istenmeyen etkilere maruz kalır.
- İstatistiksel yöntemler, nesnenin durum (pozisyon, hız ve ivmesinin) tahmini boyunca ölçüm değerlerini ve model belirsizliklerini göz önüne alarak nesne takibi problemini çözmeye çalışır.

## 2. ÇEKİRDEK TABANLI NESNE TAKİBİ

- Çekirdek tabanlı nesne izleyiciler, parametrik olmayan tahmin ediciler gurubu içerisinde yer almaktadır.
- Parametrik olmayan sistemlerde sabit bir fonksiyon yapısı söz konusu değildir ve bir tahmin gerçekleştirileceği zaman dağılıma ait tüm veri değerleri göz önünde bulundurulur. Bunun yanında parametrik sistemlerde sabit bir fonksiyon yapısı ve sabit parametre değerleri bulunmaktadır
- Çekirdek tabanlı nesne takibi yönteminin amacı, takip edilecek nesneyi basit bir geometrik şekil içerisine alarak şekil içerisinde kalan görünüm bilgisinin olasılık yoğunluk dağılımını elde etmek ve bu dağılımı ardışık video imgeleri boyunca takip edebilmektir
- Çekirdek tabanlı nesne takibi işlemini anlayabilmek için öncelikle bir nesnenin görünümüne ait histogram sunumunu ve böyle bir sunumun sahip olduğu dezavantajları bilmek gerekir.

**Tablo 2. Çekirdek fonksiyonları**

Çekirdek Fonksiyon Adı	Çekirdek Fonksiyonu ( $K(u)$ )
Uniform	$(1/2) * I( u  < 1)$
Triangle	$(1 -  u )I( u  < 1)$
Epanechnikov	$(3/4) * (1 - u^2)I( u  < 1)$
Quartic	$(15/16) * (1 - u^2)^2I( u  < 1)$
Triweight	$(35/32) * (1 - u^2)^3I( u  < 1)$
Gaussian	$(1/\sqrt{2\pi}) * \exp(-0.5u^2)$
Cosinus	$(\pi/4) * \cos(u\pi/2)I( u  < 1)$

- Çekirdek tabanlı nesne takibi işlemi tek boyutlu ve çok boyutlu veriler üzerinde gerçekleştirilebilir. Buna göre, takip edilecek nesnenin görünüm bilgisi aşağıdaki gibidir:

$$X_i = [X_{i1}, X_{i2}, \dots, X_{id}]^T, \quad i = 1, \dots, n$$

## 2.1. TEK BOYUTLU ÇEKİRDEK YOĞUNLUK TAHMİNİ

- Tek boyutlu veri kümesinin olasılık yoğunluk dağılımını hesaplanmak için Denklem 3'de belirtilen formül kullanılır. Buna göre, veri elemanının dizisi üzerindeki katkısı (olasılık yoğunluk değeri veya yoğunluk tahmini) şu şekilde hesaplanır:
- $f(x) = \frac{1}{n} \sum_{i=1}^n \left[ K\left(\frac{x-x_i}{h}\right) \right]$
- Bu formülde ( ) çekirdek fonksiyonu merkezi noktasına ve bant genişliğine sahiptir. Burada  $\int K(t) dt = 1$  eşitliğinin sağlanma zorunluluğu vardır.

## 2.2. ÇOK BOYUTLU ÇEKİRDEK YOĞUNLUK TAHMİNİ

- Çok boyutlu bir veri kümesinin olasılık yoğunluk dağılımı aşağıdaki kullanılarak hesaplanır.

$$\begin{aligned} \hat{F}_h(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} K\left(\frac{x_1-x_{i1}}{h}, \dots, \frac{x_d-x_{id}}{h}\right) \end{aligned}$$

### 3. SİLÜET TAKİBİ

- Takip edilmesi istenilen nesneler basit geometrik şekiller ile tanımlanamayan el, baş, omuz gibi karmaşık geometrik şekillere sahip olabilir.
- Siluet tabanlı yöntemler bu nesneler için doğru bir şekil tanımlayıcısı sağlarlar. Siluet tabanlı nesne takip edicilerinin asıl hedefi, önceki imgeler kullanılarak üretilen nesne modelini güncel imge içerisinde bulmaktır (Raykar ve Duraiswami, 2006).
- Bu modeller nesnenin renk histogramı, nesne kenarını veya sınır şeklini kullanır.
- Siluet tabanlı nesne takip edicileri “şekil karşılaştırmacılar” ve “sınır takip ediciler” olarak adlandırılan iki alt kategori altında incelemek mümkündür.

#### 3.1. SINIR TAKİBİ

- Şekil karşılaştırma yöntemlerinin aksine sınır takibi yöntemleri güncel imgedeki sınır çizgilerini başlangıç kabul ederek bir sonraki imgede yeni nesne sınırlarını arar.
- Sınır takibi algoritmalarının uygun sonuç vermesi için güncel imgedeki nesne ile bir sonraki imgede bulunan nesne alanlarının kesişimi olmak zorundadır.
- Sınır değerlendirme algoritmaları iki alt kategoride incelenebilir.
  - İlk yaklaşım, sınır biçim ve hareketini modellemek için durum uzay modellerini kullanır.
  - İkinci yaklaşım, gradient azalma gibi minimizasyon tekniklerini kullanarak sınır enerjisini minimize eder

## NESNE TAKİP YÖNTEMLERİNİN KARŞILAŞTIRILMASI

No	Kıstasın Tanımı
1	Aynı anda birden fazla nesne takibi gerçekleştirilebiliyor mu?
2	Dinamik arka plan modeline sahip mi?
3	Görüntü ekranına nesne giriş çıkışına izin veriliyor mu?
4	Eğitim aşamasına ihtiyacı ortadan kaldırılmış mı?
5	Başlangıç aşamasında nesne pozisyonlarının belirlenme ihtiyacı ortadan kaldırılmış mı?
6	Kapatma olayı gerçekleştiği zaman nesne takibi istenilen şekilde devam edebiliyor mu?
7	Ardışık imgeler arasındaki nesne bağı problemi için uygun bir çözüm sağlanabiliyor mu?
8	Gürültü, ışık miktarının değişimi, karmaşık nesne şekli ve hareketi gibi durumlarda takip edici nesneyi takip edebiliyor mu?
9	Gerçek zamanlı uygulama gereksinimleri karşılanabiliyor mu?

## NOKTA TABANLI NESNE TAKİBİ KARŞILAŞTIRMASI

ALGORİTMA	KİSTASLAR								
Nokta Tabanlı	1	2	3	4	5	6	7	8	9
Zhao vd. 2008	ü	X	X	ü	ü	ü	ü	ü	X
Haritaoglu vd. 2000	ü	X	ü	X	ü	ü	ü	ü	ü
Xue J. vd. 2008	ü	ü	ü	X	ü	ü	ü	ü	ü
Dong ve DeSouza, 2009	X	ü	X	ü	X	X	ü	ü	X
Veenman vd. 2001	X	X	X	X	X	X	ü	ü	ü



## ÇEKİRDEK TABANLI NESNE TAKİBİ KARŞILAŞTIRMASI

ALGORİTMA	KISTASLAR								
ÇekirdekTabanlı	1	2	3	4	5	6	7	8	9
Yilmaz, 2007	X	ü	X	ü	X	X	ü	ü	ü
Comaniu, 2003	X	X	X	ü	X	X	ü	ü	ü
Fan vd., 2005	ü	ü	X	ü	X	X	ü	ü	ü

## SİLÜET TABANLI NESNE TAKİBİ KARŞILAŞTIRMASI

ALGORİTMA	KISTASLAR								
ÇekirdekTabanlı	1	2	3	4	5	6	7	8	9
Cootes vd., 1998	ü	ü	X	X	ü	X	ü	ü	ü
Nascimento ve Marques, 2007	X	ü	X	X	X	X	ü	ü	ü
Zhu vd., 2007	X	X	X	ü	X	X	ü	ü	ü

## SONUÇ

- Bu çalışmada, video imgelerindeki hareketli nesnelerin takibini gerçekleştirebilen modern nesne takip yöntemleri hakkında detaylı bir araştırma yapılmış ve elde edilen bilgiler ışığında önerilen takip yöntemleri yapısal olarak sınıflandırılmıştır.
- Modern nesne takip algoritmalarının ayırt edici karakteristik özelliklerini belirleyebilmek amacıyla dokuz farklı kıstas belirlenmiş ve her bir kıstasa takip yöntemlerinin verdiği yanıtlar göz önüne alınarak sınıflandırma yapılmıştır. Elde edilen sonuçlar tablolar halinde listelenmiştir.
- Böylelikle okuyucuya, ortama ve şartlara bağlı olarak değişebilen nesne takip ihtiyacına en uygun yanıtı verebilecek olan nesne takip algoritmasını rahatlıkla seçebilme imkânı sunulmuştur.

## MATLAB UYGULAMASI

```
function [output] =
optical_flow(imgs, options)
t = tic;
position = [];
velocity = [];
if nargin == 0
    display('running
demo...')
    display('loading
data...')
    try
        load cat_data
    catch
        display('missing demo
file "cat_data.mat"')
        display('demo
aborted.')
        return
    end
    imgs = single(imgs);
end

if size(imgs,3) < 2
    error('need at least 2
image frames for optical flow
algorithm to operate')
end

if nargin < 2
    x_blk_size =
floor(.04*size(imgs,2));
    y_blk_size =
floor(.04*size(imgs,1));
else
    if
~isfield(options,'x_blk_size'
)
        x_blk_size =
floor(.04*size(imgs,2));
        display(sprintf('no
x_blk_size options found,
using %g',x_blk_size))
    end

    if
~isfield(options,'y_blk_size'
)
```

```
        y_blk_size =
floor(.04*size(imgs,1));
        display(sprintf('no
x_blk_size options found,
using %g',y_blk_size))
    end

    if
~isfield(options,'displayResu
lts')
        displayResults = 0;
    end
end

nframe = size(imgs,3);

for iframe = 1:nframe-1

    display(sprintf('processing
frame: %g of
%g...',iframe,nframe-1))

    frame1 =
squeeze(imgs(:,:,iframe));
    frame2 =
squeeze(imgs(:,:,iframe+1));
    %-----
    % Pozisyonu g re
    %  r nt n de i imini
    % hesapla
    [Ix,Iy] =
gradient(frame1);
    [ny, nx] = size(Ix);
    %-----
    % En yak n kareleri
    % kullanarak zamana g re
    %  r nt deki de i ikli i
    % hesapla
    It = frame2-frame1;
    %-----
    % Sayac  ba lat
    ct = 1;
```

```

%-----
%-----
%-----
% Birinci blokun x
aralıkları hesapla
x1 = 1;
x2 = x1+x_blk_size;

%-----
%-----
% X ve y yollarındaki
adamlar hesapla
xs = floor((nx-
x_blk_size)/x_blk_size);
ys = floor((ny-
y_blk_size)/y_blk_size);

%-----
%-----
% Hız vektörlerini
başlatır
% Vx, vy, hız
vektörlerinin x
bileşenlerini depolamak
için kullanılır ve
% Y-bileşenlerini
sırasıyla göstermektedir.
vx = zeros(1,ys*xs);
vy = zeros(1,ys*xs);

%-----
%-----
% Konum vektörlerini
başlatır
% X, y, optik hesaplamak
için kullanılan bloğun
merkez konumunu depolar
% Akış deşerleri.
x = zeros(1,ys*xs);
y = zeros(1,ys*xs);

%-----
%-----
% Resim boyunca dolaş
for ix = 1:xs

```

```

y1 = 1;
y2 = y1 + y_blk_size;

for iy = 1:ys

%-----
%-----
%
Gerçekleştirilmek için
degrade ve fark
görsel verilerden bir alt
etki alanı seçin
% Özerinde
hesaplama

Ix_block =
Ix(y1:y2,x1:x2);
Iy_block =
Iy(y1:y2,x1:x2);
It_block =
It(y1:y2,x1:x2);

%-----
%-----
% Problemi
doğrusal denklem olarak
dönüştürün ve bir lsqr anlamda
çözmek

% Bu yaklaşık
Lucas-Kanade Metodu olarak
bilinir

% A*u = f
% A'*A*u = A'*f
% u =
inv(A'*A)*A*f
% solve inv(A'*A)
using pseudo-inv (pinv)
% f -> It (Imge
zamanına göre deşimi)
% A -> [Ix,Iy]
(Pozisyona göre imaj
deşikliş)
% u -> [vx,vy]
(Hızlar, ne için çözmek
istediğimiz)

A = [Ix_block(:)
, Iy_block(:)];

```

```

b = -It_block(:);

A = A(1:1:end,:);
b = b(1:1:end);

P = pinv(A'*A);
u = P*A'*b;

%-----
%-----

% Mevcut alt
alana g receli h zlar.
%
% Not: bunu
ger sek bir h za ( r. [M /
s] yapard n
%  ter seveler
aras ndaki oran hakk nda
bilgi i sermesi gerekir
% (delta-t) Ve
i sindeki kom u pikseller
aras ndaki mesafe
% G r nt ler
(delta-x, delta-y). Aksi
takdirde sonu  bir
% G receli h z.
vx(ct) = u(1);
vy(ct) = u(2);

%-----
%-----

% Alt alan n
orta noktas n  hesapla
y(ct) =
(x1+x2)/2;
x(ct) =
(y1+y2)/2;

ct = ct+1;

%-----
%-----

% Yeni blo un y
aral    n  elde
y1 = y1 +
y_blk_size + 1;
y2 = y1 +
y_blk_size;

%-----
%-----

% Y y n ndeki
g r nt  boyutunu
a mad    n zdan emin olun
if y2 > ny
    y2 = ny;
end
end
%-----
%-----

% Yeni blo un x
aral    n  elde et
x1 = x1 + x_blk_size
+ 1;
x2 = x1 + x_blk_size;
%-----
%-----

% X y n nde
g r nt  boyutunu
a mad    n zdan emin olun
if x2 > nx
    x2 = nx;
end
end

position(iframe,[:,]) = [y
; x];
velocity(iframe,[:,]) =
[vx ; vy];
x = [];
y = [];
vx = [];
vy = [];
ct = 1;

end
%-----
%-----

% Girdi ba  ms z
de i keni yoksa, arsa
sonu lar 
if nargin == 0 ||
options.displayResults
figure

```

```

        for iframe =
1:size(velocity,1);
            subplot(1,1,1)

imagesc(squeeze(imgs(:,:,iframe+1))),colormap gray

title(sprintf('velocity
vectors + image\n frame
%g',iframe))
        hold on
            x =
squeeze(position(iframe,2,:))
;
            y =
squeeze(position(iframe,1,:))
;
            vx =
squeeze(velocity(iframe,2,:))
;
            vy =
squeeze(velocity(iframe,1,:))
;

plot(y,x,'.k','markersize',1)

quiver(y,x,vy,vx,'r'), hold
off

        drawnow
        pause(.1)
    end
end
output.position = position;
output.velocity = velocity;
output.num_blks_x = xs;
output.num_blks_y = ys;
output.blk_size_x =
x_blk_size;
output.blk_size_y =
y_blk_size;
output.nframes = nframe;
output.process_time =
sprintf('%3.2f
[sec]',toc(t));

```

## KAYNAKLAR

- Berbeco, R. I., Mostafavi, H., Sharp, G. C. and Jiang, S. B. 2005. "Towards fluoroscopic respiratory gating for lung tumours without radiopaque markers", *Phys. Med. Biol.* 50 4481–90.
- Bertalmio, M., Sapiro, G., and Randall, G., 2000, "Morphing active contours". *IEEE Trans. Patt. Anal. Mach. Intell.* 22, 7, 733–737.
- Bettencourt, K., Somers, D. C., 2007, "Effects of task difficulty on multiple object tracking performance", *Journal of Vision*, 7(9):898, 898a.
- Chang, W. Y., Chen, C.S., Jian, Y.D., 2008, "Visual Tracking in High-Dimensional State Space by Appearance-Guided Particle Filtering", 17(7):1154-67.
- Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intell.* 24, 5, 603-619.
- Comaniciu, D., Ramesh, V., and Meer, P. 2003. Kernel based object tracking. *IEEE Trans. Patt. Anal. Mach. Intell.* 25, 564-575.
- Cootes, T. F., Edwards, G. J., and Taylor, C. J. 1998. Active Appearance Model. *Proc European Conference on Computer Vision*, Vol. 2, pp. 484-498, Springer.