

**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**İŞARET DİLİNİN BİLGİSAYARLI YORUMLANMASI**

**YÜKSEK LİSANS TEZİ**

**Selda BAYRAK**

**Temmuz 2009**

**TRABZON**

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**İŞARET DİLİNİN BİLGİSAYARLI YORUMLANMASI**

**Bilgisayar Mühendisi Selda BAYRAK**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde  
“Bilgisayar Yüksek Mühendisi”  
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 12.06.2009  
Tezin Savunma Tarihi : 16.07.2009**

**Tez Danışmanı : Prof. Dr. Vasıf V. NABİYEV  
Jüri Üyesi : Yrd. Doç. Dr. Tuğrul ÇAVDAR  
Jüri Üyesi : Doç. Dr. Ali GANGAL**

**Enstitü Müdürü : Prof. Dr. Salih TERZİOĞLU**

**Trabzon 2009**

## ÖNSÖZ

“İşaret Dilinin Bilgisayarlı Yorumlanması” isimli bu çalışma; Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı’nda Yüksek Lisans Tezi olarak gerçekleştirilmiştir.

Böyle önemli bir konuya yönelmemi sağlayan, çalışmam boyunca ilgi ve alakasını hiçbir zaman eksik etmeyen, kıymetli bilgi ve tecrübelerinden yararlandığım, yönetici hocam Sayın Prof. Dr. Vasif V. NABIYEV’e minnet ve şükranlarımı sunmayı her zaman için bir borç bilirim.

Ayrıca, öğrenim hayatım süresince bana bilgi ve emeği geçen, tüm hocalarımı saygıyla anarım.

Öğrenim hayatımın her aşamasında maddi ve manevi desteklerini hiçbir zaman eksik etmeyen ve çalışmalarım sırasında gösterdikleri sabır ve şefkatlerinden dolayı aileme teşekkür eder, saygılarımı sunarım.

Selda BAYRAK  
Trabzon 2009

## İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET .....	V
SUMMARY .....	VI
ŞEKİLLER DİZİNİ.....	VII
TABLolar DİZİNİ.....	IX
SEMBOLLER DİZİNİ.....	X
1. GENEL BİLGİLER.....	1
1.1. Giriş .....	1
1.2. İşaret Dili .....	1
1.3. Türk İşaret Dili .....	4
1.4. İşitme Engelliler.....	5
1.5. İşaret Dili Tanıma Sistemleri.....	5
1.6. Markov Modeller .....	7
1.6.1. Saklı Markov Modeller.....	8
1.6.1.1. Saklı Markov Model Teorisi.....	9
1.6.1.2. Saklı Markov Modeldeki Üç Temel Problem.....	11
1.6.2. Saklı Markov Modeldeli Üç Temel Problemin Çözümleri.....	12
1.6.2.1. Değerlendirme Problemi.....	12
1.6.2.1.1. Forward Algoritması.....	14
1.6.2.1.2. Backward Algoritması .....	16
1.6.2.2. Durum Dizisinin Bulunması .....	17
1.6.2.2.1. Viterbi Algoritması .....	18
1.6.2.3. Model Parametrelerinin Öğrenilmesi .....	19
1.6.2.3.1. Baum-Welch Algoritması.....	20
1.6.3. Saklı Markov Model Çeşitleri .....	22
1.7. K-Ortalama Algoritmasıyla Vektör Kuantalama.....	23
1.7.1. K-Ortalama Algoritması .....	24

1.7.2.	K-Ortalama Algoritması Örneği .....	25
1.8.	Ortalama Kayma Algoritmasıyla Nesne Takibi .....	28
1.8.1	Ortalama Kayma Algoritması.....	28
1.9.	Zernike Momentlerle Özellik Çıkarma.....	30
2.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME .....	32
2.1.	Sistemin Genel Yapısı .....	32
2.2.	Nesne Takibi .....	33
2.3.	Özellik Çıkarma.....	37
2.4.	Vektör Kuantalama .....	42
2.5.	Saklı Markov Modelle Sınıflandırma .....	44
2.6.	Sistemin Test Edilmesi .....	54
3.	SONUÇLAR VE ÖNERİLER.....	57
4.	KAYNAKLAR .....	59
ÖZGEÇMİŞ		

## ÖZET

Bu tez çalışmasında işitme engellilerin kullandığı, işaret dilinin bilgisayarlı yorumlanması hedeflenmiştir. Bunun için, işaret dilinde kullanılan kelimelerinin temsil şeklini belirlemek ve bu şekli kullanarak sınıflandırma işlemlerinin yapılması gerekmektedir.

İlk olarak, verilerin elde edileceği videoda elin hareketinin takibi gerçekleştirilmiştir. Nesne takibi için ortalama kayma algoritması kullanılmıştır. Böylece zaman içinde elin şeklindeki değişim ve yörüngesi elde edilmektedir.

Bir sonraki aşamada elde edilen yörüngeden özellik vektörleri çıkarılmaktadır. El şeklinin özellik vektörlerini çıkarmak için, Zernike momentler hesaplanmaktadır.

Son olarak da çeşitli el işaretlerinin tanınması için saklı Markov model sınıflandırması önerilmiştir. Tanıma işleminde eğitilmiş sistemdeki işaretler arasındaki benzerliğe göre karar verilmektedir.

**Anahtar Kelimeler:** İşaret Dili, Ortalama Kayma Algoritması, Zernike Momentler, Saklı Markov Modeller

## SUMMARY

### Computer-Based Sign Language Interpretation

In this thesis, it is aimed to interpret sign language used by deaf people based on computer. For this, it is required to determine representations for words used in the sign language and perform classification by using these representations.

First of all, hand motion tracking in the video where data are extracted from was implemented. For object tracking, mean shift algorithm was used. Thus, change in hand shape in time and trajectory of hand were obtained.

In the next step, the feature vectors were extracted from the obtained trajectory. To extract hand shape feature vectors, Zernike moments were computed.

Finally, hidden Markov model classification was proposed for various types of hand gesture recognition. The decision regarding in recognition was made based on similarity among signs in the trained system

**Keywords:** Sign Language, Mean-Shift Algorithm, Zernike Moments, Hidden Markov Model

## ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1.1. Türk parmak alfabesi.....	3
Şekil 1.2. (a) parmak alfabesi; (b) tek işaret olarak ANTALYA kelimesinin gösterimi	3
Şekil 1.3. “Kitap aldım” cümlesinin işaret dili ile anlatımı .....	4
Şekil 1.4. Markov model.. .....	8
Şekil 1.5. Saklı Markov model .....	9
Şekil 1.6. İki durumlu Markov model .....	10
Şekil 1.7. Saklı Markov model .....	13
Şekil 1.8. Forward algoritmasının özyineleme adımı.....	15
Şekil 1.9. Backward algoritmasının özyineleme adımı .....	16
Şekil 1.10. Markov model çeşitleri.....	23
Şekil 1.11. K-ortalama algoritmasıyla iki sınıfa ayrılacak veriler kümesi.....	25
Şekil 1.12. Sınıf merkezlerinin seçilmesi, $m_1=(1,1)$ , $m_2=(2,1)$ .....	26
Şekil 1.13. K-ortalama algoritması sınıf merkezlerinin değişimi (a) .....	27
Şekil 1.14. K-ortalama algoritması sınıf merkezlerinin değişimi (b).....	28
Şekil 2.1. İşaretin yorumlanması için geliştirilen sistem.....	32
Şekil 2.2. İlk çerçevede takip edilecek nesnenin kullanıcı tarafından seçilmesi.....	34
Şekil 2.3. Seçilen nesnenin ortalama kayma algoritmasına göre ilerleyen çerçevelerde takibini gösteren iki video örneği.....	36
Şekil 2.4. Dokusuna göre özellikleri çıkarılabilecek nesnelere .....	37
Şekil 2.5. Şekil tabanlı özellik belirleme.....	38
Şekil 2.6. (p,q) derecelerine göre çıkarılan Zernike moment değerlerinden orjinal görüntünün yeniden elde edilmesi (10, 20, 30, 40).....	39
Şekil 2.7. Zernike momentlerin hesaplanması .....	40



Şekil 2.8. Takip edilen iki görüntünün herhangi bir çerçevedeki, altı farklı (p,q) değeriyle hesaplanmış moment değerleri.....	41
Şekil 2.9. Video görüntüsünde takip edilen nesnenin Zernike momentler yardımıyla çıkarılan özellik vektörü.....	42
Şekil 2.10. K-ortalama algoritmasının akış diyagramı.....	43
Şekil 2.11. Elde edilen özellik vektörünün K değeri üç olan ortalama algoritmasıyla kuantalanması.....	44
Şekil 2.12. Beş sayısının işaret dilindeki ifadesi.....	45
Şekil 2.13. Üç durumlu iki çıkışlı saklı Markov örneği.....	46
Şekil 2.14. Gözlem dizisinin olasılığının hesaplanması.....	48
Şekil 2.15. Gözlem dizisinin $\lambda$ modeline göre üretilme olasılığının bulunması.....	48
Şekil 2.16. İki durumlu üç çıkışlı saklı Markov model.....	49
Şekil 2.17. Viterbi Algoritması.....	51
Şekil 2.18. $\lambda$ modeline göre verilen gözlem dizisi, $O=\{W,S,C\}$ 'yi üreten saklı durum dizisinin bulunması.....	52
Şekil 2.19. Baum-Welch Algoritması.....	53
Şekil 2.20. Verilen $O=\{W,S,C\}$ gözlemi için 2 saklı durum içeren modelin eğitimi.....	53
Şekil 2.21. Eğitilmiş beş Markov modelin bulunduğu sistemde test işlemi.....	54
Şekil 2.22. İşaret dilindeki üç sayısının eğitimi için kullanılan örnek özellik vektörü.....	55
Şekil 2.23. İşaret dilindeki üç sayısının test için kullanılan örnek özellik vektörü.....	55
Şekil 2.24. İşaret dilinde (1, 2, 3, 4, 5) sayılarının Markov modelinin oluşturulduğu sistemde üç sayısının test işlem sonucu.....	56

## TABLÖLAR DİZİNİ

	<b><u>Sayfa No</u></b>
Tablo 1.1. Saklı Markov modelde gözlem olasılığının hesaplanması .....	14
Tablo 2.1. Verilen gözlem dizisinin olasılığının bulunması (a).....	47
Tablo 2.2. Verilen gözlem dizisinin olasılığının bulunması (b) .....	47
Tablo 2.3. Saklı durum dizisinin bulunması .....	49

## SEMBOLLER DİZİNİ

$q_t$	: Sistemin t anında bulunduğu durum
$S$	: Markov modelin durum uzayı
$V$	: Gözlem vektörünün alacağı değerler kümesi
$A$	: Durum geçiş olasılıkları matrisi
$B$	: Gözlem olasılıkları matrisi
$\pi$	: İlk durum olasılıkları vektörü
$\lambda$	: Saklı Markov model parametreleri
$O$	: Gözlem dizisi
$Q$	: Durum dizisi
$P(O, Q   \lambda)$	: Saklı durum dizisi Q için, gözlem dizisi O'nun olasılığı
$\alpha_t$	: İleri yön değişkeni
$\beta_t$	: Geri yön değişkeni
$\gamma_t(i)$	: $\lambda$ modelde t anında i durumda olma olasılığı
$\delta_t(j)$	: Tek bir yol boyunca ( ilk t gözlemini ele alan ve $S_i$ durumunda sonlanan ) en yüksek olasılığı
$\psi_t(i)$	: $\delta_t(j)$ 'yi maksimum yapacak değişkenin yolunu belirlemek için kullanılan dizi
$\xi_t(i, j)$	: t anında $S_i$ t+1 anında $S_j$ durumunda bulunma olasılığı
$X$	: d-boyutlu özellik vektöründen oluşan veri kümesi
$m_i$	: Sınıf merkezleri
$d$	: Öklit uzaklığı
$\hat{q}_u(y)$	: Takip edilecek nesnenin renk histogramı
$\hat{p}_u(y)$	: Hedef görüntüye ait y noktasındaki renk histogramı
$\hat{\rho}(y)$	: Bhattacharyya katsayısı
$\omega_i$	: Her bir nokta için histograma dayalı pixel ağırlıkları
$\hat{y}_1$	: Bir sonraki görüntüde nesnenin yeni konum
$A_{pq}$	: Zernike momenti
$V_{pq}$	: (p,q) dereceli Zernike fonksiyonu

- $\theta$  : Vektörle X eksenini arasındaki açıyı
- $\rho$  : Merkez noktasından (x,y) pikseline ait uzaklık
- $R_{pq}(\rho)$  : Radyal polinom

## **1. GENEL BİLGİLER**

### **1.1. Giriş**

İletişim, birimlerin düşünce ve görüşlerini içeren mesaj alışverişidir ve sadece insana özgü bir olay değildir. Birim kelimesi insanları, hayvanları ya da makineleri de kapsamaktadır. İletişimde jest ve mimikleri içeren reflekslerin yanı sıra sözlü dili de kapsayan karışık öğrenilmiş davranışlar da kullanılır. Davranışlar doğuştan gelen, gelip geçici ve öğrenilmiş olarak üçe ayrılır. Öğrenilmiş davranışlar, bireylerin toplum içinde doğduktan sonra, iç ve dış çevrelerinden gelen uyarıcılarla etkileşimi sonucu edindikleri kalıcı davranışlardır.

Günümüz teknolojisine, insana ait özelliklerin aktarıldığı görülmektedir. Böylece nesnelere bir aygıttan farklı olarak, bilinci olan bir varlık şekline dönüştürülmesi amaçlanmaktadır. Yapılan çalışmalar insan hayatını kolaylaştıracak birçok alanda kullanılabilir. Bu alanlardan biri de, iletişimde sorunları olan (doğuştan, hastalıkla, yaşlılıkla ya da bunların dışındaki sebeplerle) işitme, görme engelli kişilerin topluma kazandırılması için yapılan uygulamalar olabilir. Dünyada toplumun gelişimine katkıda bulunacak birçok yetenekli engelli vardır. İnsanın doğadaki en üstün varlık olduğu göz önüne alınırsa durumun ciddiyeti daha açıkça görülmektedir.

İnsana ait özelliklerin (düşünme, vücut hareketleri, mimikler...) modellendiği sistemler, engelli olan kişilerin iletişimlerinde kullanacakları davranışları öğrenmesinde yardımcı olabilir.

### **1.2. İşaret Dili**

İşaret dili, işitme engellilerin kendi aralarında iletişim kurarken, el hareketlerini ve yüz mimiklerini kullanarak oluşturdukları görsel bir dildir.

İşaret dillerinin bilimsel olarak belirlenmiş ana özellikleri ise şunlardır:

- İşaret dilleri de sözlü diller gibi bir gramer yapısına sahiptir. Sanılanın aksine sözlü dillerden daha basit bir yapıda değildir.
- Her işaret dilinin kendine özgü gramer kuralları vardır,

- ve her kavram için kullanılan işaretler de kullanıcılar arasında ortaktır. Bu özelliği ile işaret dili, konuşurken kullandığımız jestler ya da pandomimden çok farklıdır.

- Her ülkenin kendi işaret dili vardır. Örneğin, Amerika'da kullanılan işaret dili (ASL) ile Almanya'da kullanılan işaret dili birbirlerine benzemezler.

- Bir işaret dili çevrede kullanılan sözlü dilden etkilense de, farklı bir gramer yapısına sahiptir. Yani Türkçe ile Türk İşaret dili arasında mutlaka bir benzerlik olması gerekmez.

- Sözlü dillerde olduğu gibi işaret dili de erken yaşta öğrenilmelidir. İşitme engelliler 5-6 yaşına kadar işaret dili öğrenemezlerse daha sonra hem işaret dilini hem de başka dilleri öğrenmeleri zorlaşır.

- Beyin üzerindeki araştırmalara göre, sözel dillerle işaret dilleri aynı nörofizyolojik süreçlere ve aynı lokalizasyona (yani beynin sol yarımküresi) sahiptir (URL-1, 2009).

Sağırlar genellikle üç değişik iletişim yöntemi kullanmaktadır. Bu iletişim teknikleri daktilo veya parmak alfabesi, işaret dili ve dudak okumadır.

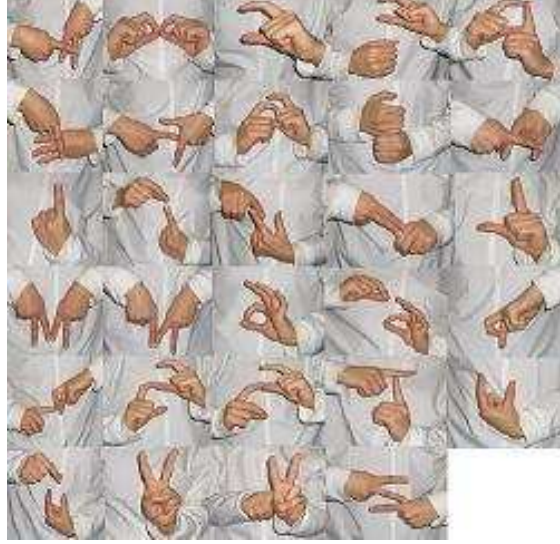
Daktilo alfabesi, 16. yy'da İspanyol filozoflar Carden, Pons, Bonet tarafından önerilmiştir. Daktilo alfabesinde her bir harf tek elle ya da iki elle kodlanarak kelimeler oluşturulur. Bu kelimelerden ise cümleler elde edilerek iletişim gerçekleştirilir. Parmak alfabesi genellikle,

- ✓ Özel isimlerin anlatımında,
- ✓ Yabancı dillerden alıntı kelimelerde,
- ✓ Bilimsel terimlerde,
- ✓ Kısaltmalarda,
- ✓ Eklerde ve
- ✓ Eş anlamlı işaretlerde kullanılır.

Dünyada 40'tan çok çeşitli parmak alfabesinin olduğu bilinmektedir. Bunlara örnek olarak Amerikan İşaret dili olan Ameslan (ASL-Ameslan- **American Sign Language**), İngiliz İşaret dili BSL (British Sign Language) veya Yetişkinler İçin Türkçe İşaret Dili gösterilebilir.

Genellikle parmak alfabesinin, dilin alfabe karakterleriyle arasında benzerlik bulunmaktadır. Türkçe'deki tüm 29 harfin parmak karşılığı vardır (Şekil 1.1). Fakat Rus veya Çince benzeri fonetik alfabelerde yalnız kısıtlı sayılarda harflerin parmak işaretleri

mevcuttur. 1963 yılından Uluslararası parmak alfabesi oluşturulmuştur. Şekil 1.2’de örnek olarak Antalya kelimesinin uluslararası parmak kodlanması görülmektedir.



Şekil 1.1. Türk parmak alfabesi



Şekil 1.2: (a) parmak alfabesi; (b) tek işaret olarak ANTALYA kelimesinin gösterimi

Parmak alfabesi, iletişim için kabul edilir araçlardan olsa da sağır için hiçbir alfabe baş harfleri, noktalama işaretlerini, boşluk veya matematiksel ifadelerini içermemektedir. İşaret dili ise kelimeler, mimikler ve jestler de kullanılarak bir bütünlük oluşturulmaktadır. Bu nedenle daktilo alfabesi harf tabanlı, işaret dili ise kelime tabanlı simgelenme olarak ele alınmaktadır. Dolayısıyla işaret dili tam bir kavrama uygun gelmektedir. Örneğin, Antalya kelimesi tek işaret olarak Şekil 1.2.(b)’de gösterilmiştir.

Sağırın iletişimde işaretler genellikle somut kavramlara dayalı olmaktadır. İşaretler de kavramlara bağlı olarak iki şekilde sınıflandırılır. Bunlardan birincisi ikonik (şekilsel) işaretler olup gerçeği ile belirli anlamda ilişkili olmaktadır. Örneğin “ev” işareti,

elin evin çatısını gösterecek şekilde ifade edilmektedir. Diğer serbest işaretlemelerde ise benzeri ilişki zayıftır ve kabullenmelere dayalıdır. Örneğin “akıllı” kavramı işaret parmağının ileriye doğru yarı bükülmüş olarak başa dayatılmasıyla ifade edilirken, “akılsız” elin yine başa dayalı olarak ters yöne çevrilmesiyle gösterilir.

Anlaşıldığı gibi bu dilin de kendine özgü grameri ve sözlükleri olmakta, ellerle beraber dudak, mimikler, gözler de önem taşımaktadır. Sağırların iletişimde dudak okuma ile iletilmesi gereken bilginin tanınması güçlendirilmiş oluyor.

İşaret dilinden sağırlar dışında, askeri tatbikatlarda, borsalarda veya bir orkestranın şef tarafından yönetiminde kullanılmaktadır. İşiten her insan da işaret dilini kullanmaktadır. Tek fark, bunun sağırlarda birincil iletişim aracı teşkil etmesidir.

### 1.3. Türk İşaret Dili

Türk İşaret Dili, tarihinin Osmanlı dönemine kadar uzanmasına karşın, bunun için Milli Eğitim Bakanlığı'nın 1995'te yayınladığı görsel bir kılavuz dışında henüz bir yazılı materyal, arşiv ya da sözlük yoktur.

Serbest olarak eksik, dolaylı yollardan ve oldukça geç yaşlarda öğrenilen bir dilin “işaret şiveleri” de dikkate alınırsa ihtiyaçlar doğrultusunda güncelleştirilmesi çok zordur. Tüm dünyada olduğu gibi Türkiye’de de çeşitli bölgelerde işaretler arasında farklılıklar olmasına rağmen işitme engellilerin birbiriyle daha kısa zaman içerisinde anlaşabildikleri görülüyor. Bu durum, işaret dilinin “sözcüklerinin” kısıtlı, “gramer” yapısının benzer ve göstererek öğrenmeye dayalı olmasından kaynaklanmaktadır. Örneğin, farklı bölgelerden olan işitme engelliler işaret diliyle “Kitabı aldım.” cümlesini kolaylıkla anlayabilmektedirler (Şekil 1.3).



Şekil 1.3. “Kitap aldım” cümlesinin işaret dili ile anlatımı®



#### **1.4. İşitme Engelliler**

Dünya Sağlık Örgütü'nün verilerine göre en gelişmiş ülkelerde engellilerin toplam nüfusa oranı %8'dir. Bu oran gelişmişlikle ters orantılı olarak artar. Buna göre ülkemizdeki engellilerin toplam nüfusa oranı %12'dir, yani 10 milyonun üzerindedir.

İşitme engellilerin sayısına dair çelişkili raporlar vardır. Birleşmiş Milletler raporuna göre bu sayı 2,5 milyondur. Ancak Milli Eğitim Bakanlığı 1998 Bütçe Raporu'na göre ülkemizde sadece 400,000 işitme engelli bulunmaktadır. Yine aynı rapora göre bu nüfusun 120,000'ini çocuklar oluşturmakta ve sadece 7,000'i okula gitmektedir.

İşitme engellilerin % 90'ının konuşan ailelere doğuyor olması ve henüz Türk İşaret Dili için bir eğitim materyali bulunmaması, söz konusu çocukların ilkokula gidene kadar herhangi bir dil öğrenmesini engellemektedir. Ayrıca, kullanıcıları tarafından eksik, dolaylı yollardan ve oldukça geç yaşlarda öğrenilen bir dilin geliştirilmesi, ihtiyaçlar doğrultusunda güncelleştirilmesi neredeyse imkansızdır Yapılan araştırmalara göre, öğrenme yaşı ilerledikçe, kişinin bir dili kullanma becerisi ve o dilin yapısına katkıda bulunma ihtimali azalır (Newport,1990).

Birçok ülkede olduğu gibi Türkiye'de de işitme engelliler için özel eğitim veren okullar bulunmaktadır. Ancak, burada da duyan öğretmenlerin çoğunlukla yüksek sesle Türkçe konuşarak eğitim verdiği bu okullarda, işaret dili eğitim sisteminin bir parçası değildir ve ülkemiz işaret dili eğitimi alanında pek çok ülkeden 50 yıl kadar geridedir.

Tüm bunlar değerlendirildiğinde, çalışmanın amaçlarından biri Türk İşaret Dili hakkındaki araştırmalara ve işitme engellilerin eğitiminde işaret dili kullanımının yaygınlaştırılmasına katkıda bulunmaktır. Ayrıca bu koşullarda işitme engelli çocuklar okuldan Türk İşaret Dili öğrenememektedir ve daha erken yaşlarda bilgisayarlı simülasyonlar eğitimde ve kişilerle iletişimde faydalı olabilir.

#### **1.5. İşaret Dili Tanıma Sistemleri**

İşaret dili tanıma sistemlerinde, genellikle verileri doğru şekilde elde etmek için sensörlü eldiven tabanlı yaklaşımlar kullanılmıştır. Fakat bu donanımsal tabanlı sistemler çok pahalı ve kullanıcı bağımlı olacağından, son yıllarda kamera tabanlı sistemler üzerine çalışmalar artmıştır. Kamera tabanlı sistemlerde ise, ardışık çerçevelerin işlenmesiyle veriler elde edilir. Burada da hareketi yorumlama kısımlarına daha çok yoğunlaşmak ve

görüntüde el bölgelerinin daha kolay tespiti için, kullanıcıların renkli eldiven giymesi istenmektedir. Ele giyilen herhangi bir belirteç kullanmadan (renkli veya çeşitli işaretler içeren eldivenler vb ), sadece kamera görüntüsünü kullanarak el, parmak pozisyonları ve yüz noktalarının bulunması henüz tam olarak çözülmemiş bir konudur.

Bilgisayarla görmede işaretlerin ve el hareketlerinin tanınması, uzun zamandan beri araştırmacıların ilgisini çekse de, ancak basit alt-problemler tanımlanıp bu problemlerde belli aşamalar kaydedilmiştir (Wu vd.,2001;Pavlovic vd.,1997). Bu alt problemlerden bazıları, belli işaretlerde kullanılan durağan el şekillerinin tanınması, el şeklinin değişiminden bağımsız olarak el izlencesinin tanımada kullanılması, baş ve yüz pozisyonunun bulunması ve belli sayıda yüz ifadesinin sınıflandırılmasıdır. Bu modalitelerin tümünün işaret dili tanımada kullanılmasına yönelik ancak sınırlı sayıda çalışma mevcuttur (Ong vd.,2005).

El hareketi tanıma sistemlerinde, kullanılan teknikler arasında yapay sinir ağları, saklı Markov modeller ve dinamik Bayes ağları bulunmaktadır. Çalışmalarda saklı Markov modellerin diğer tekniklere göre daha yüksek başarı gösterdiği görülmektedir.

Bauer vd. (2000), çalışmalarında kamera tabanlı, kişiden bağımsız, veri tabanında 97 kelime bulunan Alman işaret dili cümlelerini tanımaya çalışmışlardır. Amerikan işaret dilinin parmak alfabeti sistemlerine, Allen vd. (2003), ve Isaacs vd. (2004), 24 sabit karakterin tanındığı çalışmaları örnek olarak verilebilir.

Hem kamera hem de eldiven kullanılarak verilerin elde edildiği, Culver (2004) tarafından yapılan çalışma, Amerikan işaret dilinin 10-200 kelimesinin denendiği bir sistemdir. En doğru sonucun hem donanımsal hem de kamera tabanlı sistemin birleşiminden elde edildiğini göstermeye çalışmışlardır.

Starner vd.(1996), Amerikan işaret dilinden seçilmiş 40 işareti ayırık ya da belli kısıtları sağlayan cümleler içinde, bakış açısına bağımlı olarak tanıyabiliyorlarken, Wang vd.(2005), bakış açısından bağımsız olarak başarısı %97 olan, 100 kelimelik Çin işaret dili tanıma sistemini oluşturmaya çalışmışlardır. Ayrıca bu sistemin dudak okuma ve yürüyüş tanıma gibi sistemlerle genişletilebilecek özellikte olduğu belirtilmektedir.

Holden vd. (2005), çalışmalarında ise hem yüz hem de el nesnelerini izleyerek görüntü çerçevelerinden %97-99 başarı sağlayan, Avusturalya kelimelerinin işaret bilgilerini tanımaya çalışmışlardır.

Türkiye’de işaret dili konusunda yapılan araştırmalar, parmak alfabeti ve kelime tanımaya yönelik çalışmaları içermektedir. Parmak alfabeti tanıma sistemlerine, Gökna

vd. (2005) , Nabiyev vd. (2006) , kelime tabanlı olarak, Hakan vd. (2005) çalışmaları örnek olarak verilebilir.

Literatürde işaret dili tanınmasında yüz ifadeleri, baş ve vücut hareketlerinin de anlatılmak istenen kelime üzerinde önemli etkileri olduğu görülmektedir (Zeshan,2003). Fakat işaret tanıma literatüründe ele ait işaretler ve ele ait olmayan işaretleri beraber kullanarak işaret analizi yapan çalışma (Aran vd.,2006; Ong vd.,2005) sayısı az olduğu gibi işaret dillerinde kullanılan ele ait olmayan işaretler üzerinde yapılmış çalışma sayısı da çok değildir. Bazı çalışmalarda, sadece yüz ifadesi kullanılırken (Ming vd.,2002), bazılarında ise sadece baş hareketleri (Erdem vd.,2002) kullanılmaktadır.

Sensörlü eldiven tabanlı ve statik sistemlerde kullanılan örnek sayısı fazla iken kamera tabanlı sistemlerde 100-200 arasında değişmektedir.

Her ülkenin kendi işaret dili bulunduğu için tasarlanan sistemler ülkelere özgü olmaktadır.

## **1.6. Markov Modeller**

Verilen işaretin özelliklerini karakterize etmek için çeşitli modeller bulunmaktadır. Genel olarak işaret model çeşitlerini belirleyici (deterministik) ve istatistiksel model türleri olarak iki sınıfa ayırabiliriz. Deterministik modeller işaretin belirli özelliklerini kullanırlar. Örneğin, bu modellerde işaretin genlik, frekans, faz gibi özellikleri belirlenmektedir. İstatistiksel model sınıfında ise işaretin istatistiksel özellikleri karakterize edilmeye çalışılır. Sınıflar arası ilişkiyi tanımlamada kullanılan yöntemlerden, Gauss, Markov, saklı Markov modeller bu sınıfa örnek olarak verilebilir (Rabiner, 1989).

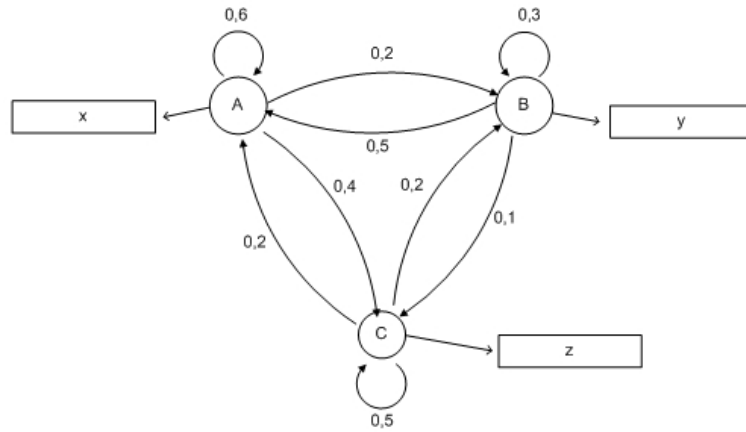
Markov modeller, istatistiksel özellikleri zamanla değişen dizilerin modellenmesinde kullanılır. Burada herhangi bir durumdaki özellik değeri hakkında, önceki durumların özellik değerlerine bağlı olarak çıkarımlar yapılabilir.

### 1.6.1. Saklı Markov Modeller

Saklı Markov modeller makine öğrenmesindeki başarılı yöntemlerdendir. Ses tanıma, şekil tanıma, el yazısı tanıma gibi çalışmalarda bu yöntem kullanılmıştır. Bu modelde durumlar doğrudan gözlenemediğinden, gözlem dizisinin en yüksek olasılıklı durum dizisi elde edilir.

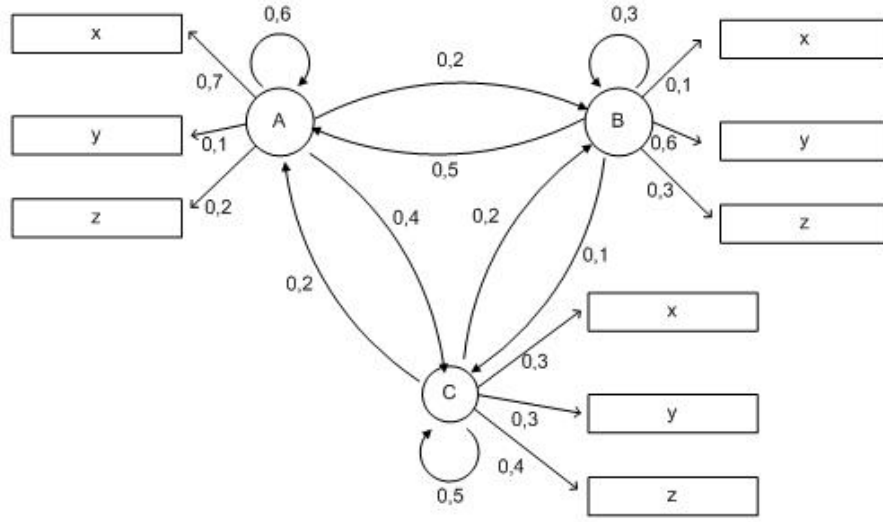
Şekil 1.4'te Markov model örneği verilmiştir. Bu modelde A, B, C olmak üzere üç durum ve bu modelin üreteceği X, Y, Z gözlemleri bulunmaktadır. Bu model, durumlar arasında olasılıklı geçişleri olan sonlu durum otomatıdır.

Bu süreçte gözlem dizisi olarak X, Y, Y verildiğinde, bu gözlemi üretecek durum dizisinin A, B, B olacağı açıkça görülmektedir. A, B, B durum dizisinin üretilme olasılığı da durum geçiş olasılıklarının çarpımı olacaktır;  $0.2 * 0.3 * 0.3 = 18 * 10^{-3}$ .



Şekil 1.4. Markov model

Şekil 1.5'te ise saklı Markov model örneği verilmiştir. Yeni modelde, tüm gözlem sembolleri tüm durumlar tarafından sonlu olasılıkla üretilmektedir. Bu değişiklik üretilecek olan gözlem dizisinin hangi durum dizisine karşılık geleceğini bulmayı zorlaştırır. Örneğin, X, Y, Y olarak verilen gözlem dizisi için oluşacak durum dizisinin ne olacağı kesin olarak söylenemez. Durum dizisi saklıdır. Burada gözlem dizisinin oluşturacağı maksimum olasılıklı durum dizisi belirlenir.



Şekil 1.5. Saklı Markov model

### 1.6.1.1. Saklı Markov Model Teorisi

Bir Markov modeli tanımlayan model parametreleri  $\lambda$  ile gösterilir.

$$\lambda = (\pi, A, B) \quad (1.1)$$

S, modelin durum uzayını gösterebilir. N, sistemin modellenildiği saklı durum sayısını ifade eder.

$$S = \{S_1, S_2, \dots, S_N\} \quad (1.2)$$

V, modelde gözlenebilecek durum uzayını gösterebilir. M, sistemdeki gözlem sembol sayısını ifade eder.

$$V = \{V_1, V_2, \dots, V_M\} \quad (1.3)$$

T uzunluğundaki durum dizisini Q, gözlem dizisini O sembolü ile ifade edelim.

$$Q = \{q_1, q_2, \dots, q_T\} \quad (1.4)$$

$$O = \{O_1, O_2, \dots, O_T\} \quad (1.5)$$

Sistemin  $t=0$  anında, durumların başlangıç olasılık vektörü  $\pi = \{ \pi_i \}$ ,

$$\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq N. \quad (1.6)$$

$t$  anında sistemin bulunduğu saklı durum,  $q_t$  ( $q_t \in S$ ) olsun. Durum geçiş olasılık matrisi  $A = \{a_{ij}\}$ ,

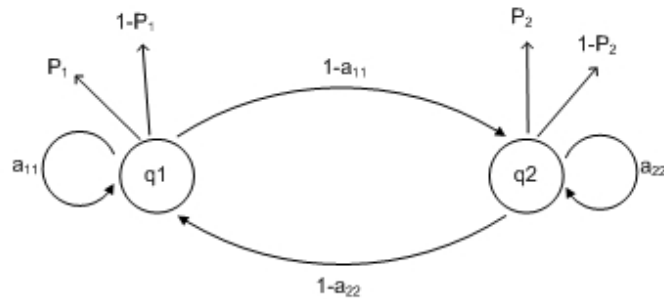
$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N. \quad (1.7)$$

Geçiş olasılıkları tüm  $i$ 'ler için,  $a_{ij} \geq 0$ ,  $\sum_{i=1}^N a_{ij} = 1$  olmasını sağlar.

Bir durumdaki gözlemlerin olasılık dağılımları  $B = \{b_j(V_k)\}$ ,

$$b_j(k) = P(V_k | q_t = S_j), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (1.8)$$

Şekil 1.6'da iki durumlu Markov model örneği verilmiştir.



Şekil 1.6. İki durumlu Markov model

Markov modelin teorisini Şekil 1.6'ya göre açıklarsak;

Durum uzayı :  $S = \{q_1, q_2\}$ ,  $N=2$ ,

Gözlem uzayı :  $V = \{Y, T\}$ ,  $M=2$ ,

Durum geçiş matrisi :  $A = \begin{pmatrix} a_{11} & 1-a_{11} \\ a_{22} & 1-a_{22} \end{pmatrix}$ ,

Gözlem olasılık matrisi :  $B = \begin{pmatrix} P_1 & 1-P_1 \\ P_2 & 1-P_2 \end{pmatrix}$  olmaktadır.

Şekil 1.6'daki  $P_1$ , 1. durumda ( $q_1$ ), oluşacak Y gözleminin olasılığını ve benzer şekilde de  $(1-P_1)$ , 1. durumdaki T gözlem olasılığını ifade etmektedir. Buradan da anlaşılacağı gibi gözlem sayısı, Y ve T olmak üzere iki tanedir.

### 1.6.1.2. Saklı Markov Modeldeki Üç Temel Problem

Markov yönteminin gerçek uygulamalarda kullanılabilmesi için üç temel problemin çözülmesi gerekir. Bu problemler aşağıda verilmektedir;

- Problem 1 :  $\lambda=(\pi, A, B)$  ile tanımlanan bir Markov model ve gözlem dizisi  $O = \{O_1, O_2, \dots, O_T\}$  verildiğinde,  $P(O|\lambda)$  gözlem dizisinin olasılığı nasıl hesaplanır?
- Problem 2 :  $\lambda=(\pi, A, B)$  ile tanımlanan bir Markov model ve gözlem dizisi  $O = \{O_1, O_2, \dots, O_T\}$  verildiğinde, ilgili en anlamlı saklı durum dizisi,  $Q = \{q_1, q_2, \dots, q_T\}$  nasıl bulunur?
- Problem 3 : Verilen  $O = \{O_1, O_2, \dots, O_T\}$  gözlem dizisinin olasılığını ( $P(O|\lambda)$ ) maksimum yapacak,  $\lambda=(\pi, A, B)$  model parametreleri nasıl belirlenir?

Birinci problemi, değerlendirme problemi olarak inceleyebiliriz. İkinci problem ise doğru durum dizisini bulmak için modelin saklı kısmının açığa çıkarmasıyla yani modelin

yapısının öğrenilmesiyle ilgilidir. Sonucu problemde de verilmiş gözlem dizisinin oluşumunu en iyi tanımlayacak model parametreleri belirlenmeye çalışılır. Model parametrelerini ayarlamak için kullanılan gözlem dizisi, saklı Markov modeli eğitmek için kullanıldığından eğitim dizisi olarak adlandırılır.

## 1.6.2. Saklı Markov Modeldeki Üç Temel Problemin Çözümleri

### 1.6.2.1. Değerlendirme Problemi

Değerlendirme probleminde  $O = \{O_1, O_2, \dots, O_T\}$  gözlem dizisinin olasılığı hesaplanır. En basit çözüm olarak T uzunluğundaki (gözlem dizisinin uzunluğu) tüm mümkün durumlar değerlendirilerek olasılık değeri hesaplanabilir.

Saklı durum dizisi Q için, gözlem dizisi O'nun olasılığı (gözlem dizisi istatistiksel bağımsız),

$$P(O|Q, \lambda) = \prod_{t=1}^T P(O_t | q_t, \lambda) \quad (1.9)$$

olur. Böylece (1.10)'daki denklemi elde ederiz.

$$P(O|Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T) \quad (1.10)$$

Öyle bir Q durum dizisinin olasılığı için (1.11)'i yazabiliriz.

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \quad (1.11)$$

O ve Q'nun bağlantılı olasılığı ise, yukarıdaki iki terimin çarpımıdır.

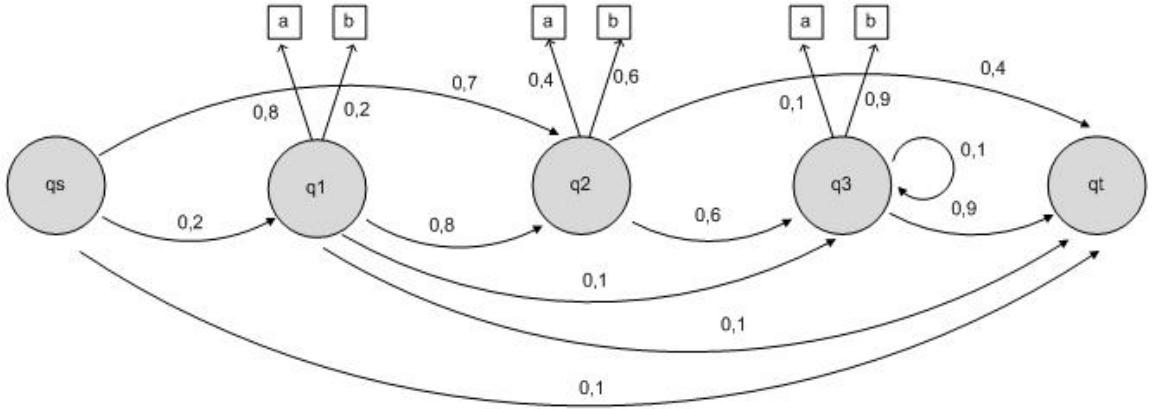
$$P(O, Q|\lambda) = P(O|Q, \lambda) P(Q|\lambda) \quad (1.12)$$



Gözlem dizisinin olasılığı, tüm mümkün durum dizileri üzerindeki bu bağlantılı olasılıkların toplanmasıyla elde edilir (1.13).

$$P(O, Q | \lambda) = P(O | Q, \lambda) P(Q | \lambda) \\ = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} \dots a_{q_{T-1} q_T} b_{q_T}(O_T) \quad (1.13)$$

Bu hesaplamayı yorumlarsak, sistem başlangıç durumunda ( $t=0$ ),  $\pi_{q_1}$  olasılıklı durumda bulunmaktadır ve  $b_{q_1}(O_1)$  olasılıklı  $O_1$  gözlemini üretmektedir. Daha sonra,  $t=1$  olduğunda sistem  $q_1$  durumundan  $q_2$  durumuna  $a_{q_1 q_2}$  olasılığıyla geçmektedir ve  $b_{q_2}(O_2)$  olasılıklı  $O_2$  gözlemini üretmektedir. Bu işlemler benzer şekilde, tüm geçişler gerçekleşene kadar ( $t=T$  olana kadar) devam edecektir.



Şekil 1.7. Saklı Markov model

Şekil 1.7'deki saklı Markov model örneğinde,  $O=\{a,a,a\}$  gözlem dizisinin üretilme olasılığını ( $P(O, Q | \lambda)$ ) (1.13)'deki ifadeye göre hesaplayalım.

Şekil 1.7'den başlangıç, geçiş ve gözlem olasılık matrisleri,

$$\pi = \begin{bmatrix} 0.2 \\ 0.7 \\ 0 \\ 0.1 \end{bmatrix} \quad A = \begin{bmatrix} 0.8 & 0.1 & 0.2 \\ 0 & 0.6 & 0.4 \\ 0 & 0.1 & 0.9 \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix} .$$

Tablo 1.1. Saklı Markov modelde gözlem olasılığının hesaplanması

Olası Q durum dizileri	Başlangıç olasılığı	Gözlem : $O_1$	Durum geçişi	Gözlem : $O_2$	Durum geçişi	Gözlem : $O_3$	Durum geçişi	$P(O, Q   \lambda)$
$q_1q_2q_3$	0.2	0.8	0.8	0.4	0.6	0.1	0.9	0.0027648
$q_1q_3q_3$	0.2	0.8	0.1	0.1	0.1	0.1	0.9	0.0000144
$q_2q_3q_3$	0.7	0.4	0.6	0.1	0.1	0.1	0.9	0.0001512

$$P(O, Q / \lambda) = 0.0027648 + 0.0000144 + 0.0001512 = 0.029304 \text{ olur.}$$

Bu hesaplama şekli çok fazla işlem yükü gerektirdiğinden burada daha etkili bir yöntem olan forward-backward algoritması kullanılır.

#### 1.6.2.1.1. Forward Algoritması

Her bir zaman adımında tüm durumlar için hesaplanan değerlerin elde tutulması hesaplama karmaşıklığını azaltacaktır. Bunun için bir ileri olasılık değişkeni  $\alpha_t$ ,

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda). \quad (1.14)$$

olarak tanımlanır. Bu değişken, t zamanına kadar gözlem dizisi  $O = \{O_1, O_2, \dots, O_T\}$ 'yi elde etme ve  $\lambda$  modelin t anında  $S_i$  durumunda bulunma olasılığını ifade eder.

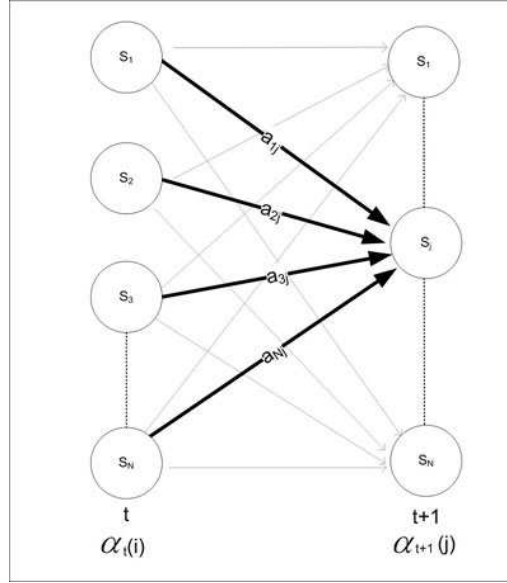
Forward algoritmasının işlem adımları aşağıda verilmiştir;

1. Başlatma:

$$\alpha_t(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (1.15)$$

2.Özyineleme:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N \quad (1.16)$$



Şekil 1.8. Forward algoritmasının özyineleme adımı

$\alpha_j(t)$ , her  $S_j$  durumu için  $t$  anına kadar oluşan gözlem dizisiyle birlikte o duruma geliş olasılığını içerir.

3.Sonlandırma:

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (1.17)$$

Backward algoritması da forward algoritmasının tersidir ve sondan başa doğru ilerler.

### 1.6.2.1.2. Backward Algoritması

Backward olasılık değişkeni  $\beta_t$ ,

$$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T | q_t = S_i, \lambda) \quad (1.18)$$

olarak tanımlanır.  $\beta_t(i)$  değişkeni, t anında  $S_i$  durumunda olma ve t+1'den T'ye kadar kısmi gözlem dizisinin,  $O = \{ O_{t+1}, \dots, O_T \}$ , olasılığını ifade eder.

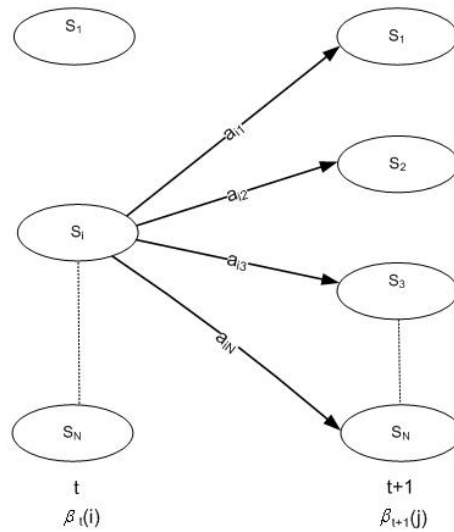
Backward algoritmasının işlem adımları aşağıda verilmiştir;

1. Başlatma:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (1.19)$$

2. Özyineleme:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N \quad (1.20)$$



Şekil 1.9. Backward algoritmasının özyineleme adımı

Forward algoritmasıyla birlikte backward algoritmasının, diğer problemlerin çözümlerinde nasıl kullanılacağı daha sonra gösterilecektir.

### 1.6.2.2. Durum Dizisinin Bulunması

Burada amaç,  $\lambda=(\pi, A, B)$  Markov modeli ve gözlem dizisi  $O = \{O_1, O_2, \dots, O_T\}$  verildiğinde en uygun saklı durum dizisini bulmaktır. Bu problemin çözümü için ilk olarak  $\gamma_t(i)$  değişkenini tanımlayalım.

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) \quad (1.21)$$

$\gamma_t(i)$ ,  $\lambda$  modelde t anında  $S_i$  durumunda bulunma olasılığını verir. Bu ifadeyi forward ( $\alpha_t(i)$ )-backward ( $\beta_t(i)$ ) değişkenleriyle yeniden düzenlersek,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)}. \quad (1.22)$$

$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i)\beta_t(i)$  normalizasyon faktöründen dolayı  $\sum_{i=1}^N \gamma_t(i) = 1$  olmaktadır.

$\gamma_t(i)$ 'yi kullanarak sistemin t anında bulunduğu en muhtemel durum olan  $q_t$ 'yi belirleyebiliriz.

$$q_t = \underset{1 \leq i \leq N}{\operatorname{arg\,max}} [\gamma_t(i)], \quad 1 \leq t \leq T \quad (1.23)$$

Bu ifade her t anında en muhtemel durumu seçerek, beklenen doğru durum sayısını artırmasına rağmen oluşan sonuç durum dizisinde sorunlar olabilir. Durum geçişlerinde sıfır olasılığı içeren (bazı i, j değerleri için  $a_{ij}=0$ ) bir modelde geçerli değilken optimum durum dizisi sayılabilir. Bundan dolayı durum geçiş olasılıklarını da değerlendiren, en iyi tek durum dizini bulan Viterbi Algoritması kullanılır.

### 1.6.2.2.1.Viterbi Algoritması

Verilen gözlem dizisine,  $O = \{O_1, O_2, \dots, O_T\}$ , karşılık gelen en iyi saklı durum dizisi,  $Q = \{q_1, q_2, \dots, q_T\}$ 'yi bulmak için tek bir yol boyunca (ilk t gözlemini ele alan ve  $S_i$  durumunda sonlanan) en yüksek olasılığı ifade eden  $\delta_t(j)$  değişkeni (1.24)'teki gibi tanımlanır.

$$\delta_t(j) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = 1, O_1, O_2, \dots, O_t | \lambda] \quad (1.24)$$

Bu ifadeden yararlanarak  $\delta_{t+1}(j)$ 'i aşağıdaki şekilde yazabiliriz,

$$\delta_{t+1}(j) = \left[ \max_i \delta_t(i) a_{ij} \right] b_j(O_{t+1}). \quad (1.25)$$

Viterbi algoritmasının işlem adımları aşağıda verilmiştir;

#### 1. Başlatma:

$\psi_1(i)$ , denklem (1.24)'ü maksimum yapacak değişkenin yolunu belirlemek için kullanılan dizidir.

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (1.26)$$

$$\psi_1(i) = 0 \quad (1.27)$$

#### 2. Özyineleme:

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[ \delta_{t-1}(i) a_{ij} \right] b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (1.28)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \quad (1.29)$$

3. Sonlandırma:

$$\rho^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (1.30)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (1.31)$$

4. Durum dizisi geriye doğru dönüş yapılır:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1 \quad (1.32)$$

Viterbi algoritması, toplam yerine maksimum olasılığa sahip durumun hesaba katılması haricinde forward algoritmasına benzerdir.

### 1.6.2.3. Model Parametrelerinin Öğrenilmesi

Verilmiş gözlem dizisinin modeldeki olasılığını maksimum yapacak,  $\lambda=(\pi, A, B)$  model parametrelerini belirlemede herhangi bir sonlu gözlem dizisi, sistemi eğitmek için kullanılır. Model parametrelerini belirlemede optimum bir yol yoktur. Fakat, Baum-Welch, EM (beklenti yükseltme), Gradyan teknikleri gibi iteratif yöntemler kullanılarak  $P(O|\lambda)$  olasılığını yerel olarak maksimum yapacak,  $\lambda=(\pi, A, B)$  model parametreleri seçilebilir.

### 1.6.2.3.1. Baum-Welch Algoritması

Saklı Markov model parametrelerini belirleme yöntemini açıklamadan önce gerekli olan,  $\xi_t(i, j)$  değişkenini tanımlayalım.

$$\xi_t(i, j) = P(q_{t+1} = S_i, q_{t+1} = S_j | \lambda) \quad (1.33)$$

$\xi_t(i, j)$ , t anında  $S_i$  t+1 anında  $S_j$  durumunda bulunma olasılığını ifade etmektedir.

Forward ve backward değişkenleriyle denklem (1.33)'ü yeniden düzenlersek,

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (1.34)$$

olur. Daha önce verilmiş gözlem dizisi için tanımladığımız, t anında  $S_i$  durumunda bulunma olasılığı  $\gamma_t(i)$  ile  $\xi_t(i, j)$  arasındaki bağıntı (1.35)'teki gibi yazılabilir.

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (1.35)$$

Eğer zaman parametresi süresince  $\gamma_t(i)$ 'nin toplamını hesaplırsak,  $S_i$  durumundan kaç kere geçildiğini bulmuş oluruz (1.36). Benzer şekilde  $\xi_t(i, j)$ 'nin toplamı da  $S_i$  durumundan  $S_j$  durumuna kaç kere geçildiğini gösterecektir (1.37).

$$\sum_{t=1}^{T-1} \gamma_t(i), S_i \text{ durumundan geçiş sayısı} \quad (1.36)$$



$$\sum_{t=1}^{T-1} \xi_t(i, j), S_i \text{ durumundan } S_j \text{ durumuna geçiş sayısı} \quad (1.37)$$

(1.36) ve (1.37)'deki denklemleri kullanarak saklı Markov model parametrelerinin,  $\lambda=(\pi, A, B)$  hesaplanması için gerekli denklemler,

$$\bar{\pi}_i = \gamma_1(i) \quad (1.38)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (1.39)$$

$$\bar{b}_j(m) = \frac{\sum_{t=1}^T \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \quad (1.40)$$

şeklinde elde edilebilir.

Bu değişkenlerin kullanıldığı, saklı Markov model parametrelerini belirleme yöntemlerinden biri olan Baum-Welch algoritmasının işlem adımları aşağıda verilmiştir;

1.  $\lambda=(\pi, A, B)$  model parametreleri başlatılır.
2. Model parametreleri kullanılarak  $\xi_t(i, j), \gamma_t(i)$  değerleri hesaplanır.
3. Hesaplanan  $\xi_t(i, j), \gamma_t(i)$  değerleri ile  $\lambda_{\text{yeni}} = (\pi, A, B)$  model parametreleri belirlenir.
4.  $P(O | \lambda_{\text{yeni}}) > P(O | \lambda)$  ise 2. adımdan itibaren işlemler tekrarlanır.

Eğitim için K tane elemandan oluşan bir eğitim seti  $X = \{O^k\}_{k=1}^K$ , kullanılacaksa model parametrelerini hesaplamak için (1.41, 1.42, 1.43)'deki bağıntılar kullanılmalıdır.

$$\bar{\pi}_i = \frac{\sum_{k=1}^K \gamma_1^k(i)}{K} \quad (1.41)$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \xi_t^k(i, j)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(i)} \quad (1.42)$$

$$\bar{b}_j(m) = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(j)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(j)} \quad (1.43)$$

Bu durumda Baum-Welch algoritmasında, eğitim şartı için kullanılan olasılık değeri hesaplamasında aşağıdaki ifadeyi geçerli olacaktır.

$$P(X/\lambda) = \prod_{k=1}^K P(O^k/\lambda) \quad (1.44)$$

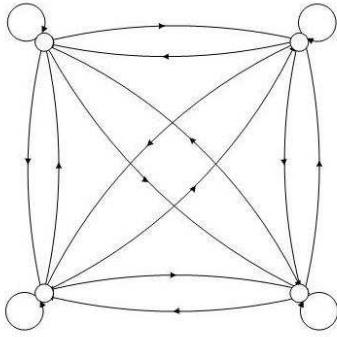
### 1.6.3. Saklı Markov Model Çeşitleri

Saklı Markov modelde durumlar arasında tüm geçişlere izin veriliyorsa, durum geçiş olasılık matrisi  $A = \{a_{ij}\}$ 'nin tüm değerleri pozitif, bu modele ergodik model denilmektedir (Şekil 1.10.(a)). Bir diğer model çeşidi de Şekil 1.10.(b)'deki soldan sağa saklı Markov'dur. Burada durum geçiş olasılık matrisinin değerlerinde,

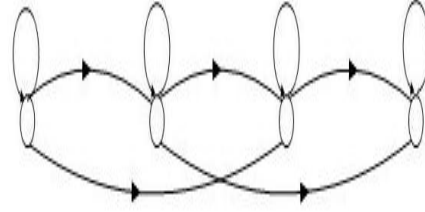
$$a_{ij} = 0, j > i + \blacktriangle \quad (1.45)$$

kısıtlaması vardır.  $\blacktriangle$  değişken değeri burada 2 olmaktadır. Ayrıca başlangıç olasılık matrisi de (1.46)'daki gibi olacaktır,

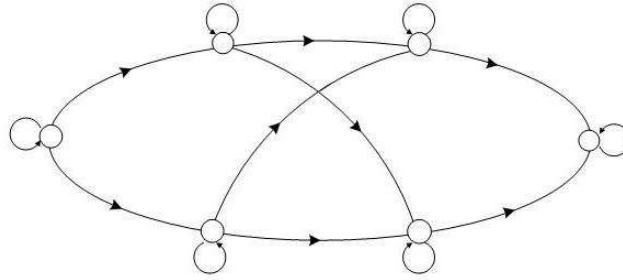
$$\pi = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (1.46)$$



(a) Ergodik model



(b) Soldan sağa Markov model



(c) Soldan sağa paralel Markov model

Şekil 1.10. Markov model çeşitleri

### 1.7. K-Ortalama Algoritmasıyla Vektör Kuantalama

Vektör kuantalama, giriş işaretinin kendisini en iyi şekilde temsil edecek olan ve daha az miktarda veri içeren ayrık uzaya yansıtmaktır. Her çıkış kod vektör; bu çıkışlar kümesi ise kod kitabı adlandırılmaktadır.

Şimdiye kadar anlatılan Markov örneklerindeki gözlem vektörü, sürekli değer almayan,  $V = \{V_1, V_2, \dots, V_M\}$  gibi ayrık yapıya sahip bir gözlem kümesinin elemanlarından oluştuğu varsayılmıştır. Eğer ayrık yapıya sahip bir Markov model tercih edilecekse ve eğitim için kullanılacak özellik vektörü sürekli değer alıyorsa, bu özellik vektörünün ayrık yapıya şekle dönüştürülmesi gerekmektedir.

Vektör kuantalama için kullanılacak yöntemlere, çarpım kod kuantalama, karar ağaçları kuantalama ve K-ortalama algoritmasıyla kuantalama örnek olarak verilebilir.

### 1.7.1. K-Ortalama Algoritması

Özellik vektörü (feature extraction) çıkarılmış bir grup verinin birden fazla sınıfa göre sınıflandırılmasıdır.

Her biri d-boyutlu özellik vektöründen oluşan  $X = \{X_1, X_2, \dots, X_N\}$  veri kümesi verilmiş olsun. K-ortalama algoritması, giriş parametresi olarak verilen bu veri kümesini K adet ( $K < N$ ) sınıfa  $S = \{S_1, S_2, \dots, S_K\}$  ayırmaktadır. Gerçekleştirilen sınıflama işlemi sonunda elde edilen kümelerin, küme içi benzerlikleri maksimum ve kümeler arası benzerliklerinin minimum olmaktadır. Sınıf benzerliği, sınıfın ağırlık merkezi olarak kabul edilen bir nesne ile sınıftaki diğer nesnelere arasındaki uzaklıkların ortalama değeri ile ölçülmektedir.

K-ortalama algoritmasının işlem adımları aşağıda verilmiştir;

1. Sınıf sayısını gösteren, K değerine karar verilir.

2. Sınıf merkezleri belirlenir. Sınıf merkezlerinin ( $m_1, m_2, \dots, m_K$ ) seçimi, nesnelere arasından K adet rastgele nokta seçilerek ya da merkez noktaların tüm nesnelere ortalama alınarak yapılabilir.

3. Her nesnenin seçilen merkez noktalara olan uzaklığı hesaplanır ve bu uzaklık değerine göre K adet sınıftan kendisine en yakın olan sınıfa yerleştirilir.

$$S_i^{(t)} = \{x_j : |x_j - m_i^{(t)}| \leq |x_j - m_{i^*}^{(t)}|, i^* = 1, \dots, k\} \quad (1.47)$$

4. Oluşan sınıfların merkez noktaları, o sınıftaki tüm nesnelerin ortalama değeri ile güncellenir.

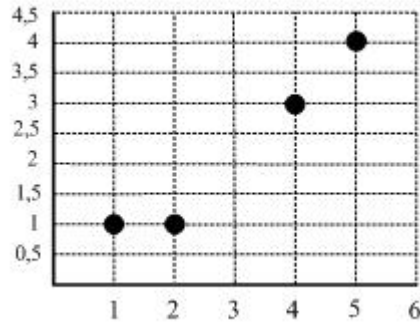
$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} X_j \quad (1.48)$$

5. Merkez nokta değişiyorsa 2. adıma gidilerek işlemler merkez nokta değişmeyene kadar tekrarlanır.

K-ortalama algoritmasının en büyük eksikliği, sınıf sayısı K değerinin önceden belirlenmesi gerektiğidir. Bu nedenle sınıflamanın başarısı K değerine göre değişebilmektedir. K-ortalama algoritmasının küresel verilerde, her zaman doğru sınıfları bulamadığı ancak sınıf sayısı doğru seçildiğinde ayrık şekildeki sınıfları etkili bir şekilde bulduğu söylenebilir.

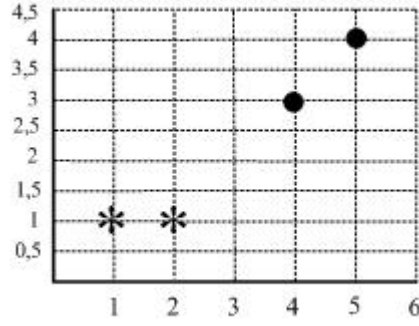
### 1.7.2. K-Ortalama Algoritma Örneği

K-ortalama algoritmasıyla iki sınıfa ayrılması gereken, Şekil 1.11'deki gibi dört tane nesneyi içeren veri kümesi verilmiş olsun. Görüldüğü gibi sınıflandırmada kullanılacak bu nesnelerin özellikleri koordinat eksenlerinin değerleri (iki boyutlu) olmaktadır.



Şekil 1.11. K-ortalama algoritmasıyla iki sınıfa ayrılacak veriler kümesi

1. İlk adım olarak iki sınıfa ayrılması gereken sınıf merkezleri seçilir (Şekil 1.12).



Şekil 1.12. Sınıf merkezlerinin seçilmesi,  $m_1=(1,1)$  ,  $m_2=(2,1)$

2.Öklid uzaklığına göre merkezlere olan uzaklıklar belirlenir (1.49).

$$d^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{matrix} m_1=(1,1) \\ m_2=(2,1) \end{matrix} \quad (1.49)$$

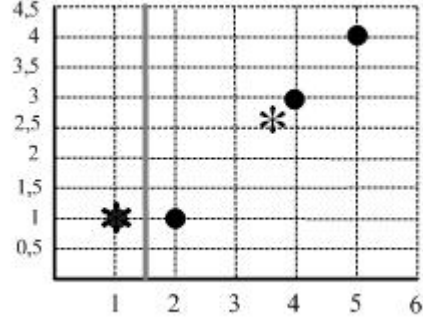
Örneğin (3,4) noktasında bulunan nesnenin  $m_1$  merkezine olan uzaklığı  $d = \sqrt{(3-1)^2 + (4-1)^2} = 3.61$  olmaktadır.

3. Hesaplanan bu uzaklık değerlerine göre verilerin hangi sınıfa ait oldukları belirlenir (sıfır ve bir değerleri iki farklı sınıfı temsil etmektedir).

$$S^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

4. Üçüncü adımda belirtildiği gibi ikinci sınıfta üç nesne bulunmaktadır. Yeni merkezlere bakılarak sınıf değişimine karar verilir. Eğer sınıf matrisi (S) bir önceki matrisle aynı ise sınıflandırma tamamlanmış olur.

$$m_1 = (1,1), \quad m_2 = \left( \frac{2+4+5}{3}, \frac{1+3+4}{3} \right) = \left( \frac{11}{3}, \frac{8}{3} \right)$$



Şekil 1.13. K-ortalama algoritması sınıf merkezlerinin değişimi (a)

5. Hesaplanan yeni sınıf merkezlerine göre uzaklıklar yeniden hesaplanır.

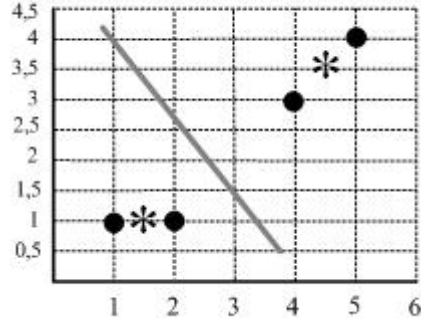
$$d^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} m_1=(1,1) \\ m_2=(\frac{11}{3}, \frac{8}{3}) \end{matrix}$$

6. Sınıf matrisinin yeni değerleri belirlenir.

$$S^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$S^1 \neq S^0$  olduğundan yeni sınıf merkezleri ve sınıf matrisleri hesaplanır.

$$m_1 = \left( \frac{1+2}{2}, \frac{1+1}{2} \right) = \left( \frac{3}{2}, 1 \right), \quad m_2 = \left( \frac{4+5}{2}, \frac{3+4}{2} \right) = \left( \frac{9}{2}, \frac{7}{2} \right)$$



Şekil 1.14. K-ortalama algoritması sınıf merkezlerinin değişimi (b)

$$d^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} m_1 = (\frac{3}{2}, 1) \\ m_2 = (\frac{9}{2}, \frac{7}{2}) \end{matrix} \quad S^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$S^1 \neq S^2$  olduğundan sınıflandırma tamamlanmış olur.

## 1.8. Ortalama Kayma Algoritmasıyla Nesne Takibi

Hareket algılama ve nesne takibi, görüntü işleme alanında kullanılan yöntemlerden biridir. Görüntüdeki hareketi ayırt ederek algılama ve algılanan hareketi takip etme yöntemi birçok farklı uygulamayla birlikte kullanılabilirdiği gibi tek başına kullanıldığı uygulamalar da vardır. Eğer gerçek zamanlı nesne takibi, başka uygulamalarla (diğer ön işleme adımları, nesne tanıma, hareketi yorumlama vb.) birlikte kullanılacaksa takip işleminin mümkün olduğu kadar hızlı olması gerekir.

### 1.8.1. Ortalama Kayma Algoritması

Bu yöntem, takip edilecek nesneden elde edilen (renk, doku vs) özelliklerden yararlanarak hedefin kısmi olarak kaybolması durumunda bile iyi bir sonuç üretmektedir. Burada amaç, takip edilecek nesnenin özelliklerine en yakın görüntü karesini bir sonraki görüntü çerçevesinde bulmaktır (Comaniciu vd.,2000).



Takip edilecek nesnenin m seviyeden (0-255) oluşan normalize edilmiş renk histogramını  $\hat{q}$  değişkeni ile gösterelim.

$$\hat{q} = \{\hat{q}_u\}_{u=1\dots m} \quad (1.44)$$

$$\sum_{u=1}^m \hat{q}_u = 1 \quad (1.45)$$

Hedef görüntüye ait m seviyeden oluşan y noktasındaki renk histogramı,

$$\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1\dots m} \quad (1.46)$$

$$\sum_{u=1}^m \hat{p}_u = 1 \quad (1.47)$$

olarak ifade edilsin. Bu iki histogram arasındaki benzerliği ölçmek için kullanılabilen ölçütlerden biri Bhattacharyya katsayısı olabilir (1.48).

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (1.48)$$

Normalize edilmiş histogramlar için bu katsayı 1 değerine ne kadar yakınsa iki görüntü arasındaki benzerlik o derece iyi olmaktadır.

Ortalama kayma algoritmasının işlem adımları aşağıda verilmiştir;

1. Şimdiki görüntüde,  $y_0$  noktasında bulunan nesne için  $\hat{p}(y_0)$  değeri hesaplanarak Bhattacharyya katsayısı,  $\hat{\rho}(y_0)$  bulunur.

2. Her bir nokta için histograma dayalı pixel ağırlıkları ifade eden,  $\omega_i$  değerleri (1.49)'daki denkleme göre hesaplanır.

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \quad (1.49)$$

3. Bir sonraki görüntüde nesne için yeni konum,  $y_1$  belirlenmeye çalışılır (1.50).  $X_i$  görüntüdeki normalize edilmiş pixellerin konumudur.

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} X_i \omega_i}{\sum_{i=1}^{n_h} \omega_i} \quad (1.50)$$

4. Eğer,  $|y_1 - y_0| <$  eşik değeri ise yeni konum belirlenmiştir. Fark değeri beklenen değerden (eşik değeri) büyükse,  $y_0 = y_1$  ile değiştirilerek 2.adımdan itibaren işlemler tekrarlanır. Burada Bhattacharyya katsayısı da kontrol değeri olarak kullanılabilir.

### 1.9. Zernike Momentlerle Özellik Çıkarma

Alt küme seçimi olarak da bilinen özellik seçme makine öğrenmesinde yaygın olarak kullanılan önemli bir ön işlemdir. En iyi alt küme seçimi, bilgiyi en az boyuttaki yani gereksiz bilgileri çıkarılmış vektörle ifade etmektir.

Momentler, bir nesnenin şeklini belirleyici olarak ifade eden karakteristikleri sunabilir. Görüntü tanıma ve sınıflandırmada nesnenin bağımsız global özellikleri için de 1960'lardan itibaren kullanılmaktadır (Liao vd.,2006 ; Heloise vd. 2004).

Zernike momentler, görüntü işleme ve bilgisayarlı görme tekniklerinde sıklıkla kullanılmaktadır. Bu momentler ölçekleme, öteleme ve rotasyon bağımsızlığından dolayı ilgi çekmektedir.

Zernike momenti  $A_{pq}$ ,

$$A_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x, y) [V_{pq}(x, y)]^*, \quad x^2 + y^2 \leq 1, \quad |q| \leq p \quad (1.51)$$

ifadesi ile tanımlanmaktadır. Denklem (1.51)'de bulunan (p,q) dereceli Zernike fonksiyonunun bağıntısı (1.54)'te verilmiştir (p-|q| her zaman çift olmalıdır).

$$V_{pq}(x, y) = R_{pq}(\rho) \exp(jq\theta), \quad x^2 + y^2 \leq 1 \quad (1.52)$$

Fonksiyonda bulunan,  $\rho$ , merkez noktasından (x,y) pikseline ait uzaklığı (1.53),  $\theta$ , vektörle X eksenini arasındaki açığı (1.54) belirtmektedir.

$$\rho = \sqrt{x^2 + y^2} \quad (1.53)$$

$$\theta = \arctan(y / x) \quad (1.54)$$

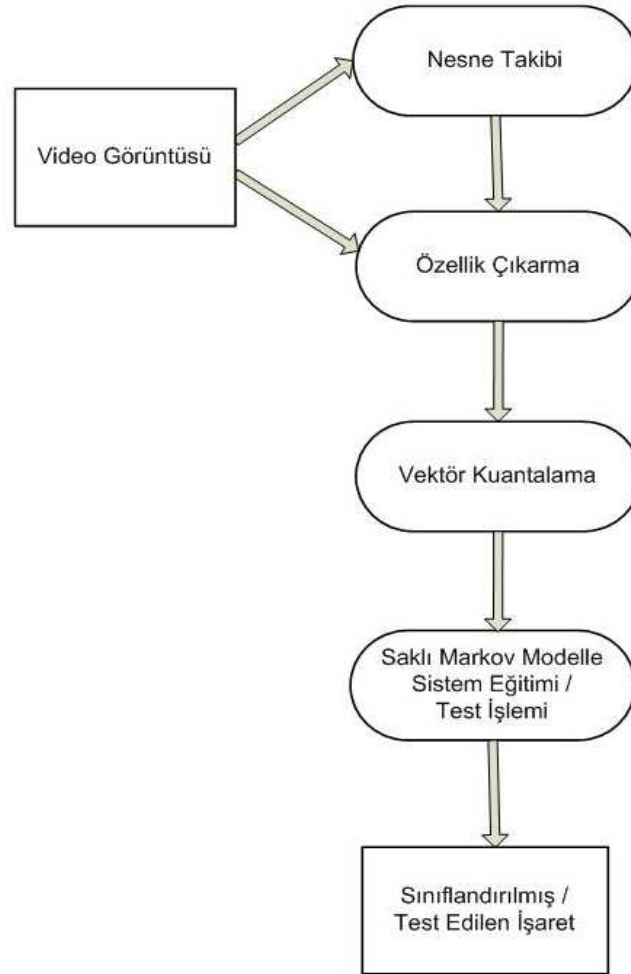
$$R_{pq}(\rho) = \sum_{l=0}^{(p-|q|)/2} (-1)^l \frac{(p-l)!}{l! \left(\frac{p+|q|}{2} - l\right)! \left(\frac{p-|q|}{2} - l\right)!} \rho^{p-2l} \quad (1.55)$$

(p,q) değişkenlerine bağlı olarak çıkarılan momentler özellik olarak kullanılabilir.

## 2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

### 2.1. Sistemin Genel Yapısı

Bu tez çalışmasında, Türk İşaret Dili kelimelerini içeren video görüntüsünden hareketlerin tanınması gerçekleştirilmiştir. İlk olarak görüntüde seçilmiş el hareketinin takibi, sonra da gerek sistemin eğitiminde gerekse test aşamasında kullanılacak özelliklerin çıkarımı için Zernike moment yöntemi kullanılmıştır. Hareketin yorumlanması aşamasında saklı Markov yöntemi kullanılmıştır. İşaretin yorumlanması için geliştirilen sistem, Şekil 2.1’de verilmiştir.



Şekil 2.1. İşaretin yorumlanması için geliştirilen sistem

## 2.2. Nesne Takibi

Nesne takibi için ilk olarak videoda takip edilecek nesnenin seçilmesi gerekir. Eğer nesne bilinen bir cisim ise eşleştirme yöntemiyle algılanabilir. Ya da özellikleri daha önceden bilinen bir cisim de olabilir. Örneğin yüz takibi yapılacaksa ten bilgisine dayalı analiz yapılabilir.

Nesnelerin farklı açılardaki görünümü, ortamdaki gece gündüz gibi ışık değişimlerinden nesnelerin etkilenmesi, gölge şekilleri vb. gibi sebeplerden dolayı nesnenin tespitinde ve takibinde sorunlar çıkabilmektedir.

Bu çalışmada nesne tespit kısmı gerçekleştirilmemiştir. Takip edilecek nesne ilk çerçevede kullanıcı tarafından seçilmektedir.

Nesne takip yöntemlerine, noktasal ve kernel takip yöntemleri örnek olarak verilebilir;

- Noktasal takip : Zamanla şekli değişmeyen ve tek noktayla temsil edilecek nesnelerin takibinde kullanılabilir. Daha büyük cisimler için çoklu nokta takibi tercih edilir. Noktaların aynı cisme ait olup olmama durumunun belirlenmesine dayalı bir yöntemdir.

- Kernel takip: Hesaplama maliyeti azdır ve geniş uygulama alanı vardır. Tek nesne takibi için model eşleştirme (template matching) ya da ortalama kayma (mean-shift) yöntemleri bulunmaktadır. Çoklu nesne takibinde durum tahmini için kalman filtreleri kullanılmaktadır.

Tez çalışmasında tek elle yapılan işaret dili kelimeleri incelendiğinden elin takibi için ortalama kayma algoritması kullanılmıştır. Bu yöntemde, takip edilen nesnenin bir sonraki çerçevede yerini bulmak için, ardışık iki çerçeveden elde edilen bölgelerin histogram benzerlikleri değerlendirilir ve hedefin kısmi olarak kaybolması durumunda bile iyi bir sonuç üretmektedir. Burada amaç, takip edilecek nesnenin özelliklerine en yakın görüntü karesini bir sonraki görüntü çerçevesinde bulmaktır.

Çalışmada oluşturulan sistem, mevcut videodan elde ettiği çerçeveleri kullanarak belirli bir büyüklükteki hareketli nesnenin güncel çerçevedeki konumunu belirlemektedir. Bunun için ilk olarak takip edilecek nesne ilk çerçevede kullanıcı tarafından belirlenir (Şekil 2.2).



Şekil 2.2 İlk çerçevede takip edilecek nesnenin kullanıcı tarafından seçilmesi

Bir görüntünün histogramı, herhangi bir renk değerinin görüntüdeki sıklığını belirtir. Histogram diyagramı yardımıyla görüntü hakkında birçok özellik çıkarılabilir ve görüntü üzerinde iyileştirmeler yapılabilir. Örneğin, koyu tonlu görüntünün renk değerleri diyagramın sol bölgesinde, açık tonlu görüntünün renk değerleri diyagramın sağ bölgesinde yığılma gösterir. Histogram yardımıyla, histogram eşleme yapılarak görüntüdeki zıtlık artırılıp daha net içerikli görüntüler elde edilebilir. Ayrıca gri görüntünün ikili görüntüye dönüştürülmesinde kullanılan, eşik değerinin seçimi de yapılabilir.

Görüntünün histogramının çıkarılması:

```

for i=y1:y2
  for j=x1:x2
    r=floor(g(i,j,1)/16)+1;
    g=floor(g(i,j,2)/16)+1;
    b=floor(g(i,j,3)/16)+1;
    hist(r,g,b)=hist(r,g,b)+1;
  end
end
end

```

Benzer şekilde bir sonraki çerçevede elde edilen görüntü için de renk histogramı çıkarılır ve her bir nokta için histograma dayalı pixel ağırlıklarını ifade eden,  $\omega_i$  değerleri hesaplanır.

Pixel ağırlık değerlerinin hesaplanması:

```

for i=y1:y2
  for j=x1:x2
    r=floor(g(i,j,1)/16)+1;
    g=floor(g(i,j,2)/16)+1;
    b=floor(g(i,j,3)/16)+1;
    u=(r-1)*256+(g-1)*16+b;
    x(1,t)=i; x(2,t)=j;
    w_i(t)=sqrt(q_u(u)/p_u(u));
    t=t+1;
  end
end
end

```

Elde edilen bu piksel ağırlık değerlerine göre görüntünün yeni konumu ( $y_1$ ) belirlenmeye çalışılır. Eğer,  $|y_1 - y_0| < \text{eşik değeri}$  ise, takip edilen nesnenin yeni konumu bir sonraki çerçevede belirlenmiştir.

Bir sonraki çerçevedeki görüntünün yeni konumunun belirlenmesi:

```

y1=x*w_i/sum(w_i);
if sqrt((y1(1)-y0(1))^2+(y1(2)-y0(2))^2)<eşik değeri
  break
end

```



Şekil 2.3. Seçilen nesnenin ortalama kayma algoritmasına göre ilerleyen çerçevelerde takibini gösteren iki video örneği



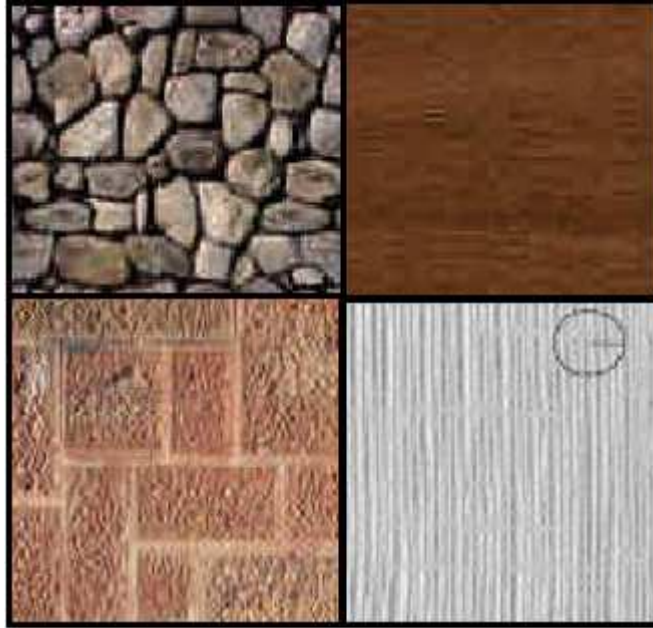
### 2.3. Özellik Çıkarma

Bir algoritma için giriş verisinin boyutunu indirgemenin özel bir şeklidir. En iyi özellik seçimi, bilgiyi en iyi şekilde ifade edecek, fazlalıkların çıkarıldığı yeterli bilgiyi içermelidir. Bir nesnenin genel özellikleri, renk, doku, şekil vb.'dir. Domain özel özelliklere ise parmak izi, insan yüzleri örnek olarak verilebilir.

Özellik çıkarma işlemlerdeki hesaplama yükünü azaltacaktır. Öznellikten dolayı özellik vektörünü en iyi temsil edecek tek bir yöntemden bahsedilememektedir.

Özellik çıkarma yöntemlerine;

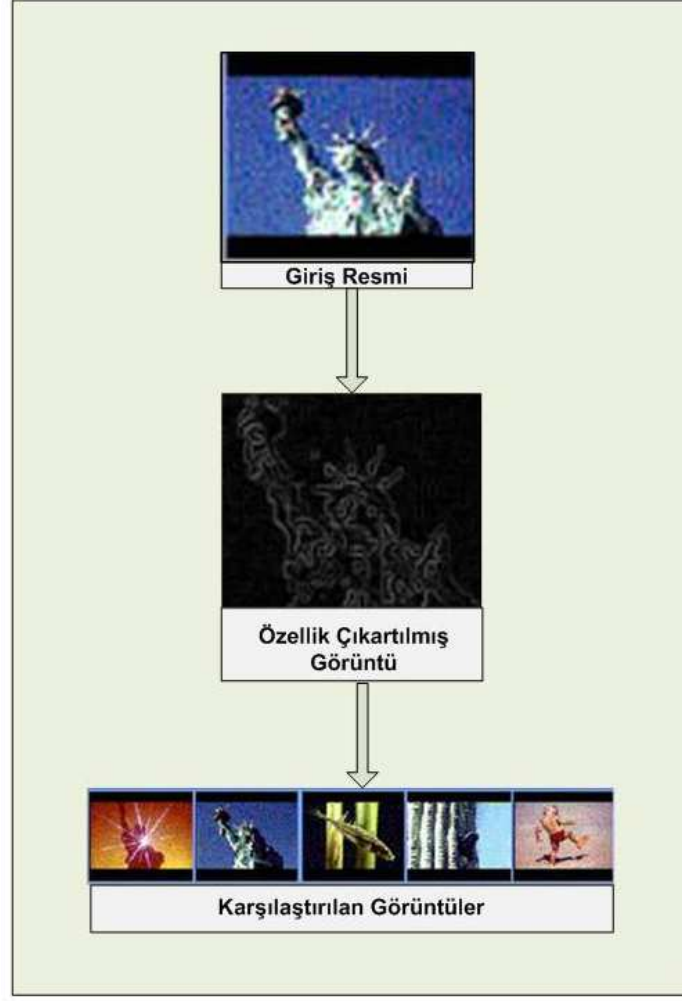
- ✓ Renk uzaylarında nesnenin histogram dağılımına göre özellik belirleme,
- ✓ Nesnenin dokusuna göre özellik belirleme :pürüzsüz-dalgalı yüzey, yatay-dikey çizgiler içeren yapılar, homojenlik, enerji, zıtlık gibi özellikler,



Şekil 2.4. Dokusuna göre özellikleri çıkarılabilecek nesneler

✓ Şekil tabanlı özellik belirleme :

- Global özellikler : Momentlere dayalı özellikler, yuvarlaklık özelliği vs.
- Lokal özellikler : Sınır segmentasyonu, örnek olarak verilebilir.

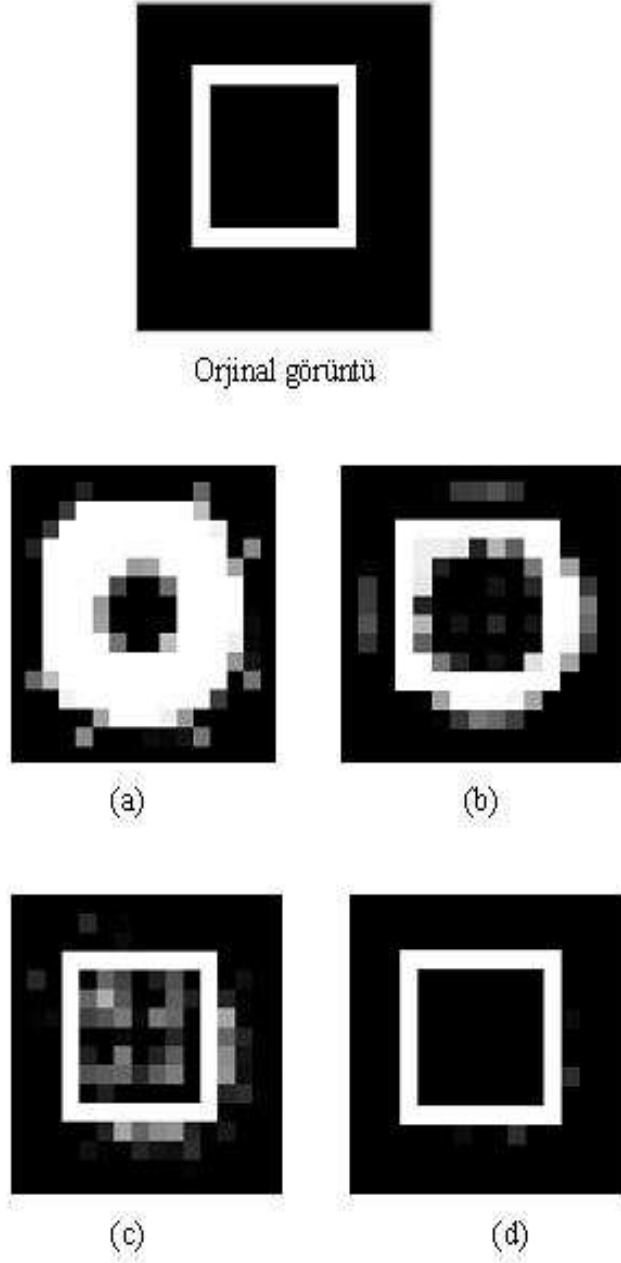


Şekil 2.5. Şekil tabanlı özellik belirleme

Şekil tabanlı global özelliklerden olan momentlere dayalı özellik çıkarma yöntemi, görüntü işleme ve bilgisayarlı görme tekniklerinde sıklıkla kullanılmaktadır. Ölçekleme, öteleme ve rotasyon bağımsızlıklarından dolayı ilgi çekmektedirler.

Momentler, uzun zamandır istatistik teorisinde ve klasik mekanikte kullanılmaktadırlar. İstatistikçiler moment olarak ortalama, varyans, çarpıklık oranı ve kurtosis kullanırlarken, mekanikçiler momentleri ağırlık merkezi bulmada ve atalet momenti hesaplamada kullanırlar. 1960'ların başında Hu, matematiksel moment teorisinden yedi değişmez moment geliştirdi. Bu yedi moment, ölçeklendirme, döndürme ve dönüşüm işlemleri altında değişmemekte ve karakter tanıma gibi iki boyutlu örüntü tanıma problemlerinin çözümünde gözle görülür başarı sağlamaktadır. Böylece karakter tanıma, nesne tanıma, radar görüntülerinden uçak ve gemi tanıma gibi uygulamalar ortaya çıkmıştır.

Bu çalışmada momentlerin bir çeşidi olan, Zernike momentler kullanılmıştır. Bu momentler ölçeklemeden, ötelemeden ve rotasyondan bağımsızdır. Hesaplanan moment değerlerinden yararlanarak özellikleri çıkarılan nesnenin yeniden elde edilmesi sağlanabilir (Şekil 2.6).



Şekil 2.6. (p,q) derecelerine göre çıkarılan Zernike moment değerlerinden orjinal görüntünün yeniden elde edilmesi (10, 20, 30, 40)



Şekil 2.7. Zernike momentlerin hesaplanması

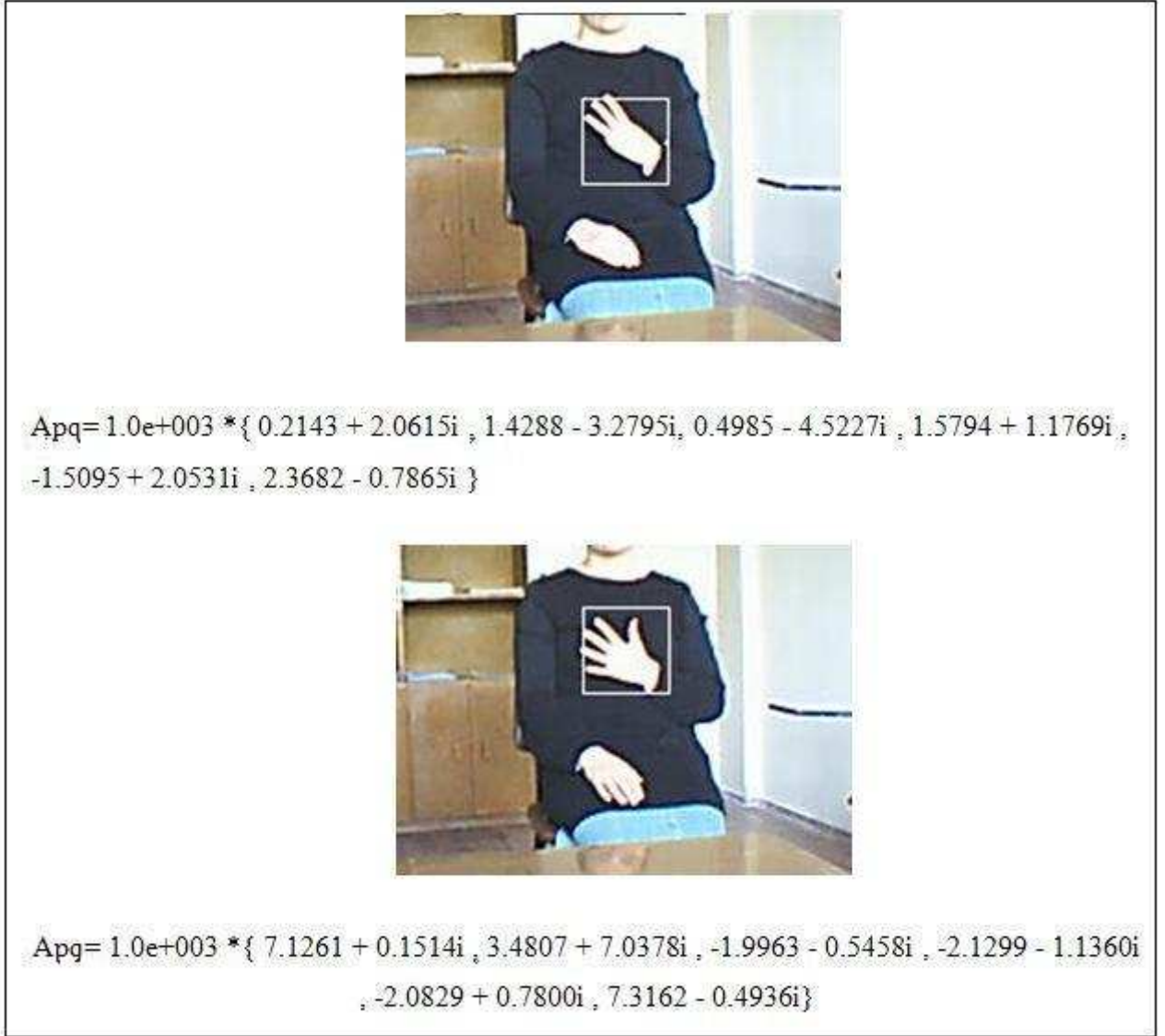
Bir nesneye ait (p,q) dereceli Zernike momentinin hesaplanması:

```

for s=0: (p-abs(q))/2
    d1 = ((-1)^s)*fac(p-s) * (rho.^(p-2*s));
    d2 = fac(s) * fac((p+abs(q))/2-s) * fac((p-abs(q))/2-s);
    R_pq = R_pq + d1/d2;
end
V_pq = R_pq * exp(j * q * theta);
G = image * conj(V_pq);
A_pq=(p+1) * sum (sum (G))/pi;
  
```

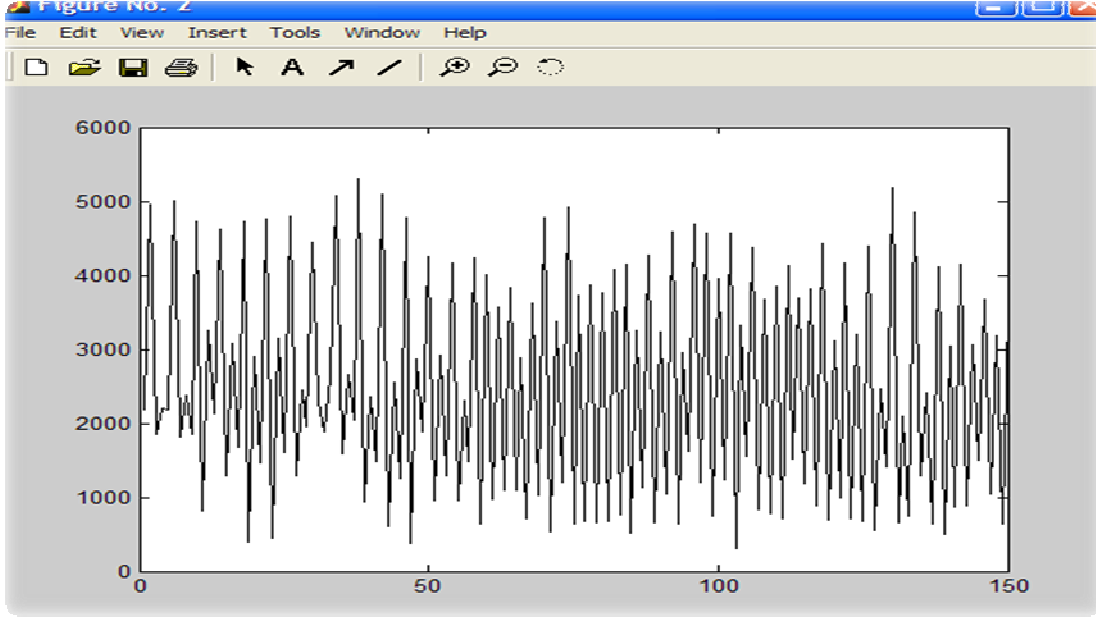
Şekil 2.8’de takip edilen iki görüntünün herhangi bir çerçevedeki, altı farklı dereceli momentinin değerleri gösterilmiştir. Moment derecesi (p,q) değerleri,

(p,q) = {(10,2), (10,4), (12,2), (12,4), (14,2), (14,2)} şeklinde seçilmiştir.



Şekil 2.8. Takip edilen iki görüntünün herhangi bir çerçevedeki, altı farklı (p,q) değeriyle hesaplanmış moment değerleri

Bu moment değerleri tüm video boyunca var olan çerçeveler için ya da sadece takip edilen nesnenin şekilsel değişiminin gerçekleştiği çerçeveler için hesaplanarak özellik vektörü oluşturulmaya çalışılır (Şekil 2.9).



Şekil 2.9. Video görüntüsünde takip edilen nesnenin Zernike momentler yardımıyla çıkarılan özellik vektörü

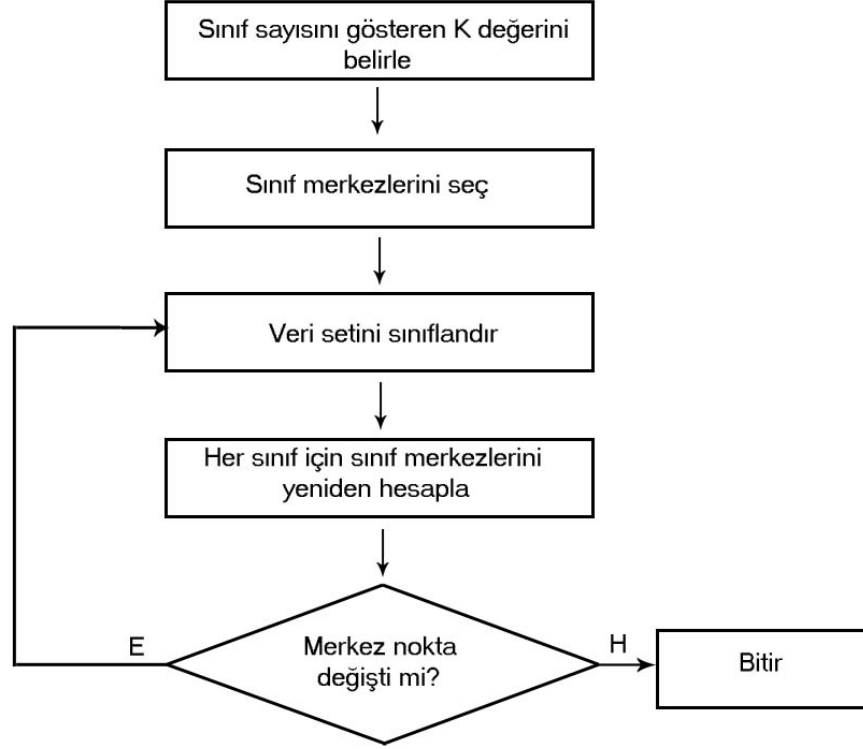
#### 2.4. Vektör Kuantalama

Dijital olarak bir işareti işlerken, daima en uygun hangi şekilde saklanabileceği ve doğru şekilde nasıl temsil edilebileceği karşılaşılan sorunlar arasındadır. İşaretin dijital olarak ifade edilmesi sonlu sayıda eleman olmasını gerektirir. Vektör kuantalama olarak adlandırılan bu işlem, giriş data uzayındaki vektörleri, işareti yeniden elde etmek için kullanılacak vektörlerin sonlu kümesine haritalamaktır (denk düşürmektir). İdealde bu işlem esnasında bilgi kaybının olmaması istenir. Bu yüzden işaretin içindeki gereksiz verileri elimine ederek bilgi saklanmaya çalışılır. Düşük bant genişliği olan bir telefon hattında konuşma işaretinin iletimi böyle bir kodlamaya örnek olarak verilebilir.

Vektör kuantalama için çarpım kod kuantalama, karar ağaçları kuantalama ve sınıflama algoritması olan, K-ortalama algoritmasıyla kuantalama kullanılabilir.

Sınıflandırma algoritmaları giriş örnekleri boyunca benzer veri noktalarının gruplarını bulmak için kullanılır. K-ortalama algoritması eğitim setinden verilen sayı kadar (K) sınıf elde etmede kullanılan etkili bir algoritmadır. Başlangıç olarak sınıf merkezleri rastgele seçilir. Seçilen ilk merkezler için veri seti sınıflandırılır. Ancak başlangıçta sınıf merkezlerini rastgele seçtiğimiz için ilk iterasyonda istenilen başarı büyük ihtimalle sağlanmayacaktır. Bu yüzden elde ettiğimiz kümelerin ortalamasını alarak yeni sınıf merkezlerini hesaplarız ve elde

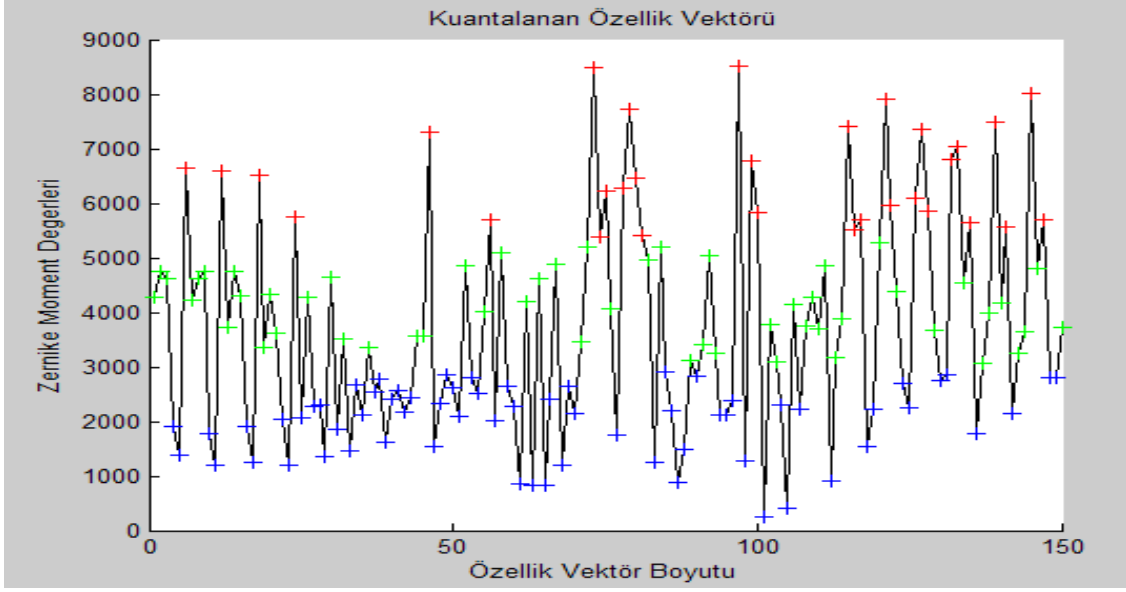
ettiğimiz yeni merkezlere göre verimizi yeniden sınıflandırırız. Bu işlem sürekli yapıldığında her iterasyonda sonuç daha da iyileşecektir.



Şekil 2.10. K-ortalama algoritmasının akış diyagramı

Burada dikkat edilmesi gereken bir durum ise, başlangıçta var olan rastgelelikten dolayı aynı sınıf içinde birden fazla birbirine yakın temsilci seçilebilme ihtimalidir. O yüzden sınıflama sırasında ayrıca sınıfların birbirlerine olan yakınlığının test edilmesi gerekebilir.

Tez çalışmasında sistemin özellik vektörünün çıkarılması aşamasında kullanılan Zernike moment yöntemiyle elde edilen sürekli değerlerden oluşan özellik vektörünün, ayrık yapılı şekle dönüştürülmesi gerekmektedir. Bu amaçla K-ortalama algoritması özellik vektörünü kuantalamak için kullanılmıştır.



Şekil 2.11. Elde edilen özellik vektörünün K değeri üç olan ortalama algoritmasıyla kuantalanması

Şekil 2.11'deki özellik vektörü üç sınıfa (kırmızı, yeşil, mavi renkle gösterilmiş etiketler) ayrılmıştır. Özellik vektörünü daha iyi temsil etmek için farklı K değerine göre kuantalama yapıp en iyi sonuç seçilebilir.

## 2.5. Saklı Markov Modelle Sınıflandırma

Sınıflandırma; birçok bilim dalında kullanılan bir karar verme işlemidir. Görüntü sınıflandırma işleminde amaç, bir görüntüdeki bütün pikselleri karşılık gelen sınıf içine otomatik olarak yerleştirmektir. Dinamik bir hareketin modellenmesi için saklı Markov modeller, zaman gecikmeli sinir ağları, dinamik Bayes ağları, sonlu durum makineleri gibi zamansal teknikler kullanılabilir.

Saklı Markov modeller, Markov zincirleri olarak da bilinir ve olasılık kuramının önemli kavramlarından biridir. Saklı Markov modeller, stokastik süreçleri modelleyebilen sonlu durum ağlarıdır ve bir işaretin stokastik olarak modellenmesidir. Modelde her durum, gözlem vektörü uzayına farklı bir bölge ve karakteristiği tanımlar. Durumlar arası geçişler de modellenen öznitelik değişimlerini ele alır. Durumlar içerisinde verilen herhangi bir öznitelik gözlem vektörünün duruma uygunluğunu veren çıktı olasılık dağılım fonksiyonları yer alır.

Bu yaklaşım ilk olarak 1965-70 yıllarında kullanılmaya başlanmış ve 1985 -90 yıllarında sesli ifade tanımda çok kullanılan bir yöntem olmuştur. Sesli ifade tanımda



kullanılması kısaca, ardışık kısa süreli sesli ifade kesimlerinin birlikte ele alınması ile ardı ardına gelebilecek bu kesimler için bir model oluşturmak ve bu modelden yararlanarak uzun süreli sesli ifade kesimlerinin tanınmasını sağlamak şeklinde özetlenebilir.

Bu çalışmada, işaret dilindeki bir kelimenin ifade edilmesinde hareket esnasında zamanla elin şeklindeki değişim göz önüne alınarak dinamik bir hareketin modellenmesi gerçekleştirilmiştir (Şekil 2.12).



Şekil 2.12. Beş sayısının işaret dilindeki ifadesi

Markov süreci ya da zinciri kesikli bir  $t$  zamanındaki  $N$  durumdan biridir. Genelde sistemin tam bir tanımı önceki durumların sırasının bilinmesine bağlıdır. Birinci dereceden bir Markov sürecinde sistemin şu andaki durumu sadece bir önceki duruma bağlıdır. Durum geçiş olasılıkları zamandan bağımsız olduğundan dolayı, bir Markov süreci durum geçişleri ile tanımlanabilir.

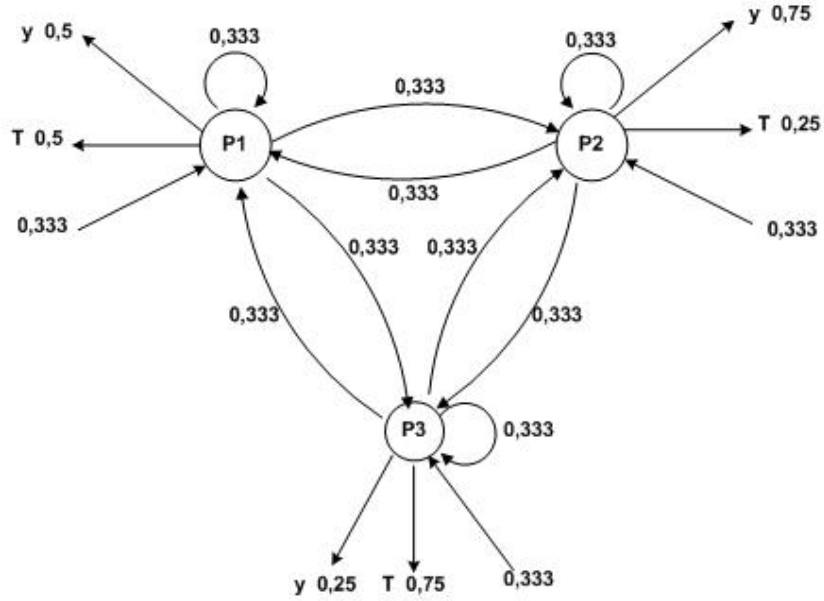
Bir işaretin saklı Markov yöntemiyle modellenmesi üç alt probleme ayrılabilir;

✓ Değerlendirme problemi; verilen bir model için belirli bir gözlem dizisinin olasılığının hesaplanması. Test edilecek bir örneğin tanıma aşamasında değerlendirilmesi için kullanılacaktır.

✓ Durum dizisinin bulunması problemi; verilen bir model için belirli bir gözlem dizisini oluşturan durum dizisinin bulunması.

✓ Model parametrelerinin öğrenilmesi problemi; verilen bir model için bir dizi gözlemi oluşturan bu modelin olasılığının yüksek olması için model parametrelerinin belirlenmesi. Sistemin bir eğitim seti ile birlikte eğitilmesi.

Birer örnekle bu problemleri inceleyelim.



Şekil 2.13. Üç durumlu iki çıkışlı saklı Markov model

Şekil 2.13'teki örnek için  $\lambda=(\pi, A, B)$  modelini yazalım,

$$\pi = \begin{bmatrix} 0,333 \\ 0,333 \\ 0,333 \end{bmatrix} \quad A = \begin{bmatrix} 0,333 & 0,333 & 0,333 \\ 0,333 & 0,333 & 0,333 \\ 0,333 & 0,333 & 0,333 \end{bmatrix} \quad B = \begin{bmatrix} 0,5 & 0,5 \\ 0,75 & 0,25 \\ 0,25 & 0,75 \end{bmatrix}$$

Değerlendirme problemi: Bu modelde  $V=\{Y,T\}$  gözlem kümesi için verilen  $O=\{Y,Y,T\}$  gözlem dizisinin olasılığını bulalım.

Tablo 2.1. Verilen gözlem dizisinin olasılığının bulunması (a)

Durum	Gözlem dizisi $O=\{ Y,Y,T \}$ olasılıkları				
	İlk Y gözlemi	P1 durumuna geçiş olasılıkları	P1 durumunun ikinci Y gözlemini üretme olasılığı	Geçiş olasılığı	T gözlemi
P1	$0.333*0.5=0.1665$	$0.1665*0.333=0.054$	$(0.054+0.0831+0.0277)*0.5=0.0831$		
P2	$0.333*0.75=0.24975$	$0.24975*0.333=0.0831$			
P3	$0.333*0.25=0.08325$	$0.08325*0.333=0.0277$			

Benzer şekilde geçiş olasılıkları tüm durumlar için belirlendiğinde sonuç tablosu Tablo 2.2 elde edilecektir.

Tablo 2.2. Verilen gözlem dizisinin olasılığının bulunması (b)

Durum	Gözlem dizisi $O=\{ Y,Y,T \}$ olasılıkları				
	İlk Y gözlemi	Durum geçiş olasılıkları	İkinci Y gözlemi	Durum geçiş olasılığı	T gözlemi
P1	0.1665	0.054	0.0831	0.0276	0.0414
P2	0.24975	0.0831	0.1246	0.0414	0.0207
P3	0.08325	0.0277	0.0415	0.0138	0.0621
Verilmiş $\lambda$ model için $P(O \lambda) = 0.0414 + 0.0207 + 0.0621 = 0.1242$ olur.					

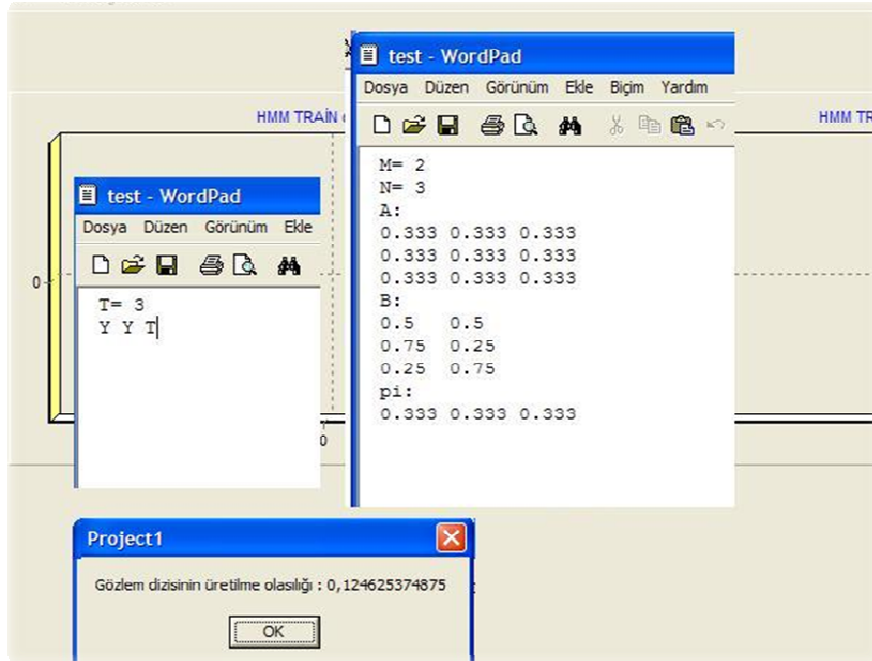
Bu problemin çözümü için önerilen forward algoritması ise Şekil 2.14'teki gibidir.

```

//Başlangıç olasılığının hesaplanması
for (i = 0; i < N; i++)
    alfa[0][i] = pi[i]*B[i][O[0]-1];
// Geçiş ve gözlem olasılıklarının hesaplanması
for (t = 0; t < T-1; t++) {
    for (j = 0; j < N; j++) {
        toplam = 0.0;
        for (i = 0; i < N; i++)
            toplam += alfa[t][i] * (A[i][j]);
        alfa[t+1][j] = toplam*(B[j][O[t+1]-1]);
    }
}
//Gözlem dizisinin olasılığının hesaplanması
olslk = 0.0;
for (i = 0; i < N; i++)
    olslk += alfa[T-1][i];

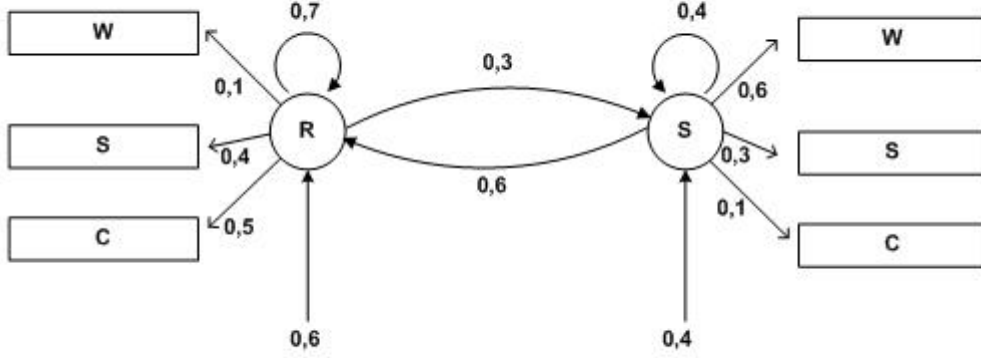
```

Şekil 2.14. Gözlem dizisinin olasılığının hesaplanması



Şekil 2.15. Gözlem dizisinin  $\lambda$  modeline göre üretilme olasılığının bulunması

Durum dizisinin bulunması problemi: Şekil 2.16'da verilmiş örnek için  $O=\{W,S,C\}$  gözlemini üretecek saklı durum dizisini bulalım.



Şekil 2.16. İki durumlu üç çıkışlı saklı Markov model

Bu örnek için verilmiş  $\lambda=(\pi, A, B)$  modelini çıkaralım,

$$\pi = \begin{bmatrix} 0,6 \\ 0,4 \end{bmatrix} \quad A = \begin{bmatrix} 0,7 & 0,3 \\ 0,4 & 0,6 \end{bmatrix} \quad B = \begin{bmatrix} 0,1 & 0,4 & 0,5 \\ 0,6 & 0,3 & 0,1 \end{bmatrix}$$

Tablo 2.3. Saklı durum dizisinin bulunması

Gözlemi üretecek mümkün durum dizileri	Başlangıç olasılığı	Gözlem olasılığı (W)	Durum geçiş olasılığı	Gözlem olasılığı (S)	Durum geçiş olasılığı	Gözlem olasılığı (C)	Sonuç $*10^{-6}$
RRR	0,6	0,1	0,7	0,4	0,7	0,5	5880
RRS	0,6	0,1	0,7	0,4	0,3	0,1	504
RSR	0,6	0,1	0,3	0,3	0,4	0,5	1080
SRR	0,4	0,6	0,4	0,4	0,7	0,5	13440
RSS	0,6	0,1	0,3	0,3	0,6	0,1	324
SRS	0,4	0,6	0,4	0,4	0,3	0,1	1152
SSR	0,4	0,6	0,6	0,3	0,4	0,5	8641
SSS	0,4	0,6	0,6	0,3	0,6	0,1	2592

Tablo 2.3'ten de görüleceği gibi  $\lambda$  modelde  $O=\{ W,S,C \}$  gözlemini üretecek maksimum olasılık değerine sahip  $Q=\{ S,R,R \}$  saklı durum dizisi olmaktadır.

Bu problemin çözümü için Viterbi algoritması kullanılmaktadır. Viterbi algoritması genel anlamıyla bir dinamik eşleştirme yöntemidir. Saklı Markov model için kullanımı, tanınacak kelimeye ait öznitelik gözlem vektörünü modele ait durumlara dağıtarak en uygun durum-gözlem vektör eşleşmesini bulmak ve buna ait olasılık değerini hesaplamak olarak gerçekleştirilir.

Bu problemin çözümü için kullanılan Viterbi algoritması Şekil 2.17'de verilmiştir.

Model parametrelerinin öğrenilmesi problemi: Parametre tahmin işlemi saklı Markov modelinin eğitilmesidir. Burada her durum için olasılık dağılım fonksiyonunun parametreleri ve durumlar arası geçiş olasılık değerleri belirlenmiş olur.

Baum-Welch algoritması, saklı Markov model parametre tahmin işlemini maksimum olabilirlik yaklaşımı ile çözen bir iyileştirme tekniğidir (2.19).

```

//Başlatma
for (i = 0; i < N; i++) {
    delta[0][i] = pi[i] * B[i][O[0]-1];
    ksi[0][i] = 0; }

//Rekürsif
for (t = 1; t < T; t++) {
    for (j = 0; j < N; j++) {
        max = 0.0; maxid = 1;
        for (i = 0; i < N; i++) {
            deger = delta[t-1][i] * A[i][j];
            if (deger > max) { max = deger; maxid = i;
            }
        }

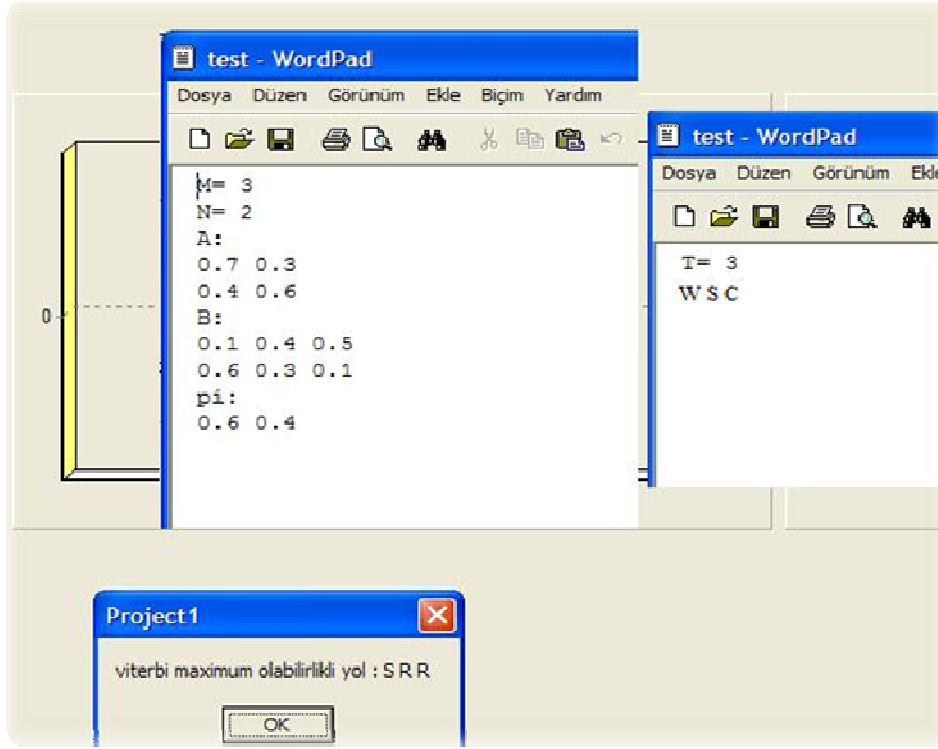
        delta[t][j] = max * B[j][O[t]-1];
        ksi[t][j] = maxid;
    }
}

//Sonlandırma
olslk = 0.0;
q[T-1] = 1;
for (i = 0; i < N; i++) {
    if (delta[T-1][i] > olslk) {
        olslk = delta[T-1][i];
        q[T-1] = i;
    }
}

//Dizinin elde edilmesi
for (t = T - 2; t >= 0; t--)
    q[t] = ksi[t+1][q[t+1]];

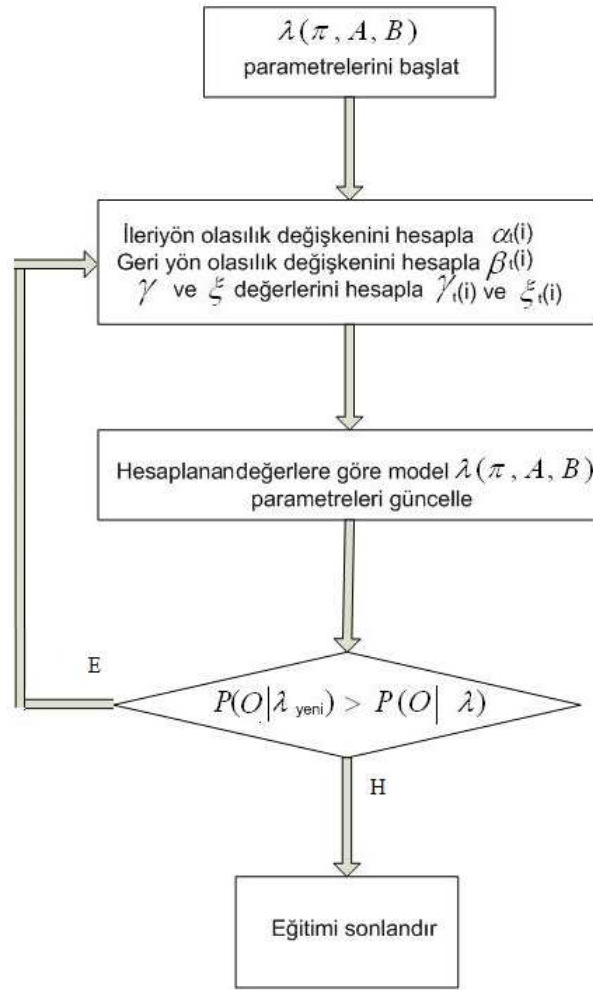
```

Şekil 2.17. Viterbi algoritması

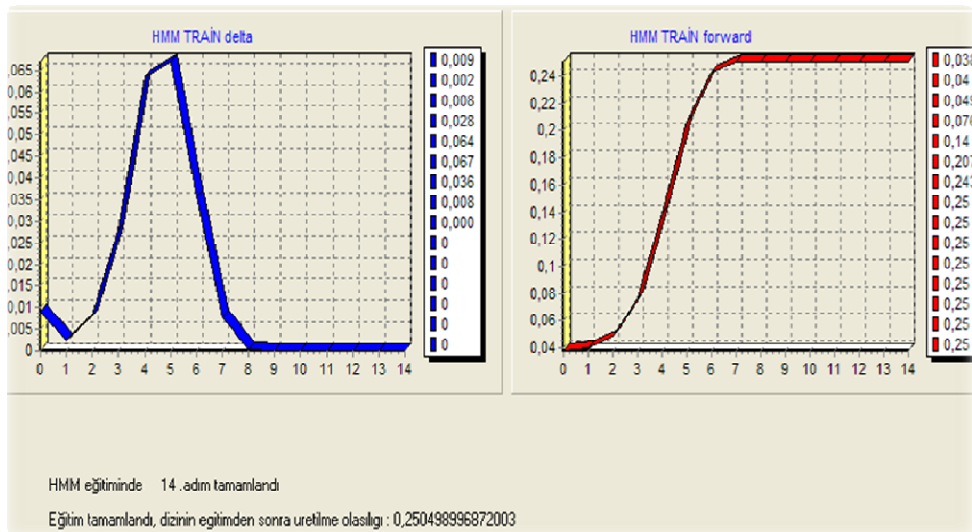


Şekil 2.18.  $\lambda$  modeline göre verilen gözlem dizisi,  $O=\{ W,S,C \}$ 'yi üreten saklı durum dizisinin bulunması





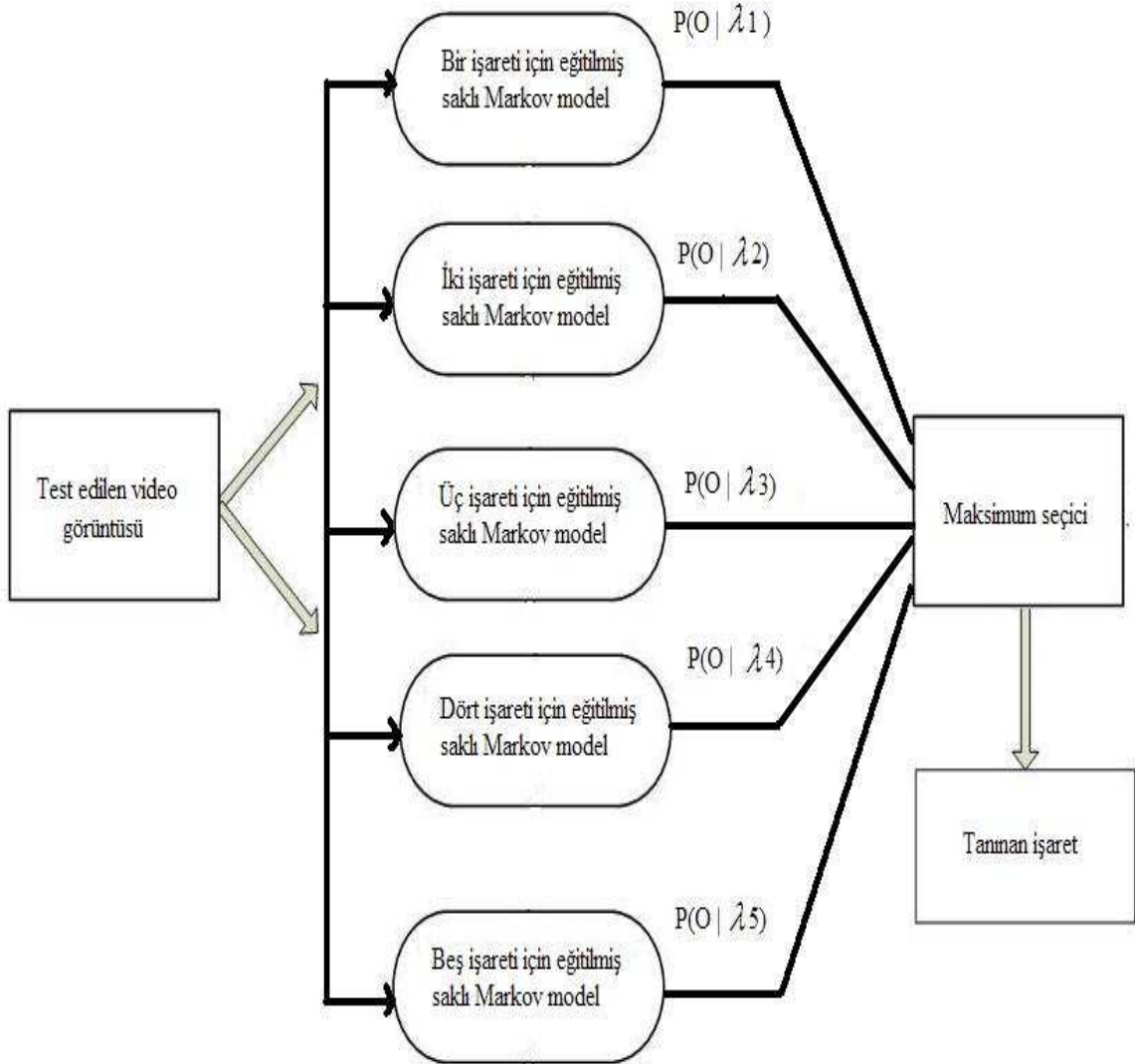
Şekil 2.19. Baum-Welch algoritması

Şekil 2.20. Verilen  $O=\{W,S,C\}$  gözlemi için iki saklı durum içeren Markov modelin eğitimi

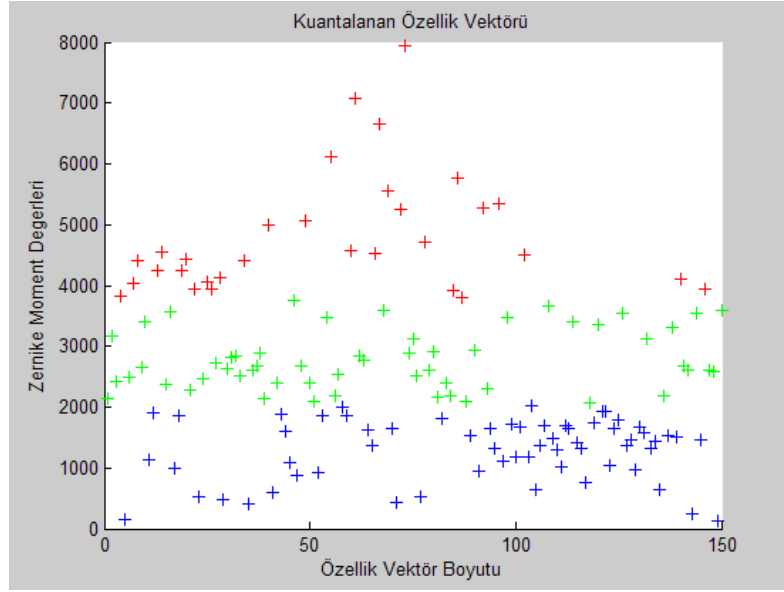
## 2.6. Sistemin Test Edilmesi

Bu tez çalışmasında oluşturulan dinamik Türk İşaret Dili tanıma sisteminin eğitimi için, işaret dilindeki sayılardan ilk beş sayı seçilmiştir. Dolayısıyla test aşaması için eğitilmiş, her biri bir sayıyı temsil eden beş saklı Markov model bulunmaktadır.

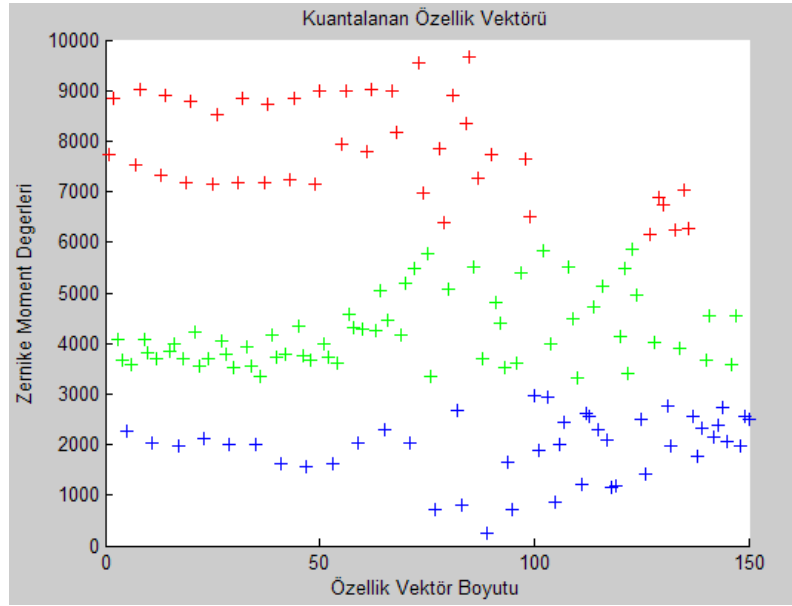
Test işlemi için kullanılan videodaki nesnenin özellik vektörü oluşturulup eğitilmiş olan bu beş modele giriş olarak verildiğinde, hangi model üzerinde ürettiği çıkış olasılık değeri daha maksimumsa test edilen işaretin o sınıfın elemanı olduğu söylenir (Şekil 2.21).



Şekil 2.21. Eğitilmiş beş Markov modelin bulunduğu sistemde test işlemi



Şekil 2.22. İşaret dilindeki üç sayısının eğitimi için kullanılan örnek özellik vektörü



Şekil 2.23. İşaret dilindeki üç sayısının test için kullanılan örnek özellik vektörü



Şekil 2.24. İşaret dilinde (1, 2, 3, 4, 5) sayılarının Markov modelinin oluşturulduğu sistemde üç sayısının test işlem sonucu

### 3. SONUÇLAR VE ÖNERİLER

İşaret dili tanıma sistemleri genel olarak statik ve dinamik olmak üzere ikiye ayrılabilir. Statik sistemlerde sabit görüntüde elin herhangi bir duruşunun yorumlandığı sistemlerdir. Dinamik sistemler ise elin işareti sırasındaki hareketinin yorumlanması ile ilgilidir.

Dinamik sistemlerde, görüntüde yorumlanacak elin hareket bilgisinin elde edilmesi için kullanıcıdan elektronik eldiven veya renkli eldiven takması istenmektedir. Bu yöntemlerin dışında hareketli görüntüden el bilgisi, çeşitli görüntü işleme teknikleriyle elde edilebilir. Böylece kullanıcıdan bağımsız, maliyeti daha ucuz sistemler oluşturulabilir. Çoğu alıcılarla dolu elektronik eldiven barındıran donanımsal tabanlı sistemler çok pahalı olduğundan, son yıllarda görüntü tabanlı sistemler üzerine çalışmalar arttığı görülmüştür.

Görüntüden işaret dili karakter tanıma konusunda Türkiye dâhil çeşitli ülkelerde çalışmalar olmasına rağmen, bilgisayarlı işaret dili tanıma konusunda yeterli özellikte bir çalışma bulunmamaktadır. Oluşturulan sistemler, bir anlatımın işaret dilinde yorumlanmasını sağlayacak şekilde değil de parmak alfabesinin tanınmasını ve birebir kelime olarak çevirisi yapacak şekildedir. Türk işaret dilinin veri tabanı olmadığı için yapılan çalışmalar kullanıcı bağımlı olmaktadır. Ayrıca her ülkenin işaret dili farklı olduğu için tasarlanan sistemler de ülkelere özgü olmaktadır.

Bu tez çalışmasında işaret dilindeki kelimeler, kullanıcıdan bağımsız bir şekilde çeşitli görüntü işleme teknikleri kullanılarak oluşturulan dinamik bir sistemle yorumlanmaya çalışılmıştır. Bu şekliyle daha esnek bir sistem olduğu söylenebilir; daha farklı problemlere de genelleştirilebilir.

Gerçek zamanlı hareketin yorumlanmasında işlemlerin mümkün olduğu kadar hızlı olması gerekir. Bu yüzden nesne takibi ve diğer ön işleme adımları, sistem tanıma zamanının az bir kısmını tüketmelidirler. Bu çalışmada sınıflandırma öncesi kullanılan ortalama kayma algoritmasıyla nesnenin takibi ve takip sırasında özellik vektörünün çıkarılmasında kullanılan Zernike moment yöntemi sistemin hızlı olmasını sağlamıştır.

Burada takip edilecek el bölgesi videodaki ilk çerçevede elle seçilmektedir. Takip işleminde de seçilmiş olan bölgenin takibi yapılmaktadır. Kelimeyi temsil edecek olan el şeklindeki değişimlerin algılanmasının daha rahat bir şekilde yapılabilmesi için takipteki pencere boyutlarının zamanla değişebilir olması önemli olabilmektedir.

Arařtırmalarda, iřaretlerin belli bir zaman iinde srekli bir hareketi ierdikleri iin tanıma yntemlerinden saklı Markov modeller bu alanda en ok tercih edilen etkili tekniklerden olduėu biri olduėu anlařılmıřtır. Bu alıřmada da sınıflandırma yntemi olarak tercih edilmiřtir.

#### 4. KAYNAKLAR

- Aran, O., Ari, İ., Benoit, A., Carrillo, A.H., Fanard, F.X., Campr, P., Akarun, L., Caplier, A., Rombaut, M. ve Sankur, B., SignTutor: An Interactive Sign Language Tutoring Tool, Proceedings of eNTERFACE'06, Dubrovnik, Croatia, 2006.
- Allen, J.M.; Asselin, P.K. ve Foulds, R., American Sign Language finger spelling recognition system, Bioengineering Conference, 2003.
- Bauer, B., Hienz, H. ve Kraiss K., Video-Based Continuous Sign Language Recognition Using Statistical Methods, 15th International Conference on Pattern Recognition (ICPR'00) - Volume 2, 2000.
- Comanicu, D., Ramesh V. ve Meer, P, Kernel-Based Object Tracking, IEEE Trans. Pattern Anal. Mach. Intell., 25 (5) (2003), pp. 564–577.
- Culver, V.R., A Hybrid Sign Language Recognition System Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on Volume 1, 31 Oct.-3 Nov. 2004 Page(s): 30–33.
- Erdem, U.M. ve Sclaroff, S., Automatic Detection of Relevant Head Gestures in American Sign Language Communication, Proceedings of Int. Conference on Pattern Recognition, 2002, vol. 1, pp. 460-463.
- Göknaar, G., Yıldırım, T., El İşaretlerinin Yapay Sinir Ağları ile Algılanması, IEEE 13. Sinyal İşleme ve İletişim Uygulamaları Kurultay kitabı, 16-18 Mayıs 2005, Kayseri.
- Haberdar, H. Albayrak, S., Real Time Isolated Turkish Sign Language Recognition From Video using Hidden Markov Models with Global Features. ISCIS 2005, Lecture Notes in Computer Science Vol. 3733. Springer-Verlag, Berlin Heidelberg (2005) 677-688
- Heloise Hse, A. Richard Newton, Sketched Symbol Recognition using Zernike Moments, icpr, vol. 1, pp.367-370, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 1, 2004
- Holden, E., Lee,G. ve Robyn, A. Owens: Australian sign language recognition. Mach. Vis. Appl. 16(5): 312-320 , 2005
- Isaacs, J. ve Foo, S.,Optimized Wavelet Hand Pose Estimation for American Sign Language Recognition Authors: Evolutionary Computation, 2004. CEC2004. Congress on. 07/2004; 1:797- 802 Vol.1.
- Liao S. X. ve Pawlak, M., A Study of Zernike Moment Computing. ACCV (1) 1998: 394-401

- Ming, K.W., Ranganath, S., Representations for Facial Expressions, Proceedings of International Conference on Control Automation, Robotics and Vision, Dec. 2002 pp. 716-721.
- Nabiyev V., Köse C. , Bayrak S., An artificial neural network approach for sign language vowels recognition, The international scientific conference on Problems of Cybernetic and Informatics (PCI), October 2006.
- Newport, E. L., Maturation constraints on language learning, Cognitive Science ,14 , 1990,11-28.
- Starner, T. , Pentland, A, Realtime American Sign Language Recognition from Video Using Hidden Markov Models, Technical report, MIT Media Laboratory, 1996.
- Ong, S.C.W., Ranganath, S, Automatic Sign Language Analysis: A survey and the Future beyond Lexical Meaning, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.27, no.6, 2005, pp.873-891.
- Pavlovic, V., Sharma, R. ve Huang, T. S., Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review., IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7,1997, pp. 677–695.
- Rabiner, L.R., A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE, vol. 77, Feb. 1989, pp. 257-286.
- Quek, F.K.H., Toward a Vision-Based Hand Gesture Interface, World Scientific Publishing Co., Inc., 1994, pp. 17–31.
- URL-1, <http://turkisaret dili.ku.edu.tr>, 16 Nisan 2009.
- Wang Q., Chen X., Zhang L., Wang C., Gao W. ,Viewpoint invariant sign language recognition Image Processing, 2005. ICIP 2005. IEEE International Conference on Volume 3, Issue , 11-14 Sept. 2005 Page(s): III - 281-4.
- Wu, Y. , Huang, T.S, Hand modeling, analysis, and recognition for vision based human computer interaction, IEEE Signal Processing Magazine, 2001,v.21, p.51–60.
- Zeshan, U., Aspects of Türk İşaret Dili (Turkish Sign Language), Sign Language and Linguistics, vol. 6:1,2003, 43-75.



## ÖZGEÇMİŞ

Selda BAYRAK, 1981 yılında Trabzon'da doğdu. İlk öğrenimini Fatih İlkokulu'nda, orta öğrenimini Mehmet Akif Ersoy İlköğretim Okulu'nda, lise öğrenimini Fatih Süper Lisesi'nde tamamladı. Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü'nden 2006 yılında mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans öğrenimine başladı. 2007 güz yarıyılından itibaren Bilgisayar Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalışmaktadır. Yabancı dil olarak İngilizce bilmektedir.