

# Bilgisayar Görüsü ve İmge İşleme

July 2, 2001

## Contents

<b>1 Giriş</b>	<b>1</b>
<b>2 Bilgisayar görüşü nedir?</b>	<b>1</b>
<b>3 İmge işleme (Image Processing)</b>	<b>2</b>
3.1 3 Boyutlu görüntüleme . . . . .	3
3.2 Fourier Metodları . . . . .	3
3.3 Fourier Uzayında imge yumuşatmak . . . . .	4
3.4 Gerçek uzayda imge yumuşatmak . . . . .	4
3.5 Kenar Tanımlama (Edge Detection) . . . . .	4
3.6 Gradient tabanlı kenar ucu tanımlama . . . . .	4
3.7 Kenar Birleştirme . . . . .	5
3.8 Yerel Kenar birleştirici metodlar . . . . .	5
3.8.1 Bölge Ayrırma (Region Splitting) . . . . .	5
3.8.2 Bölge büyümesi (Region Growing) . . . . .	6
3.9 Optik Akış . . . . .	6
<b>4 Cisim Tanımlama</b>	<b>7</b>
4.1 Değişmezler (Invariants) . . . . .	7
4.2 Bilgisayar görüşü için geometrik modelleme . . . . .	7
4.2.1 Tel çerçeve ( Wire frame) modeller . . . . .	8
4.2.2 Hesaplanabilir Katı Cisim Geometrisi ( Computational Solid Geometry) . . . . .	8

4.2.3	Sınır Gösterimi . . . . .	8
4.2.4	3 boyutlu modellerde istenebilecek özellikler . . . . .	9
4.3	Model tabanlı tanımlama . . . . .	9
4.3.1	Ağaç(Tree) tabanlı arama metodları . . . . .	9
4.3.2	Çizge araması ( Graph searching) . . . . .	9
<b>5</b>	<b>Çoklu görü geometrisi ( Multiple View Geometry)</b>	<b>10</b>
5.1	Türdeş konaç sistemi ( Homogeneous Coordinate System) . . . . .	10
5.2	Kamera kalibrasyonu . . . . .	11
5.3	Üçgenleme . . . . .	12
<b>6</b>	<b>Yazılım kaynakları</b>	<b>12</b>

## 1 Giriş

Önceleri yapay zekanın bir alt dalı olan bilgisayar görüşü (Computer Vision) bu gün akademik anlamda kendi çizgisinde ilerlemektedir. 1960'larda bir lisans öğrencisinin ödevi olabilecek, "bilgisayarlar nasıl görür ve tanımlar " sorusu bu gün başlı başına bir bilim dalı oluşturmuştur. Bu dökümanın amacı bilgisayar görüşü ve onun en önemli aracı olan imge işleme konularında başlangıç seviyesinde bir bilgi vermektir. Döküman şimdilik taslak aşamasında olması nedeniyle eksiktir ve anlatılan konuların genişliği nedeniyle hep eksik kalacaktır. Her türlü öneriyi ve eleştiriye ozdenkem@btae.mam.gov.tr e-posta adresine yazabilirsiniz.

## 2 Bilgisayar görüşü nedir?

Bilgisayar görüşü , çevreyle herhangi bir fiziksel etkileşim olmaksızın, optik araçlar kullanarak cisimleri algılama ve buna göre bilgi toplama ya da çeşitli süreçleri yönetmektir. Bu tanım çerçevesinde bilgisayar görüşü doğal olarak çok değişik bilim dallarıyla iş birliği içerisinde olmalıdır. Bir örnek vermek gerekirse en klasik bilgisayar görüşü problemlerinden olan kameradan görülen bir cismin 3 boyutlu resmini çıkarma sorunu ele alalım. Bu tarz bir sistemde şu disiplinlere ihtiyaç olacaktır,

- **Makine mühendisliği** : Cismi kameraya düzgün gösterebilmek kameraları ya da cismi hareket ettiren düzenek tasarımı.

- **Optik mühendisliği** : Kameradan alınan imge kalitesini iyileştirmek için çeşitli optik yöntemler kullanılabilir.
- **Kamera teknolojisi** : Kameralar çok çeşitlilik göstermektedir. Bunlar arasında duyarlılık, çözünürlük, ağırlık, elektriksel gürültüye dayanıklılık, titreşime, manyetik dalgalara, radyasyona, dayanıklılık , projeksiyon şekli ( perspektif, çizgisel gibi).
- **Sinyal işleme donanımı** : Video kameralar genelde bilgisayarların işleme kapasitesini çok zorlayacak miktarda bilgi üretmektedir bu yüzden özelleşmiş donanıma ihtiyaç olacaktır.
- **Yapay zeka** : Verilerin tanımını yapacak örüntü algılama(Pattern Recognition) yapacak yazılımın ve karar alacak uzman sistemlerin(Expert Systems) algoritmaları gerçekleştirilmelidir.
- **Yazılım Mühendisliği** : Algılanan veriyi işleyip karar alma gibi karmaşık işlemleri yapacak bir yazılım gerekecektir.
- **Matematik** : 2 boyutlu resimlerden 3 boyutlu cisimler çıkarmak ve bunları imge işlemlerini yapmak yüksek miktarda üst düzey matematik kullanmayı gerektirecektir. Bu da özellikle elektronik Mühendisliğinin ilgi alanına girmektedir.

### 3 İmge işleme (Image Processing)

Bir çok disiplinle ortak çalışan Bilgisayar görü sistemlerinin en ayrılmaz parçası imge işleme algoritmalarıdır. Bu algoritmalar girdi ve çıktı olarak 3 ayrı kate-goride incelenebilir.

```

İmge işleme      girdi : imge -> çıktı : imge
İmge analizi    girdi : imge -> çıktı : ölçümler
İmge Anlama     girdi : imge -> çıktı : tanımlama

```

Bu dökümanda temelde imge işleme ile ilgili konulara yer vereceğiz. Önce bazı tanımlamaları yapalım. İmge dediğimiz şey 2 boyutlu ve  $n*m$  büyüklüğünde bir matris ile modellenir. Bu matrisin her bir elemanı o pikselin siyah ve beyaz arasındaki şiddetini belirler. Piksellerin bit cinsinden değerleri ve imgenin yatay ve dikey piksel cinsinden boyutları, resmin kalitesini ve büyüklüğünü etkiler. 8 bitlik bir siyah beyaz bir imgede, eğer bir piksel 0 ise "siyah", 255 "beyazdır". Aradaki değerlerde orantılı olarak siyah-beyaz değerler alırlar.

İmgeleri daha iyi anlayabilmek için bir örnek verelim.  $200*200$  boyutunda ve her piksel için 8 bit kullanılmışsa.  $200*200*8/8$ (bit sayısı/ bir byte taki bit sayısı) = 40000 byte= 40 kilobyte lık bir alan kapsar. Bu bilgiler eğer resim

ikil eşlemlı (bitmap) řeklinde kayıtlıysa saklanırsa dođrudur. Bu dosyaların byklđn kltecek deđiřik sıkıřtırılmıř imge biimleri vardır.

Renkli imgelerin gsterim biimi biraz daha farklıdır. Deđiřik gsterim biimleri olmakla birlikte en yaygın olanı RGB [Red(Kırmızı), Blue(Mavi), Green(Yeřil)] gsterimidir. Bu gsterim her rengin bu  rengin karıřımıyla oluřturulabileceđini varsayar. Matematiksel olarak bu  renk iin ayrı ayrı matrisler dřnlebilir.

Bir imge iřleme algoritmasında ki girdisi bir imge ıktısı bařka bir imgedir,  eřit deđiřik durum gzlenebilir.

1. ıktıdının bir noktasındaki kona ( Coordinate) girdinin aynı konaına bađlıdır.
2. ıktının bir noktasındaki kona girdinin aynı konacının komřu konalarına bađlıdır.
3. ıktının bir noktasındaki kona, girdideki her piksel deđerine bađlıdır.

### 3.1 3 Boyutlu grntleme

En kolay  boyutlu imge gsterim řekli muhtemelen "derinlik haritası" ( depth map) řeklinde tutlan imgelerdir. Bu imgelerin 2 boyutlu gri dzeyi imgelerden farkı gri dzeyi bilgisi yerine derinlik dzeyi bilgisi bulundurmasıdır.

### 3.2 Fourier Metodları

Fourier dnřmler sinyal iřlemede nemli lde kullanılan bir yntemdir. Her sinyal temelde belirli bir frekanstaki sinyalin ve onun frekansının katları kadar farklı sinyalin, deđiřik oranlarda dođrusal(lineer) biimde birleřmesinde oluřur. rneđin tek boyutlu bir sinyali dřnelim ( ses mesela). Bu sinyal  $f, 2f, 3f \dots nf$  gibi frekansları olan deđiřik periyodik sinyallerden oluřacaktır. Bu sinyallerin katsayıları bize sinyalin karakteri hakkında nemli lde ip ucu verecektir. Sayısal ses tek boyutlu analog bir sinyalin rneklenmesiyle elde edilir ( rneđin saniyede 8000). Daha sonra bu sinyal yaklařı 20 ms'lik pencerelele blnr ve herbirinin Fourier dnřm alınır ve sonuc olarak ortaya spectogram denen ve sesin genel karakteriřtiđini veren bir grafik elde edilir.

İmgelerde ses gibi analog bir sinyalin rneklenmesi sonucu oluřmuřtur bu yzden ses analizinde kullanılan tekniklerin kullanılmasına řařırmamalıdır. İmgenin tek farkı 2 boyutlu bir sinyal olmasıdır.

Peki resimdeki frekansları azlıđı ya da fazlalıđı ne anlama gelmektedir. Eđer bir resimdeki frekanslar yksek ise resimde hızlı deđiřimler olmaktadır. Eđer frekans genel olarak azsa, o zaman entropisi az bir resimdir.

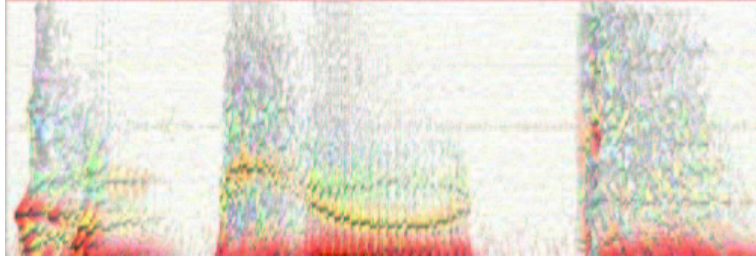


Figure 1: Bir spectrogram

Fourier methodları sürekli fonksiyonlar için tanımlanmıştır fakat resimlerdeki bilgi sayısal ve kesiklidir. Bunlar için DFT(Discrete Fourier Transfrom) kullanılır. DFT direk olarak kullanılırsa zaman alıcı bir algoritmadır fakat eğer problem düzgün parçalanırsa çok hızlı bir şekilde hesaplanabilir ve bu da FFT ( Fast Fourier Transform) algoritmasının temelidir.

### 3.3 Fourier Uzayında imge yumuşatmak

Bir imgede gürültü demek imgelerde genelde ani inişler ve çıkışlar demektir, bunlarda yüksek frekans bileşenlerine denk gelir. Eğer biz imgemizi bir frekans filtresinden geçirip yüksek frekanslara izin vermesek doğal olarak bir gürültünün etkisini azaltmış oluruz.

### 3.4 Gerçek uzayda imge yumuşatmak

Gerçek uzayda en kolay imge yumuşatma yöntemi, bir pikselin çevre piksellere göre ortalamasını almaktır.

### 3.5 Kenar Tanımlama (Edge Detection)

İmge işlemede kenar tanımlamak önemli bir yer tutmaktadır. Bunun nedenleri arasında kenarların bulunmasının az zaman alması ve cisimlerin tanımlanmasıyla ilgili önemli ipuçları vermesidir. Temelde bir kenar farkı yüzeylerin yan yana gelmesiyle oluşmuştur.

### 3.6 Gradient tabanlı kenar ucu tanımlama

Kenarların uçlarını tanımlamak için kullanılan yöntemlerden biri de kenar üzerindeki eğimin sürekliyken süreksiz bir atlama yapmasıdır. Bu bir köşenin göstergesidir. Değişik kenar çeşitleri vardır. Bunlardan bazıları :



### 3.7 Kenar Birleştirme

Kenar tanımlayıcılar aynı doğru üzerindeki noktaları bize ger dönerler. Bizim bir sonra yapmamız gereken bütün bu noktaları birleştirip tek bir doğruya çevirmemizdir. Ayrıca karşımıza çıkacak sorunlardan bazıları da şunlardır. Kenarın küçük parçaları kaybolmuş olabilir. İmgedeki gürültü resimde olmayan kenarları varmış gibi gösterebilir. Genelde kenar birleştirme yöntemleri ikiye ayrılır.

**Yerel Kenar birleştiriciler** Bu tarz algoritmalarda her nokta kendi komşusuyla olan ilişkisine bakıp kenarı oluşturular.

**Genel Kenar Birleştiriciler** Bu tarz algoritmalarda ise, resmin genelindeki bütün noktalar aynı anda belirli bir benzerlik kriterine uyuyor mu yoksa uymuyor mu diye denetlenir, doğru denklemleri gibi.

### 3.8 Yerel Kenar birleştirici metodlar

Bir çok kenar detektörü kenarın üzerindeki herhangi bir noktadaki eğimin yönü hakkında bilgi verir. Bu bilgi, kenarı oluştururken oldukça faydalıdır çünkü bir birine komşu noktalar aynı türe ve sahip olacaktır.

Yerel kenar birleştirici metodlar genelde belirli bir noktada başlayıp komşularını bir benzerlik testine sokarak devam ederler. Eğer noktalar benzerlik testine uyarlıysa noktalar şu andaki kenar kümesine eklenirler.

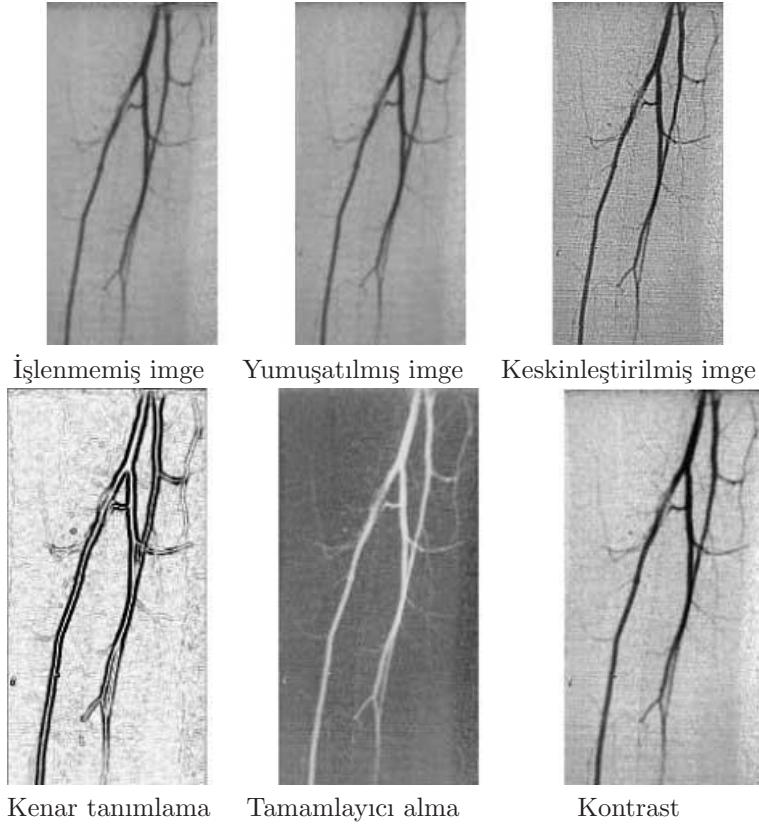
Daha sonra yeni eklenmiş noktalarla aynı işlem sürdürülür. Eğer noktanın çevresindeki herhangi bir nokta kenar özelliklerini taşıyorsa kenarın sonuna gelmiş demektir.

#### 3.9 Kesimleme(Segmentation)

İmgelerden bilgi almanın bir diğer yolunda kesimleme değildir. Kesimleme birbirine benzeyen belirli bir alanın diğerlerinde ayırmaya yarar.

- **2 Boyutlu** : Resimdeki pikseller yoğunluklarındaki değişime göre gruplanırlar.
- **3 Boyutlu**: Resimdeki pikseller derinliklerinin değişim hızına göre gruplanırlar.

Temel kesimleme şekilleri şunlardır.



### 3.8.1 Bölge Ayırma (Region Splitting)

Bölge ayırmanın temel prensibi şudur. İlk önce imge bir bütün olarak ele alınır. Sonra imgeye bakılır. Eğer bütün parçaları uyumluysa algoritma sonra erer. Eğer uyumsuzsa bölge 4 ana parçaya ayrılır ve her biri için algoritma devam eder. Bu işlem daha fazla bölünme olmayana kadar devam eder. Bir böl ve fethet taktiğidir. Olabilecek en kötü sonuç en küçük piksele kadar herşeyin bölünmesidir. Bu tür sonuçları engellemek için her bölünmeden sonra komşulara bakılır, uygun birleşme yapılabilir mi diye.

### 3.8.2 Bölge büyümesi (Region Growing)

Bölge büyümesi , bölge ayırmanın tam tersi yoldan giden bir algoritmadır. Başlangıç değerleri olarak küçük bölgeler birleştirilir. Herhangi bir başlangıç tohum(seed) nokta bulunarak bu yapılır. Bölgenin büyümesi durunca şu an herhangi bir bölgeye dahil olmamış noktalar seçilerek işleme devam edilir.

### 3.9 Optik Akış

Eğer zaman içinde bir dizi imge alırsak ve eğer ortamda hareket eden cisimler varsa, imge ve hareket eden cisim hakkında faydalı bilgiler toplanabilir. Örneğin hareket eden bir gök cismi düşünün. Değişik zamanlarda çekilmiş fotoğraflar incelenerek hangi pikselin cisme hangisinin hareket etmeyen uzaya ait olduğu anlaşılabilir. Eğer hareketleri düzgün inceleyebilirsek şu sorulara cevap verebiliriz.

1. Kaç tane hareket eden cisim var ?
2. Hangi yönlere hareket ediyolar ?
3. Düzgün yoksa belirli bir fonksiyona göre mi hareket ediyorlar
4. Ne kadar hızlılar.

Bir imge dizisinden, optik akış diye bir fonksiyon hesaplarız. Her piksel için  $V=(u,v)$  bulunur ki bu fonksiyon sahesinde şu değerler hesaplanır. Piksel resimde ne kadar hızlı hareket etmektedir. Piksel hangi hızda hareket etmektedir.

## 4 Cisim Tanımlama

Tanımlama adımı bir görü sisteminin son amaçlarından biridir. Cisim tanımlama

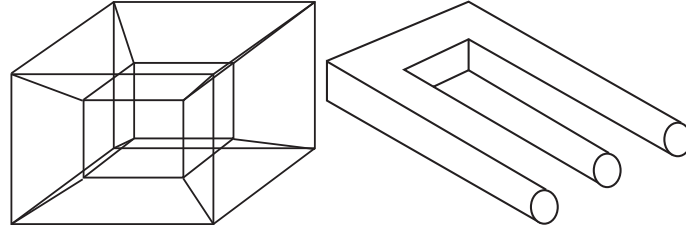
1. Cisimlere çarpmadan hareket etmek
2. Değişik cisimleri tutmak ve hareket ettirmek
3. Cisimleri incelemek

Ve diğer bir çok görev için kullanılır.

### 4.1 Değişmezler (Invariants)

Değişmezler basit cisimleri tanımlamakta kullanılabilir.

1. Cisimleri karşılaştırmak için çok basit bir yoldur.
2. Cisimlerin yerini ve yönünü tanımlamakta kullanılabilirler.
3. Bazı uygulamalar için yetersiz kalırlar.



## 4.2 Bilgisayar görüsü için geometrik modelleme

Bir geometrik modelleme için 3 boyutlu cisimleri gösterebilme yeteneğine ihtiyacımız vardır. Bu gösterimde :

1. Geometri
2. Topoloji
3. Diğer özellikler( Tolerans değerler, Renk, Kaplama etc.)

İmgelerden elde ettiğimiz 3 boyutlu görüntüleri gerçek modellerle karşılaştırmak en önemli uygulamalardan biridir.

### 4.2.1 Tel çerçeve ( Wire frame) modeller

1. Kenarlar ve köşe konaçları bir listede tutulur.
2. Polyhedral gösterim uygulanır.
3. Kenar tabanlı eşleştirme yapılır.
4. Karmaşık ve Olanaksız cisimler gösterilebilir.

### 4.2.2 Hesaplanabilir Katı Cisim Geometrisi ( Computational Solid Geometry)

Bu modelde cisim belirli bir şekli olan daha küçük cisimlerden oluşur. Örneğin küpler küreler vs. Modelde küme işlemleri kullanılır. Örneğin birleşme, kesişme, fark vs.

### 4.2.3 Sınır Gösterimi

Sınır gösterimi Hesaplanabilir katı cisim geometrisinin daha karmaşık ve gelişmiş bir halidir. Büyük bir bir topolojik ağ da birbirine bağlanmış küçük cisimlerden oluşmaktadır. Bu gösterim biçimi bilgisayar görüsü için en doğal gösterim yollarından biridir. Bu tarz gösterimde temelde iki çeşit özellik vardır

1. Topoloji : Cisimlerin ( yüzler, kenarlar, noktalar) birbirlerine nasıl bağlandığını gösterir
2. Geometri : Bağlı olan her cismin yapısını gösterir.

Bir köşe sadece x,y,z noktasından oluşur. Kenarlar çizgi, yada eğimli yaylardan oluşabilir.

#### 4.2.4 3 boyutlu modellerde istenebilecek özellikler

Bilgisayar görüşü için 3 boyutlu gösterim şunlara uygun olmalıdır.

1. Modeldeki cisim özelliklerine , gözlenen cisim arasında direk eşleyebilme
2. Direk konum ve yönelim hesaplanabilirliği
3. Görünümün her hangi bir bakış açısından hesaplanabilmesi.

### 4.3 Model tabanlı tanımlama

1. Tanımlama görüntü ile model arasında bir eşleme problemidir.
2. Eşleme klasik bir yapay zeka problemidir.
3. Bir çok değişik metod bu problemi aşmak için uygulanmıştır.
4. Hesaplama olarak karmaşık ve yoğun bir işlemdir.

Şimdi bunlardan önemli 2 tanımlama şeklini görelim.

#### 4.3.1 Ağaç(Tree) tabanlı arama metodları

Ağacın her düğümü tanımlanacak cisimler için belirli bir model primitiv eşlemesini gösterir. Ağacın m tane dalı vardır ki bunlar bunların herbiri bir model primitivine(kenar, yüzey vb.) denk gelir. Ağacın her düzeyi bir model primitivini betimler. Düzey ve düğüm bir eşleşmiş çift oluşturur. Sonra ağaçta bir tarama yapılır ve sonuçta bir eşleşme çifti ortaya çıkar. Sonuca yorumlama ağacı denir.

#### 4.3.2 Çizge araması ( Graph searching)

Cisim modeli ve manzara özellikleri (scene features) ilişkisel çizge yapısında (relational graph structure) gösterilir ki bu gösterim biçimi bilgisayar görü sistemlerinde oldukça yaygındır.

Bir çizge şunlardan oluşur

1. Birbirine bağlarla ilişkilendirilmiş düğüm kümesi
2. Her düğüm bir cisim özelliğini betimler
3. Düğümler gösterdikleri özelliklere göre değişik biçimde isimlendirilebilir (büyüklük, biçim, alan vb.)
4. Bağlar özellikler arasındaki ilişkileri belirler. Örneğin: Özelliklerin merkezi arasındaki uzaklık

Tanımlama :

1. Manzaradaki cisimle elimizdeki modeli eşleme sorunu.
2. Eşleme görünüm engelleyen diğer cisimleride göze almak zorundadır.
3. Elimizdeki cisim görüntüden elde edilen cisimden fazla özellikler sergileyecektir. Örneğin görünmeyen kısımlar.
4. Bu yüzden amacımız bütün model yerine bir alt kümeyi eşleme olmalıdır.
5. Bu oldukça büyük bir arama işlemidir.
6. Çizge teorisi başlı başına büyük bir konudur.

## 5 Çoklu görü geometrisi ( Multiple View Geometry)

Daha önceden belirtildiği üzere bilgisayar görüşünde en önemli sorunlardan birisi, bir kaç açıdan fotoğrafı çekilen bir cismin 3 d imgesini oluşturmaktır. Bu işlemin nasıl yapıldığını anlamadan önce bazı temel kavramlardan bahsetmek gerekir.

### 5.1 Türdeş konaç sistemi ( Homogeneous Coordinate System)

N boyutlu bir uzaydaki noktalar arasındaki ilişkiler dönüşüm olarak tanımlanır. Doğrusal dönüşümler en önemli dönüşümlerdir ve bir matrix ile gösterilebilirler. N boyutlu bir uzaydan M boyutlu bir uzaya olan doğrusal dönüşüm  $T[M][N]$  boyutlu bir matris ile belirlenebilir. Çok değişik dönüşüm biçimleri vardır. Bunlardan ilk akla gelenler :

1. Yer değişimi (Displacement) : konum ve yönelim değişimi. 6 tane özgürlük boyutu (D.O.F) vardır. Bunları 3 ü konum 3 de yönelim içindir. En önemli doğrusal dönüşümlerden biridir. Temelde yerel koordinat sistemindeki bir cismin hareketini belirlemek için kullanılır.

2. Perspektif izdüşüm : 3 boyutlu uzaydaki perspektif bir kameranın 11 özgürlük boyutu vardır ve 6 sı kameranın yer değişimi içindir. Perspektif bir kamera temelde 3d -i 2d dönüşüm yapar ve z bileşeninin büyüklüğüne göre oranlama yapar.
3. Orthogonal izdüşüm : Perspektif gibi z ye göre oranlama yapılmaz.

Yukarıda sayılan dönüşümler 3 boyutlu uzayda ya da  $3 \rightarrow 2$  boyutlu uzaylarda geçmektedir. Bunların benzerleri  $N$  boyutlu bir uzayda da vardır.

Daha önce belirtildiği gibi bu dönüşümler matris olarak tanımlanır ve bir  $\mathbf{x}$  vektörü dönüştürülmek istendiğinde bu matrisle çarpılır. Türdeş konaç sistemi bu tarz sistemlerde bildiğimiz normal konaç sistemine göre oldukça fazla yarar sağlar.

Türdeş konaç sisteminde,  $N$  boyutlu bir nokta  $N + 1$  bir elemanla gösterilir yani bir fazlalık vardır. Örneğin 2d uzay  $(x, y)$  noktası türdeş konaç sisteminde  $(kx, ky, k)$  gösterilir.  $k$  burada değişen değerler alabilir ama sonuç hep aynıdır.  $(1, 2) = (1, 2, 1) = (2, 4, 2) = (3, 4, 3)...$  vb. yani  $(x, y, w)$  türdeş noktası türdeş olmayan sistemde  $(x/w, y/w)$  ye dönüşür.  $w$  nin sıfır olduğu zaman nokta sonsuzdadır. Yani  $(1, 2, 0)$   $(5, 3, 0)$  noktaları sonsuzdadır. Homojen konaç sistemi sayesinde sonsudaki noktaları gösterebiliyoruz. Bu önemli bir sonuçtur, çünkü perspektif izdüşümde paralel doğrular sonsudaki bir noktada birleşir . Başka bir deyişle sonsudaki noktalar imge düzleminde gerçek bir noktaya denk gelmelidir,  $M \times x$  dönüşümünde  $x$  in sonsuzda mı ya da normal bir boyutta mı olduğu bizi ilgilendirmemelidir ve türdeş konaç sisteminin kullanılması bunu sağlar.

## 5.2 Kamera kalibrasyonu

Bir cismin tanımlanmasından önceki ilk süreç kamera parametrelerinin bulunmasıdır. Bu bir kalibrasyon cismi sahesinde olabilir. Bu cisimden ve belirli bir kameradan çekilen imgeden elde edilen eşleştirmeler kamera kalibrasyon matrisini veririr. Örnek verelim. Bir tane perspektif kameramız olsun. Bu kameramızın bir dış bir de iç parametreleri vardır. Dış parametreleri konum ve yönelim (  $3+ 3 = 6$  özgürlük derecesi) iç parametreleri ise ( optik uzunluk,  $x$  yönündeki birim uzunluk,  $y$  yönündek birim uzunluk vb. toplam 5 tane). Bir perspektif kameranın  $3*4$  boyutunda bir matrisi vardır ve toplam 11 parametrenin bulunması gerekmektedir. Bir kalibrasyon algoritması temelde 3d-2d eşlenikleri girdi olarak alır. Bu eşleşmeler ya imge işleme algortimaları sayesinde ya da bizim tarafımızdan elimizle işaretlenerek bulunur. Bu algoritmanın 11 parameter için 11 denkleme ihtiyacı vardır ve her 3d-2d eşleşmesinde 2 denklem gelir. Sonuçta bu problemi çözmek için 6 eşleşme yeterlidir.

Fakat  $2d$  (imge üzerindeki) ölçümler büyük bir doğrulukla yapılabilmesine rağmen  $3d$  noktaların ölçümü oldukça hatalı olabilmektedir. Bunu telafi edebilmek için çok daha fazla sayıda eşleşmeye ihtiyaç vardır (Genelde  $5*$  minimum). Bu denklemleri direk çözmek yerine bir iterasyonla hatayı en aza in-

dirgemek gerekir. Bunun için Newton İterasyonu ya da onun bir varyasyonu olan Levenberg-Marquard algoritması kullanılabilir.

Daha detaylı bir bilgi için “Multiple View Geometry : Richard Hartly, Andrew Zisserman” mükemmel bir kaynaktır. Kamera kalibrasyonu hakkındaki sorularınızı bana çekinmeden yazabilirsiniz. (ozdenkem@btae.mam.gov.tr).

### 5.3 Üçgenleme

İnge üzerindeki bir pikseli yaratan 3 boyutlu noktayı bulmak için en az iki kamera ya da farklı açılardan çekilmiş 2 ayrı görüntü gerekir. Bunu şöyle kısaca anlatabiliriz. Bir pikselin 3 boyutlu uzaya dönüşümü bir ışındır. Bu ışın üzerindeki herhangi bir nokta bu pikseli yaratabilir. Aynı noktanın izdüşümüne ait 2 ayrı ışını kesiştirebilirsek bu noktayı bulmuş oluruz. Buna üçgenleme denir ve en az iki ayrı kameraya gerek vardır. Fakat daha az hatalı sonuçlar elde etmek için daha çok kamera kullanılır.

## 6 Yazılım kaynakları

Bilgisayar görüşü hakkında bazı yazılım ve araştırma linkleri aşağıda verilmiştir.

- Active Appearance Models - Extensions and Cases
- BrainTech’s Odysee Development Studio
- Camera Calibration
- Convex grouping algorithm
- Disparity Analysis of Images
- Edge list approximation code
- Gradient Vector Flow Active Contour
- GSnake
- Hidden Markov Model routines
- Intel Open Source Computer Vision Library
- Intel Video Capture Card Libraries for Linux
- KLT
- Logical/Linear Operators
- Maximum-Flow Stereo Algorithm

- MeasTex
- MegaWave
- Microsoft Easy Camera Calibration Tool
- MMach
- NIST Handwriting OCR Testbed
- PCVideo Homepage
- Perceptual Organization Software
- Philip Torr's stereo vision code
- SAMPLEX Color Classifier
- SatherVision
- Segmentation of Skin-Cancer Images
- SketchUp
- SRI Stereo Engine Software
- Steerable Pyramid
- SUSAN
- TargetJr - A C++ Computer Vision Environment
- ToolDiag
- Uncalibrated Stereo by Singular Value Decomposition
- University of Calgary vision software
- UNL Fourier Features (UFF)
- XVision visual tracking software