Bilgisayar Görmesi

Ders 7:KORELASYON VE İKİ BOYUTLU DÖNÜŞÜMLER

Dr. Öğr. Üyesi Serap ÇAKAR

Korelasyon

Korelasyon bir görüntüde bilinen bir şeklin varlığını tespit etmek için kullanılır. Bir görüntüde arama yapan bu yaklaşımın birçok dezavantajı vardır. Nadiren görüntüdeki nesnenin yönü ve gerçek boyutu bilinebilir. Ayrıca, bunlar bir nesne için bilinse bile bütün nesneler için tutarlı olması çok zordur.

Korelasyon

Sabit bir kamera kullanan bir bisküvi üreticisi bir tepsideki iyi biçimli yuvarlak bisküvilerin sayısını template eşlemesi ile sayabilir. Bununla birlikte eğer görev, sonar bir görüntüdeki batık bir gemiyi aramak ise korelasyon kullanmak için en iyi yöntem olmayacaktır.

Korelasyon

Klasik korelasyon template’in ortalamasını ve altındaki görüntünün ortalamasını göz önünde bulundurur. Sabit bir görüntüde örneğin görüntü boyunca sabit bir aydınlık ve piksel değerlerinin dağılımının sabit olduğu görüntüde korelasyon aşağıda gösterilen teknikteki gibi konvolüsyon olarak basitleştirilebilir.

Korelasyon

- Korelasyon template eşlemesinin nerede olduğunu bulmak için kullanılır,

- Eğer *N* x *M'lik bir görüntü I*(*X,Y*) ile ve n x m'lik bir template *t*(*i,j*) ile gösteriliyorsa;

Korelasyon

$$corr(X,Y) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left[t(i,j) - I(X+i, Y+j)\right]^2$$

$$= \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left[t(i,j)^2 - 2t(i,j)I(X+i,Y+j) + I(X+i,Y+j)^2\right]$$

$$= \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left[t(i,j)\right]^2 - 2\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} t(i,j)I(X+i,Y+j) + \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left[I(X+i,Y+j)\right]^2$$

$$A \qquad\qquad\qquad B \qquad\qquad\qquad C$$

Korelasyon

A görüntü boyunca sabittir, dolayısı ile ihmal edilebilir. B, t ile I'nın konvolüsyonudur. Eğer görüntünün ortalama ışık yoğunluğu sabit ise C sabittir.

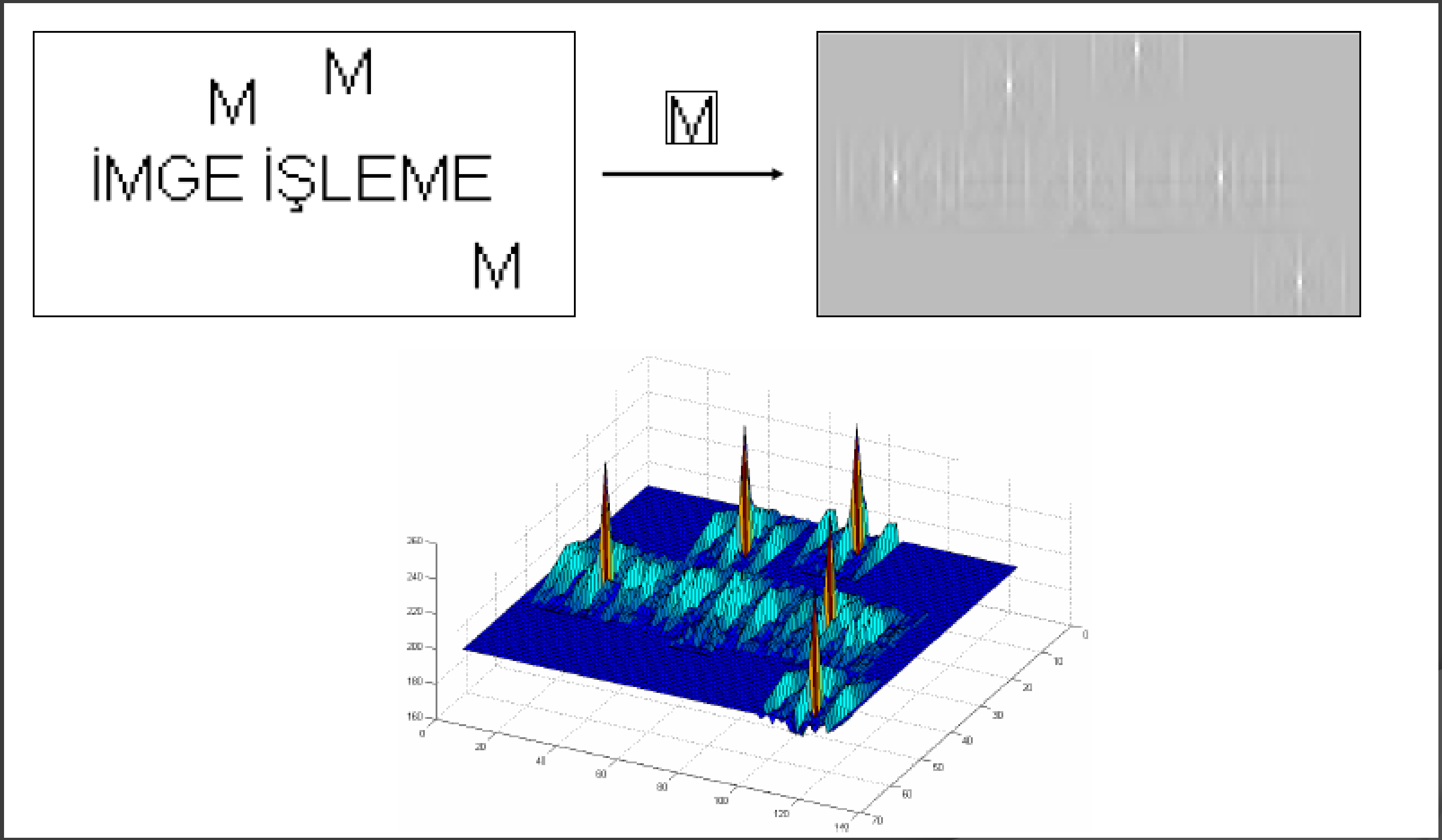
Korelasyon

Bu, çarpma, kare alma ve toplama içeren korelasyonu çarpma ve toplama içeren konvolüsyon olarak azaltır. Böylece, eğer ışık yoğunluğu görüntü boyunca sabit ise korelasyon yerine konvolüsyon kullanmaya değer.

Matlab

- C = CONV2(A,B) ; %konvolüsyon
- R = CORR2(A,B);   %korelasyon

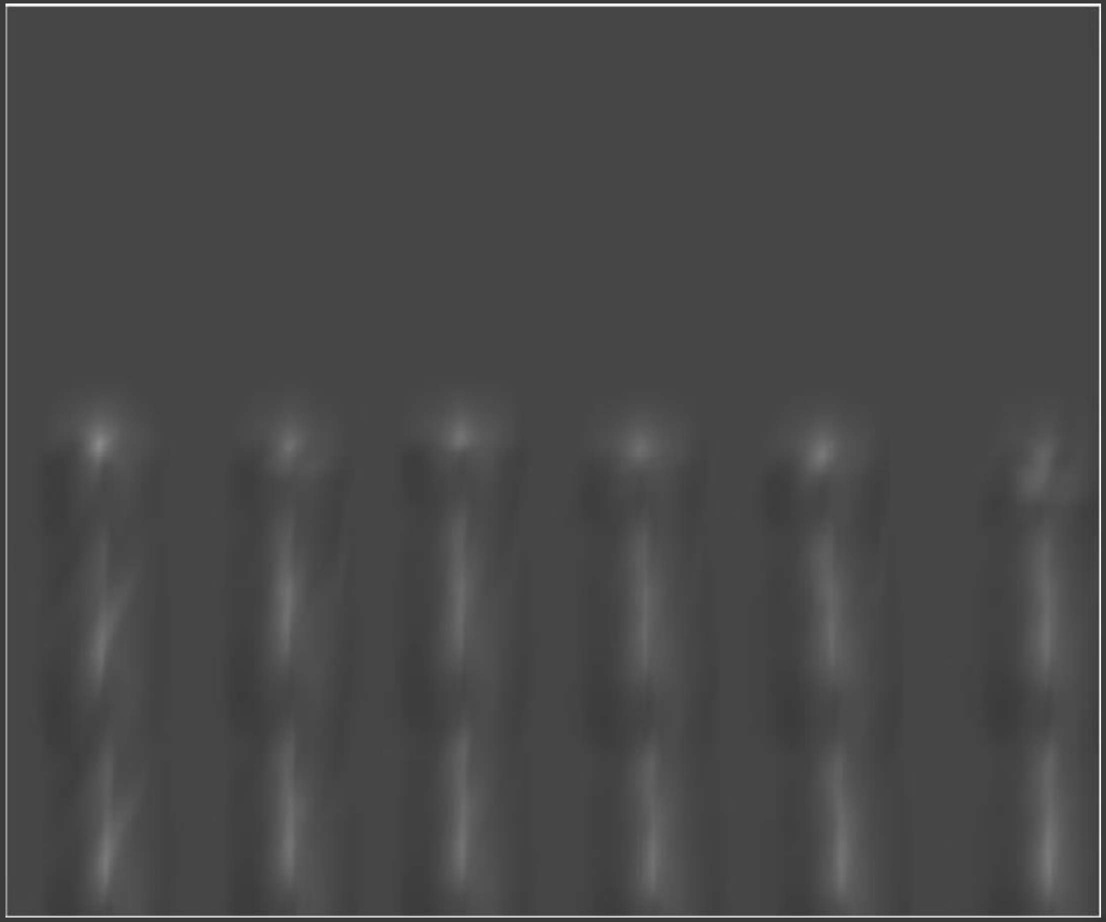
Örnek uygulama



Örnek uygulama

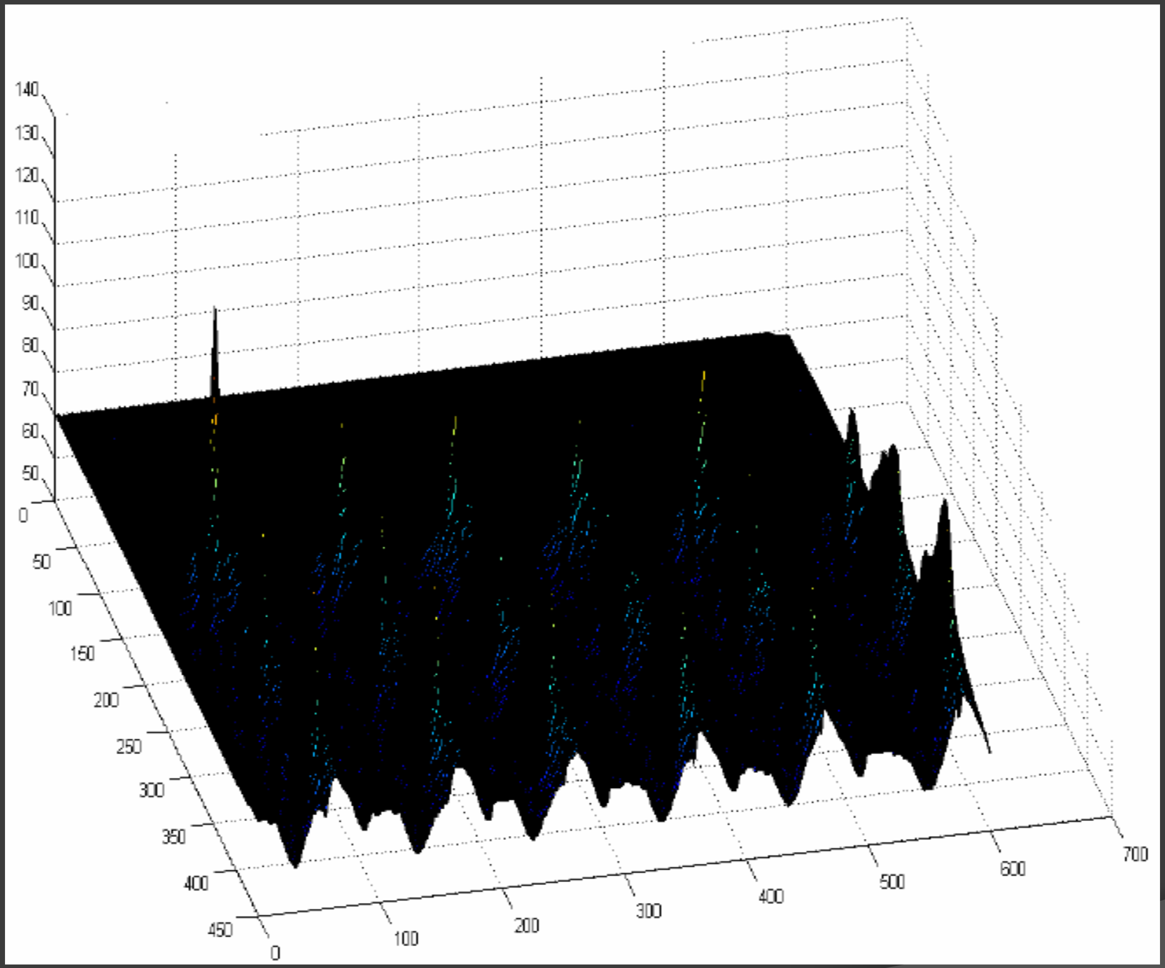
Sağlam matkap ucu

Örnek uygulama



Elde edilen korelasyon görüntüsü

Örnek uygulama



Elde edilen korelasyon yüzeyi

İki boyutlu geometrik grafik dönüşümleri

Bir görüntüyü yaklaştırmak, uzaklaştırmak, döndürmek, ötelemek genellikle faydalıdır. Bu operasyonlar Bilgisayar Grafiklerinde ve çoğu matematiği kapsayan grafik tekslerinde çok yaygındır. Bununla birlikte Bilgisayar grafikleri dönüşümleri iki boyutlu nesne koordinatlarından yeni iki boyutlu nesne koordinatları oluşturur. Örneğin eğer (x', y') yeni koordinatlar ve (x, y) orijinal koordinatlar ise (x', y') = f(x, y) eşlemesi oluşturulur.

İki boyutlu geometrik grafik dönüşümleri

Bu, görüntü işlemede tatmin edici bir yaklaşım değildir. Görüntü işlemede *x, y* ve *x', y'* değerleri tamsayı değerlerdir. Bütün x ve y değerleri için tanımlanmış olan f fonksiyonu, bütün *x' ve y'* değerleri için tanımlanmamış olabilir. f'in F olarak adlandırılan kayıplı ters dönüşümünü hesaplamak gerekir. Böylece yeni görüntüdeki her bir piksel için eski görüntüden gelen bir yoğunluk değeri tanımlanır.

İki boyutlu geometrik grafik dönüşümleri

İki problem ile karşılaşılır;

1. $0 \leq x \leq N\text{-}1$, $0 \leq y \leq M-1$ değer aralığı F fonksiyonu ile adreslemek için yeterli genişlikte olmayabilir. Örneğin bir görüntü merkez pikseli etrafında 90 derece döndürüldüğünde eğer görüntü 1:1 oranında değilse, görüntünün bir kısmı ekranın altında ve üstünde gözükmeyecektir ve yeni görüntü ekran için yeterli genişlikte olmayacaktır.

İki boyutlu geometrik grafik dönüşümleri

2. Her (x', y') pozisyonu için (x, y) pozisyonu haricinde yeni grilik seviyelerine ihtiyaç duyulur. Böylece yeni dizi pozisyonu veren bir fonksiyona ve eski diziye ihtiyaç duyulur.

$I(x, y)$ = F(eski görüntü, x', y')

İki boyutlu geometrik grafik dönüşümleri

f’(*x’*,*y’*)’nün (*x’*,*y’*) tamsayı çiftini vermesi olası olmadığından eski görüntünün bir arguman olarak verilmesi gerekir. Üretilen x ve y değerlerinin basitçe yuvarlanması veya f’(*x’*,*y’*) pozisyonu etrafındaki piksellerin ortalaması kullanılabilir. Grafikteki matris yöntemini kullanmak hala mümkündür. Orijinal piksel pozisyonunun hesaplanması için sonuç piksel pozisyonuna ters dönüşüm uygulanabilir.

İki boyutlu geometrik grafik dönüşümleri

- x yönünde sx ile ve y yönünde sy ile ölçekleme (yaklaştırma ve uzaklaştırma denktir)

$$(x', y', 1) = (x, y, 1)\begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

İki boyutlu geometrik grafik dönüşümleri

- x yönünde tx ile ve y yönünde ty ile öteleme (kamerayı sağa, sola, yukarı ve aşağı çevirmeye denktir)

$$(x',y',1)=(x,y,1)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -tx & -ty & 1 \end{bmatrix}$$

İki boyutlu geometrik grafik dönüşümleri

- Bir görüntüyü saat yönünün tersinde döndürme

$$(x',y',1)=(x,y,1)\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

İki boyutlu geometrik grafik dönüşümleri

- Bir görüntüyü yatay yönde germe

$$(x', y', 1) = (x, y, 1)\begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

İki boyutlu geometrik grafik dönüşümleri

- Bir görüntüyü dikey yönde germe

$$(x',y',1)=(x,y,1)\begin{bmatrix} 1 & \beta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ters Dönüşümler

- x yönünde sx ile ve y yönünde sy ile ölçekleme (yaklaştırma ve uzaklaştırmaya denktir).

$$(x',y',1)=(x,y,1)\begin{bmatrix} 1/sx & 0 & 0 \\ 0 & 1/sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ters Dönüşümler

- x yönünde tx ile ve y yönünde ty ile öteleme (kamerayı sağa, sola, yukarı ve aşağı çevirmeye denktir)

$$(x',y',1)=(x,y,1)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ tx & ty & 1 \end{bmatrix}$$

Ters Dönüşümler

- Bir görüntüyü saat yönünde döndürme. Bu dönme orijini normal grafik orijini olarak kabul eder ve yeni görüntü eski görüntünün α ile saat yönünde döndürülmüş halidir.

$$(x', y', 1) = (x, y, 1)\begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & & 1 \end{bmatrix}$$

Geometrik Dönüşümler

Bu dönüşümler dönüşüm matrislerinin çarpılması ile ve görüntüye uygulanması ile kombine edilebilir.

Matlab’de örnekler

```
im=imread('C:\lena512.JPG');
```

Ölçekleme

```
tform = maketform('affine', [0.5 0 0; 0 0.5 0; 0 0 1]);
imt = imtransform(im, tform);
```

Döndürme

```
tform = maketform('affine', [cos(0.52) –sin(0.52) 0; sin(0.52)
                                                  cos(0.52) 0; 0 0 1]);
imt = imtransform(im, tform);

imshow(imt);
```

Matlab’de örnekler

```
im=imread('C:\lena512.JPG');
```

Yatay yönde germe

```
tform = maketform('affine', [1 0 0; 0.3 1 0; 0 0 1]);
imt = imtransform(im, tform);
```

Dikey yönde germe

```
tform = maketform('affine', [1 0.3 0; 0 1 0; 0 0 1]);
imt = imtransform(im, tform);

imshow(imt);
```

Matlab'de örnekler

- Aynalama
- L1=fliplr(im);
- L2=flipud(im);

Renkli resim için aynalama

```
I = imread('onion.png');
I2 = flipdim(I ,2); %# horizontal flip
I3 = flipdim(I ,1); %# vertical flip
I4 = flipdim(I3,2); %# horizontal+vertical flip
subplot(2,2,1), imshow(I) subplot(2,2,2), imshow(I2)
subplot(2,2,3), imshow(I3) subplot(2,2,4), imshow(I4)
```