

Birinci baskı için ilgili düzeltmeler aşağıda verilmiştir.

Sayfa 13

Örnek Problem; Zıt yönlerde doğru giden ışık ışınları arasındaki hız farkı nedir?

Bu problemin cevabı Denklem (15) kullanılarak verilebilir;

$$v_{toplam} = \frac{v_1 + v_2}{1 + \frac{v_1 \cdot v_2}{c^2}} = \frac{2v}{2} = v \quad (15)$$

Sayfa 134

Şekil 98’de 2D Ayrık Dalgacık Dönüşümü kullanılarak gerçekleştirilen görüntü keskinleştirme (pansharpening) uygulaması sunulmuştur. Bu uygulamaya ait uygulama kodu Şekil 99’da verilmiştir.



$$MSE(ps,msi)+MSE(gray(ps),pan)=250.9724$$

Şekil 98: 2D Ayrık Dalgacık Dönüşümü kullanılarak gerçekleştirilen görüntü keskinleştirme (pansharpening) uygulaması.

Sayfa 139

Şekil 107’de Dominant piksel tabanlı median filtre için temel uygulama kodu verilmiştir.

```
> fnc=@(x,c,T) median(x( abs(x-x(c,c)) <= T ));  
> s=Kampus;  
> for i=1:3,  
> s(:,i)=blkproc(s(:,i),[1 1],[8 8],fnc,9,40);  
> end
```

Şekil 107: Dominant piksel tabanlı median filtre için temel uygulama kodu.

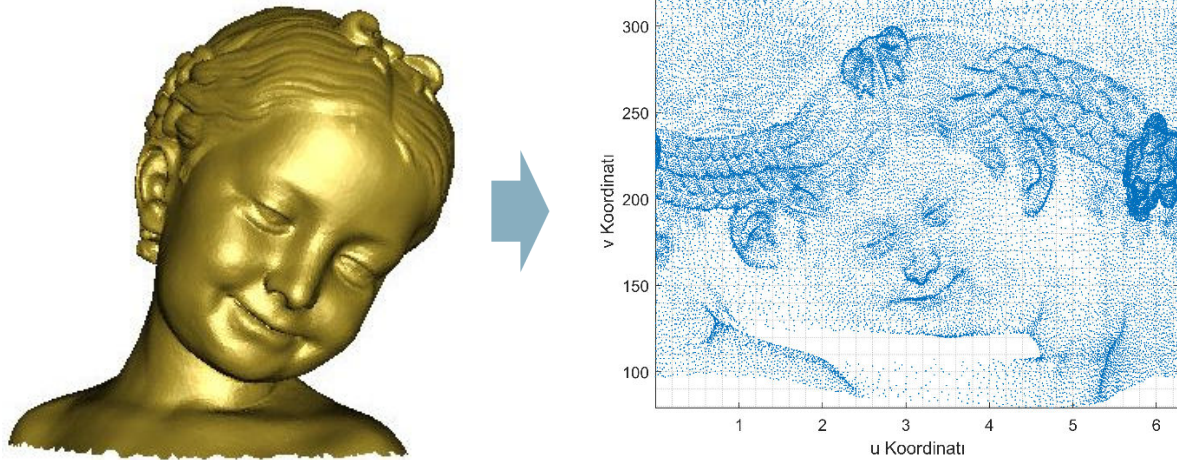
Burada, $\frac{\partial I}{\partial x}$ görüntünün x yönündeki değişim oranını temsil eder (yatay türev). $\frac{\partial I}{\partial y}$

görüntünün y yönündeki değişim oranını temsil eder (dikey türev).

Dijital görüntülerde türev almak, sayısal farklılaştırma operatörleriyle yapılır. En yaygın kullanılan sayısal farklılaştırma operatörleri, ileri fark operatörü ve **komşu** farklar operatörüdür. İleri fark operatörü, Denklem (56) kullanılarak tanımlanır.

$$\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y) \quad (56)$$
$$\frac{\partial I}{\partial y} \approx I(x, y+1) - I(x, y)$$

Sonraki Baskıya Eklenecek Yeni Konular



Şekil xxx : Bir Nokta bulutunun silindirik projeksiyon kullanılarak gösterilmesi.

Şekil xxx için kullanılan uygulama fonksiyonu (cylindricalProjection.m) aşağıda verilmiştir.

```
function [u, v] = cylindricalProjection(x, y, z)
    % 1) Açı (theta) hesapla: [ -pi, pi ] aralığında sonuç üretir.
    theta = atan2(y, x);
    % 2) Theta değerini [0, 2*pi] aralığına kaydır.
    negatifIndeks = theta < 0;
    theta(negatifIndeks) = theta(negatifIndeks) + 2*pi;
    % 3) Çıktı koordinatlarını belirle:
    %   u = theta, v = z
    u = theta;
    v = z;
end
```

cylindricalProjection(.m fonksiyonunun kullanımı aşağıda gösterilmiştir.

```
[u,v]=cylindricalProjection(x,y,z);
plot(u,v,'.','markersize',1);shg
axis tight
shg
xlabel('u Koordinatı');
ylabel('v Koordinatı');
grid minor
shg
```