BİLGİSAYARLARLA GÖRME PROJE ÖDEVİ

Projenin Amacı:

Stereo-Vision konusu, aynı doğrultuda ve aynı yöne bakacak şekilde yerleştirilen kameralardan alınan görüntülerle derinlik hesaplamayı içerir. Derinlik hesaplama, görüntüdeki cisimlerin kameraya olan uzaklığını göreceli olarak belirlemedir.

Yöntem:

- 1. Kameralardan alınan y eksenine göre hizalanmış ve açısal bozulma giderilmiş görüntüler gri seviyeye dönüştürülür.
- 2. Merkez pikselin etrafında bakılacak alanın büyüklüğünü belirleyen correlation window size (w*2+1 x w*2+1) belirlenir.
- 3. Sağ görüntüde correlation window un aranacağı daha geniş bir alan belirlenir. Search region olarak adlandırılan bu alan merkez pikselden d miktarı doğu ve batı yönünde mesafeye eşittir. Search region 2*d+1 genişliğinde ve w yüksekliğinde bir alanı kapsar.
- 4. Görüntü sınırlarının aşılmasını engellemek için d kadar padding işlemi yapılır.
- 5. Sol görüntüdeki her bir piksel için aşağıdaki işlemler tekrarlanır:
 - Piksel pencerenin merkezinde olacak şekilde correlation window sol görüntüden kırpılır.
 - Sağ görüntüdeki search region içinden, correlation window ile aynı büyüklükte bir search window kırpılır.
 - Correlation window ile search window karşılaştırılır.
 - Search region içinde kaydırılan search windowlar için aynı işlem tekrarlanır.
 - Correlation window a en çok benzeyen search window un arasındaki mesafe ölçülerek o piksel için kayma miktarını gösteren disparity değeri belirlenir.
- 6. Karşılaştırma işlemi için sum of squarred distance, simple cross correlation ve normalized cross correlation yöntemleri kullanılır. Birbirine en çok benzeyen windowları bulmak için sum of squarred distance değerlerinin minimumu, simple cross correlation ve normalized cross correlation için maksimumu aranır.
- 7. Her bir piksel için kayma miktarını gösteren disparity map gri seviyede gösterilir. Kameraya yakın olan cisimlerin kayma miktarı uzakta olanlara göre daha büyük olacağı için, yakındaki cisimler beyaza yakın, uzaktaki cisimler siyaha yakın görünür.

Correlation window, search window ve search region görüntüler üzerinde gerçekçi olmayan boyutlarla gösterilmiştir:



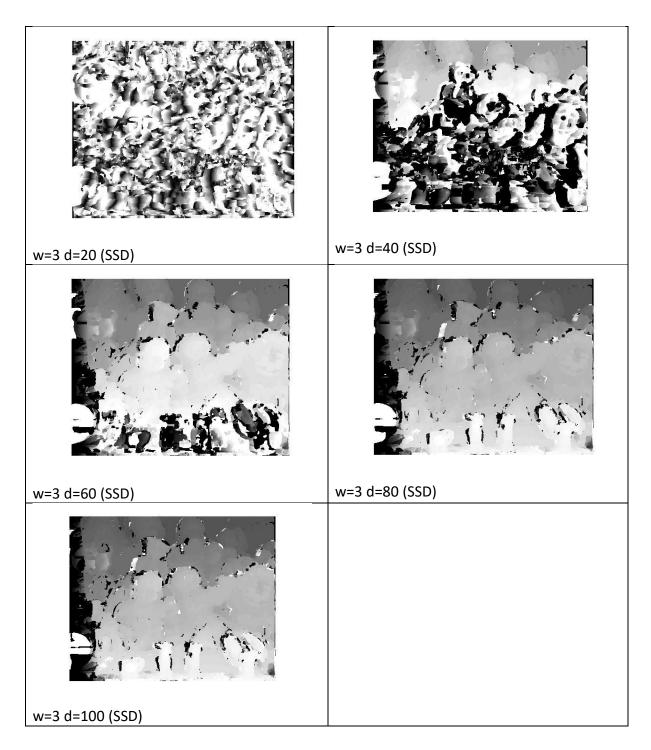


8. Correlation window ve search window' un benzerliklerini ölçmek için sum of square distance, simple cross correlation ve normalized cross correlation metotları kullanılmıştır. Metotlar aşağıda formülize edilmiştir.

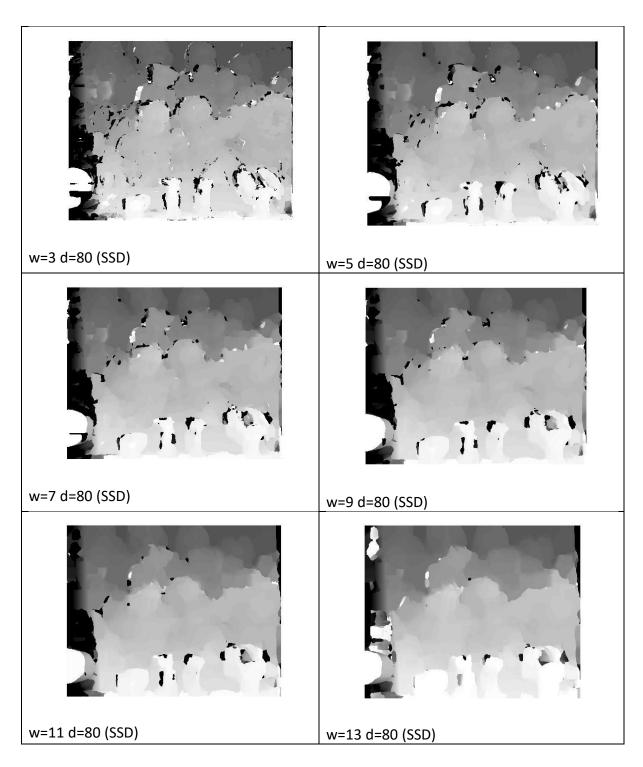
Normalized
$$\frac{\sum\limits_{v=0}^{Rlength}\sum\limits_{u=0}^{Rwidth}R(u,v)\cdot S(c+u,l+v)}{\sqrt{\sum\limits_{v=0}^{Rlength}\sum\limits_{u=0}^{Rwidth}\sum\limits_{v=0}^{Rlength}\sum\limits_{u=0}^{Rwidth}S^{2}(c+u,l+v)}}$$
Simple Cross-Correlation
$$\sum\limits_{v=0}^{Rlength}\sum\limits_{u=0}^{Rwidth}R(u,v)\cdot S\left(c+u,l+v\right)$$
Sum of Squared Differences
$$\sum\limits_{v=0}^{Rlength}\sum\limits_{u=0}^{Rwidth}R(u,v)-S\left(c+u,l+v\right)^{2}$$
Sum of Squared Differences
$$\sum\limits_{v=0}^{Rlength}\sum\limits_{u=0}^{Rwidth}(R\left(u,v\right)-S\left(c+u,l+v\right))^{2}$$

R ve S matrisleri, anlık merkez piksel için, sol görüntüdeki correlation window u ve sağ görüntüdeki search window u temsil etmektedir. u ve v kaydırma miktarlarıdır. Bu projede kameralar aynı y düzleminde bulunduğundan sadece x ekseninde değişim beklenmektedir. Bu nedenle kaydırma işleminde yalnızca v ele alınmıştır.

Deneysel Sonuçlar:

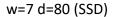


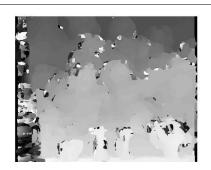
Tablo incelendiğinde search region büyüklüğünün arttırılmasıyla, sum of square distance ile bulunan derinlik daha doğru bir biçimde elde edilmiştir. d 80'den büyük seçildiğinde derinlik görüntüsünden önemsenecek bir değişiklik yaşanmamıştır. Bu nedenle kullanılan görüntü için d=80 olarak belirlenmiştir.



Tablo incelendiğinde window size ın küçük seçilmesi pikselin sağdaki görüntüde bulunmasını zorlaştırmıştır. Pikselin çevresinde bakılan alan arttıkça pikselin sağ görüntüdeki yeri daha doğru bir şekilde saptanmış ve derinlik görüntüsü daha gerçekçi sonuçlar vermiştir. Derinlik görüntülerindeki gürültü azalmıştır. Ancak window size ın arttırılması programın çalışma zamanı açısından dezavantajlıdır.







w=7 d=80 (NCC)



w=7 d=80 (SCC)

	Min	Max	Mean	Std Dev	Time (sec)	W	d
SSD	0	73	5.4668	13.0049	258	7	80
NCC	0	73	4.7957	11.4321	549	7	80
SCC	0	73	10.0486	16.0129	571	7	80

Üç farklı yöntem kullanılarak elde edilen derinlik görüntülerinin gerçek değerlerden farklarına bakılarak hata matrisi hazırlanmıştır. Yukarıdaki tabloda yapılan hatanın maksimum ve minimum değerleri, ortalama hata değeri, hatanın sapma miktarı hesaplanmıştır. Farklı yöntemlerin farlı çalışma süreleri hesaplanmış ve en hızlı çalışan yöntemin SSD olduğu görülmüştür. Ancak hata oranları karşılaştırıldığında NCC yöntemi, diğer iki yönteme göre daha başarılı sonuçlar vermiştir. Daha önce yapılan araştırmalar incelendiğinde NCC yönteminin görüntüler arasındaki ışık-parlaklık farkını dikkate almayarak daha doğru sonuçlar verdiği sistemimizde de görülmüştür. NCC yönteminin dezavantajı, SSD ye göre yaklaşık iki kat uzun sürede çalışmasıdır. Bu üç yöntem arasında hem süre hem ortalama hata miktarı hem de hata sapma miktarı açısından en başarısız yöntem SCC olmuştur.

Program Kodu:

```
% STEREO VISION - DISPARITY COMPUTATION %
clc
clear all
close all
viewL ori=imread('C:\Users\Mukaddes\Dropbox\About Master\Bilgisayarlarla
Görme\zzz Ödev\Odev 3 Proje\viewL.png');
% imshow(viewL ori);
viewL ori = im2double(viewL ori);
viewL = rgb2gray(viewL ori);
viewR ori=imread('C:\Users\Mukaddes\Dropbox\About Master\Bilgisayarlarla
Görme\zzz Ödev\Odev 3 Proje\viewR.png');
% figure, imshow(viewR ori);
viewR ori = im2double(viewR ori);
viewR = rgb2gray(viewR ori);
w = 7; % Correlation window size = w*2+1 %
d = 80; % Search region size = d*2+1 %
```

```
viewR = padarray(viewR,[d d]); % Zero-Padding for image borders %
viewL = padarray(viewL, [d d]);
q = load ('disp.mat');
ground truth = g.L;
% SSD disp mat = uint8(sumOfSquarredDistance(w, d, viewL, viewR));
% tic
% SSD disp mat = uint8(normalizedCrossCorrelation(w, d, viewL, viewR));
% toc
tic
SSD disp mat = uint8(simpleCrossCorrelation(w, d, viewL, viewR));
toc
img = mat2gray(SSD disp mat);
figure
SSD err = double(ground truth-SSD disp mat);
min SSD err = min(SSD err(:))
max SSD err = max(SSD err(:))
mean SSD err = mean2(SSD err)
std SSD err = std(SSD err(:))
% SUM of SQUARRED DISTANCE METHOD %
function [disp mat1] = sumOfSquarredDistance(w, d, viewL, viewR)
[height width] = size(viewL);
% Window Allocation
corr win = zeros(w+1, w+1);
search win = zeros(w+1, w+1);
SSD = 0;
    for i = d+1: height-d
       for j = d+1: width-d
          min = 9999999999;
           % Correlation window coordinates %
           c low row = i - w;
           c high row = i + w;
           c low col= j - w;
           c high col = j + w;
           % Crop correlation window from viewL %
           corr_win = viewL((c_low_row : c_high row) , (c low col :
c_high_col));
           % (search region only x direction) %
           for k = j-d+w : j+d-w
               %Search window coordinates %
```

```
s low row = i - w;
               s high row = i + w;
               s low col = k - w;
               s high col = k + w;
               % Crop search window from viewR %
               search win = viewR((s low row : s high row) , (s low col :
s high col));
               SSD = sum(sum((corr win - search win).^2));
               if SSD < min
                  min = SSD;
                  % Calculate and record disparity %
                  disp mat(i,j) = (j - k);
               end
           end
       end
   end
% Remove padding
disp mat1 = disp mat(d+1 : height-d ,d+1 : width-d);
end
% NORMALIZED CROSS CORRELATION METHOD %
function [disp mat1] = normalizedCrossCorrelation(w, d, viewL, viewR)
[height width] = size(viewL);
% Window Allocation
corr win = zeros(w+1, w+1);
search win = zeros(w+1, w+1);
NCC = 0;
    for i = d+1: height-d
       for j = d+1: width-d
           max = 0;
           % Correlation window coordinates %
           c low row = i - w;
           c high row = i + w;
           c low col= j - w;
           c high col = j + w;
           % Crop correlation window from viewL %
           corr win = viewL((c low row : c high row) , (c low col :
c high col));
           % search region only x direction %
           for k = j-d+w : j+d-w
               %Search window coordinates %
               s low row = i - w;
               s high row = i + w;
               s low col = k - w;
               s high col = k + w;
```

```
% Crop search window from viewR %
               search_win = viewR((s_low_row : s_high_row) , (s_low_col :
s high col));
               corr win = double(corr win);
               search win = double(search win);
               ncc = sum(sum(corr win.*search win));
               lcc = sum(sum(corr win.^2));
               rcc = sum(sum(search win.^2));
               NCC = ncc / sqrt(lcc*rcc);
               if NCC > max
                  max = NCC;
                  % Calculate and record disparity %
                  disp mat(i,j) = abs(j - k);
               end
           end
       end
    end
% Remove padding
disp mat1 = disp mat(d+1 : height-d ,d+1 : width-d);
end
% SIMPLE CROSS CORRELATION METHOD %
function [disp mat1] = simpleCrossCorrelation(w, d, viewL, viewR)
[height width] = size(viewL);
% Window Allocation
corr win = zeros(w+1, w+1);
search win = zeros(w+1, w+1);
SCC = 0;
    for i = d+1: height-d
       for j = d+1: width-d
          max = 0;
           % Correlation window coordinates %
           c low row = i - w;
           c high row = i + w;
           c low col= j - w;
           c high col = j + w;
           % Crop correlation window from viewL %
           corr win = viewL((c low row : c high row) , (c low col :
c high col));
           % (search region only x direction) %
           for k = j-d+w : j+d-w
               %Search window coordinates %
```

```
s low row = i - w;
                s high row = i + w;
               s_low_col = k - w;
               s high col = k + w;
                % Crop search window from viewR %
                search win = viewR((s low row : s high row) , (s low col :
s high col));
               corr_win = double(corr_win);
               search_win = double(search_win);
               mean_corr = mean2(corr win);
               mean search = mean2(search win);
                                sum(sum((corr win-mean corr).*(search win-
               SCC
mean search)));
               if SCC > max
                  max = SCC;
                   \mbox{\%} Calculate and record disparity \mbox{\%}
                   disp mat(i,j) = abs(j - k);
                end
           end
       end
    end
% Remove padding
disp mat1 = disp mat(d+1 : height-d ,d+1 : width-d);
end
```