

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 $(w*2+1 \times w*2+1)$ 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
Cross-Correlation
NCC(c, l)

$$\frac{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R(u, v) \cdot S(c+u, l+v)}{\sqrt{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R^2(u, v) \cdot \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} S^2(c+u, l+v)}}$$

Simple Cross-Correlation
SCC(c, l)

$$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R(u, v) \cdot S(c+u, l+v)$$

Sum of Squared Differences
SSD(c, l)

$$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} (R(u, v) - S(c+u, l+v))^2$$

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.

Deneyisel Sonuçlar:



w=3 d=20 (SSD)



w=3 d=40 (SSD)



w=3 d=60 (SSD)









w=3 d=80 (SSD)



w=3 d=100 (SSD)

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.

	
<p>w=3 d=80 (SSD)</p>	<p>w=5 d=80 (SSD)</p>
	
<p>w=7 d=80 (SSD)</p>	<p>w=9 d=80 (SSD)</p>
	
<p>w=11 d=80 (SSD)</p>	<p>w=13 d=80 (SSD)</p>

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 artırılması programın çalışma zamanı açısından dezavantajlıdır.



w=7 d=80 (SSD)



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 farklı çalışma süreleri hesaplanmış ve en hızlı çalışan yöntemin SSD olduğu görülmüştür. Ancak hata oranları karşılaştırıldığında NCC yöntemi, diğer iki yöntemle 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]);

g = load ('disp.mat');
ground_truth = g.L;

% tic
% SSD_disp_mat = uint8(sumOfSquarredDistance(w, d, viewL, viewR));
% toc

% 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 = 999999999;

            % 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

% 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);

SCC      =      sum(sum((corr_win-mean_corr).*(search_win-
mean_search)));

    if SCC > max
        max = SCC;
        % 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

```