|  |
| --- |
| **YILDIZ TEKNİK ÜNİVERSİTESİ - BİLGİSAYAR MÜHENDİSLİĞİ** |
| **Bilgisayarla Görme Proje Ödevi** |
| **Shape from Stereo** |

|  |
| --- |
| Melike Nur Mermer - 15501010  09.05.2017 |

**Giriş:**

Bu çalışmada 2 kameralı bir sistemin sağ ve sol kamerasından alınan görüntüler arasındaki kayma miktarını ifade eden “Stereo disparity” bulma işlemi gerçekleştirilmiştir. Her pikselin kayma miktarı belirlendikten sonra en çok kayma yakın objelerde en az kayma ise en uzak objelerde olacağından her pikselin kayma miktarlarından her objenin kameraya olan yakınlığını gösteren “Disparity map” oluşturulmuş ve gerçek değerler ile karşılaştırılmıştır. Disparity bulma işlemi için 3 farklı yöntem (Sum of squared distances, Cross correlation, Normalized cross correlation) kullanılmıştır ve her yöntem için minimum, maksimum hata miktarları ile hataların standart sapması hesaplanmıştır.

**Yöntem:**

Elimizde stereo bir sistemin sağ ve sol olmak üzere 2 kamerasından alınan görüntüler mevcuttur ve bu görüntülerin düşeyde hizalanmış olduğu bilinmektedir. Yani görüntülerde sadece yatay eksen yönünde kayma mevcuttur. Her piksel için yatay eksendeki kayma miktarının belirlenmesi amacıyla 3 farklı disparity bulma yöntemi uygulanmıştır. İşlem adımları şu şekilde açıklanabilir:

1. RGB formatındaki görüntüler gri seviyeye ve double değerlere dönüştürülmüştür.
2. Sağdan çekilmiş görüntünün ilk pikselinden başlanarak pikselin etrafında belirlenen pencere boyutu kadarlık kısımdaki pikseller soldan çekilmiş görüntüde aranmak üzere seçilir.
3. Burada w pikselin çevresinden alınacak piksel uzunluğu olmak üzere kenarlardaki piksellerin etrafında oluşturulan pencereler w+1 boyutunda, diğer pikseller 2w+1 boyutundaki pencereler ile karşılaştırılmaktadır.
4. Ground truth değerlerine bakılarak gerçekte pikseller arasında en büyük kaymanın 73 piksel olduğu görülmüştür. Bu nedenle sağdaki görüntüden alınan pencerenin soldaki görüntü üzerinde maksimum eşleşmeyi sağladığı yeri bulmak üzere soldaki görüntüde 75 piksellik bir alan belirlenmiştir. Sağdan alınan pencere sol görüntü üzerinde 75 piksel boyunca yatayda kaydırılarak maksimum eşleşmenin sağlandığı çerçeve bulunmuştur.
5. Bulunan çerçevenin merkezi sağdaki pikselin soldaki görüntüde bulunduğu yeri ifade etmektedir. Soldaki çerçevenin merkezi ile pikselin sağdaki konumunun farkı kayma (disparity) değerini vermektedir.
6. En çok kayma miktarına sahip olan piksellerin kameraya en yakın, daha az kaymaya sahip piksellerin daha uzak olduğu düşünülerek tüm pikseller için bu şekilde belirlenen kayma miktarlarında oluşan değerler kayma haritasını oluşturmaktadır. Kayma haritasında beyaza yakın (yüksek intensitye sahip) objeler kameraya yakın, daha düşük intensitye sahip objeler kameradan uzakta bulunmaktadır.
7. Elde edilen kayma haritası gerçek değerler ile karşılaştırılarak en büyük-en küçük-ortalama hata değerleri ve hataların standart sapması hesaplanmıştır.

**Uygulama:**

Sağ ve sol görüntülerden alınan pencerelerin benzerliklerinin ölçülmesi 3 farklı yöntem kullanılmıştır. Bu yöntemlerin ilki olanfarkların karesinin toplamı yöntemi farklı pencere boyutları için uygulanarak standart sapmaya göre en uygun pencere boyutu belirlenmiştir. 3 farklı kayma miktarı belirleme yöntemi belirlenen pencere boyutu için gerçekleştirilmiş ve sonuçları karşılaştırılmıştır.

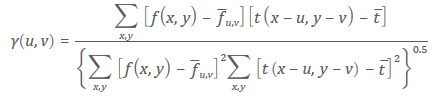
Kayma miktarının bulunmasını sağlayan denklemler ve benzerlik bulmak için bu denklemde kullanılan yöntemler aşağıda verilmektedir.





*Sum of squared distances(minimum için arama yapılır): ψ(u,v)=(u-v)2*

Cross correlation: *ψ(u,v)=u.v*

Normalized cross correlation: 

SSD yönteminin farklı pencere boyutları için uygulanması sonucunda elde edilen değerler aşağıdaki tabloda verilmektedir. Pencere boyutuna bağlı olarak standart sapmanın değişimini gösteren grafik de aşağıda verilmiştir.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pencere boyutu** | **Standart sapma** | **Ortalama hata** | **Minimum hata** | **Maksimum hata** | **Süre(s)** |
| 5 | 12,63 | 2,85 | -75 | 71 | 28,10 |
| 7 | 12,42 | 2,73 | -75 | 72 | 28,59 |
| 9 | 12,39 | 2,74 | -75 | 72 | 29,95 |
| 11 | 12,44 | 2,76 | -75 | 72 | 30,52 |
| 13 | 12,52 | 2,82 | -68 | 72 | 31,29 |

Bu tablodaki değerlerden pencere boyutu büyüdükçe işlem süresinin arttığı görülmektedir. SSD yönteminde standart sapmanın en düşük değeri 9 piksellik pencere boyutu ile elde edilmektedir.

İkinci yöntem olan karşılıklı ilinti için farklı boyutlardaki pencerelerle denemeler yapılarak en iyi eşleşen pikseller bulunmuştur. Elde edilen sonuçlar aşağıdaki tabloda verilmektedir.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pencere boyutu** | **Standart sapma** | **Ortalama hata** | **Minimum hata** | **Maksimum hata** | **Süre(s)** |
| 9 | 13,44 | 3,42 | -75 | 72 | 404,23 |
| 11 | 13,33 | 3,34 | -75 | 72 | 393,84 |
| 13 | 13,33 | 3,31 | -75 | 72 | 426,40 |
| 15 | 13,39 | 3,35 | -75 | 72 | 641,04 |

Tablodan görüldüğü gibi korelasyon işlemi SSD’den çok daha uzun sürmektedir. Bu yöntemin en başarılı olduğu pencere boyutu ise SSD’den farklı olarak 11 pikseldir. 11 piksellik çerçeve ile hem en kısa sürede işlem tamamlanmış hem de en düşük standart sapma elde edilmiştir.

Üçüncü yöntem olan normalleştirilmiş karşılıklı ilinti için de aynı denemeler yapılmıştır, ilgili tablo aşağıda verilmektedir.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pencere boyutu** | **Standart sapma** | **Ortalama hata** | **Minimum hata** | **Maksimum hata** | **Süre(s)** |
| 9 | 10,41 | -0,92 | -75 | 68 | 559,22 |
| 11 | 10,18 | -1,10 | -75 | 67 | 570,28 |
| 13 | 10,11 | -1,25 | -75 | 68 | 574,26 |
| 15 | 10,06 | -1,27 | -75 | 68 | 580,65 |
| 17 | 10,01 | -1,32 | -75 | 68 | 583,37 |

**Sonuç:**

Uygulanan 3 yöntemin en düşük standart sapmayı elde ettiği pencere boyutlarında alınan sonuçlar aşağıdaki tabloda verilmektedir. Bu sonuçlara göre normalleştirilmiş karşılıklı ilinti (NCC) en iyi standart sapmayı elde ederken işlem süresinin de en uzun olduğu yöntemdir. Farkların karesinin toplamı (SSD) yönteminde ise işlem süresi çok kısa olmakla birlikte ikinci en iyi sonuç elde edilmiştir.

Ayrıca her yöntemin farklı pencere boyutlarında en iyi sonucu verdiği görülmüş ve SSD yönteminin küçük pencere boyutlarında daha uygun olduğu anlaşılmıştır. En iyi sonuçları elde eden NCC yöntemi için de pencere boyutu büyük tutulduğunda elde edilen sonuçlar daha iyidir.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Standart sapma** | **Ortalama hata** | **Minimum hata** | **Maksimum hata** | **Süre(s)** |
| **SSD** | 12,39 | 2,74 | -75 | 72 | 29,95 |
| **CC** | 13,33 | 3,34 | -75 | 72 | 393,84 |
| **NCC** | 10,01 | -1,32 | -75 | 68 | 583,37 |

**Program kodları:**

clear all;

left\_im=imread('C:\Users\Melike Nur Mermer\Desktop\Odev2\viewL.png');

right\_im=imread('C:\Users\Melike Nur Mermer\Desktop\Odev2\viewR.png');

left\_im=rgb2gray(left\_im);

right\_im=rgb2gray(right\_im);

left\_d=im2double(left\_im);

right\_d=im2double(right\_im);

w=8;

d1=75;

% left\_d=padarray(left\_d, [0 d1]);%zero-padding

% right\_d=padarray(right\_d, [0 d1]);

[yuk,gen]=size(left\_d);

% disp=zeros(size(left\_d));

% maks=zeros(size(left\_d));

tic

for i=1:yuk

minr = max(1, i-w);

maxr = min(yuk, i+w);

for j=1:gen

% p=left\_d(i,j);

minc = max(1, j-w);

maxc = min(gen, j+w);

mind = 0; %düşey yönde kayma yok

maxd = min(d1, gen-maxc);

maxcorr=0;

normcorrel=0;

norm\_payda=0;

% corr=0;

corr\_mat=right\_d(minr:maxr, minc:maxc);

numBlocks = maxd - mind + 1;

blockDiffs = zeros(numBlocks, 1);

correl=zeros(numBlocks,1);

for d=mind:maxd

wind\_mat=left\_d(minr:maxr, (d+minc):(d+maxc));

blockIndex = d - mind + 1;

% blockDiffs(blockIndex, 1) = sum(sum((corr\_mat-wind\_mat).^2));

normed(blockIndex)=blockDiffs(blockIndex, 1)./norm1;

for k=1:2\*w+1

for l=1:2\*w+1

% correl(blockIndex)=correl(blockIndex)+left\_d(i+k,j+l)\*right\_d(i+k,j+l-d));

normcorrel(blockIndex)=normcorrel(blockIndex)+(left\_d(i+k,j+l)-ort\_wind)\*(right\_d(i+k,j+l+d)-ort\_corr);

norm\_payda(blockIndex)=norm\_payda(blockIndex)+((left\_d(i+k,j+l)-ort\_wind).^2)\*((right\_d(i+k,j+l+d)-ort\_corr).^2);

end

end

normed(blockIndex)=normcorrel(blockIndex)/sqrt(norm\_payda(blockIndex));

end

% [temp, sortedIndeces] = sort(blockDiffs);

[temp, sortedIndeces] = sort(normed,'descend');

bestMatchIndex = sortedIndeces(1, 1);

bm = bestMatchIndex + mind -1;

DbasicSubpixel(i, j) = bm;

end

end

DbasicSubpixel=uint8(DbasicSubpixel);

imshow(DbasicSubpixel);

toc

%error

load('disp.mat');

figure;

imshow(L);

A\_error = double(L)-double(DbasicSubpixel);

min\_err=min(min(A\_error));

max\_err=max(max(A\_error));

ort\_err=mean(mean(A\_error));

std\_err=std2(A\_error);