

Emre Çelik

Augmented Reality – Homework 2

Part 1

1.1

Homography matrisinin hesaplanması için $A \cdot H = 0$ formülü kullanıldı. Her nokta için 2 denklem kullanılarak A matrisi oluşturuldu. Sonrasında A matrisine SVD(Singular Value Decomposition) uygulandı. SVD' den gelen VT matrisinin son satırı alınıp 3x3 olarak homography matrisi elde edildi. (Ref : <http://ros-developer.com/2017/12/26/finding-homography-matrix-using-singular-value-decomposition-and-ransac-in-opencv-and-matlab/>)

Örnek :

Source : { 1, 2 }, { 5, 3 }, { 4, 10 }, { 1, 6 }

Destination : { 1, 2 }, { 5, 3 }, { 4, 10 }, { 1, 6 }

Homography Matrix

0,577350269189627	-3,90925237209274E-16	7,51752063945007E-16
7,3880254445041E-16	0,577350269189626	-1,28726473870844E-15
2,88944740545985E-16	-7,93181640714469E-17	0,577350269189625

1.2

(x,y,1) olarak 3x1 tipinde matrix alan ve bunu homography ile çarpan bir fonksiyon yazıldı. $H(3 \times 3) * Matrix(3 \times 1) = 3 \times 1$ lik (u,v,1) noktası döndürüldü.

1.3

1.2 de ki işlemin aynısı inverse homography matrisi ile yapıldı. $Inv H * (u,v,1)$ yapıp 3x1 lik (x,y,1) noktası elde edildi.

(Unity projesinde Part1 klasörü altında '1.1, 1.2, 1.3' isimli sahnede bu 3 madde implement edilmiş ve konsola basılmıştır.)

(Homomgraphy, Homography.cs isimli dosya içerisinde implement edilmiştir.)

1.4

700x900 lük image üzerinde bazı noktalar referans alınıp, bunlarla eşleşen noktalar diğer resimlerde manuel olarak işaretlendi. Homography sonuçları ölçüm ile kıyaslanıp noktaların origine olan uzaklıklarına göre hata oranı hesaplandı.

(Unity projesinde Part1 klasörü altında ‘1.4, 1.5, 1.6’ isimli sahnede bu 3 image ile ilgili bütün işlemler implement edilip konsola basılmıştır.)

Reference Image : { 100, 100 }, { 300, 200 }, { 100, 400 }, { 200, 500 }, { 200, 300 }

Image 1 : { 755, 706 }, { 1221, 933 }, { 767, 1393 }, { 997, 1624 }, { 994, 1166 }

Image 2 : { 740, 508 }, { 1208, 732 }, { 724, 1208 }, { 960, 1448 }, { 968, 968 }

Image 3 : { 876, 748 }, { 1376, 980 }, { 924, 1460 }, { 1168, 1664 }, { 1148, 1220 }

Homography for Image 1

-0,0031161902494851 -8,16309522819019E-05 -0,738587588814459

0,00011958727221196 -0,00326360939909998 -0,67414094277978

1,05322751235532E-07 -3,85678978309085E-08 -0,00140718318310882

(x,y) : 900 , 100 , 1

(u,v) : 2698,06136091006 , 678,346574781862 , 1

Manuel Ölçüm : { 2612 }, { 681 }, { 1 }

Error : %3,064253

Homography for Image 2

-0,00391577890205559 0,000195775952328927 -0,874011925125165

6,47793614040075E-05 -0,00375565021037435 -0,485871159421452

2,70078388946095E-08 1,4235420264124E-07 -0,00170012468176279

(x,y) : 900 , 100 , 1

(u,v) : 2635,22040897553 , 483,355413928124 , 1

Manuel Ölçüm : { 2612 }, { 476 }, { 1 }

Error : %0,9101557

Homography for Image 3

0,00353929449612746 0,000454734249551114 0,771686579548595

7,50892291020165E-05 0,00355887000796724 0,635981721457835

1,80094033270144E-07 2,66155398354433E-07 0,00129163970211113

(x,y) : 900 , 100 , 1

(u,v) : 2703,78791183207 , 715,679587211247 , 1

Manuel Ölçüm : { 2664 }, { 712 }, { 1 }

Error : %1,428709

1.5

İstenen noktaların karşılıkları bütün image' ler için hesaplanıp konsola basılmıştır.(1.4,1.5,1.6 scene içerisinde)

Projection For Image 1

(x,y) : 7,5 , 5,5 , 1

(u,v) : 542,019811354902 , 491,391471165729 , 1

(x,y) : 6,30000019073486 , 3,29999995231628 , 1

(u,v) : 539,217736360631 , 486,374684520674 , 1

(x,y) : 0,100000001490116 , 0,100000001490116 , 1

(u,v) : 525,099279966921 , 479,296905074305 , 1

1.6

İstenen noktaların karşılıkları bütün image' ler için hesaplanıp konsola basılmıştır.(1.4,1.5,1.6 scene içerisinde)

Inverse Projection For Image 2

(u,v) : 500 , 400 , 1

(x,y) : -4,48213086543912 , 50,8674388181148 , 1

(u,v) : 86 , 167 , 1

(x,y) : -188,397247245151 , -56,4384671081268 , 1

(u,v) : 10 , 10 , 1

(x,y) : -225,231299261579 , -128,663809679767 , 1

Part 2

Resimlerden biri referans seçildi ve bu resim üzerinde 5 nokta(usb' lerin olduğu kısımlar) işaretlendi. Diğer resimlerde de aynı noktalar unity' de yazılan arayüz sayesinde bütün resimlerde işaretlendi. Her resim için, o resim ile referans resim arasında homography hesaplandı. Referans resimde istenilen noktaya uzun bir silindir yerleştirildi. Diğer resimlerde silindiri uygun görebilmek için silindire uygulanacak R,T ve S için hesaplama yapıldı. Referans resmin origin' i ile seçilen noktalardan biri arasındaki uzaklık bulunup originin projeksiyonu ile seçilen noktanın projeksiyonu arasındaki uzaklık bulunup bu uzaklıklar oranlandı ve böylece scale faktörü elde edilmiş oldu. Rotasyon hesabı için yine aynı 2' şer noktalar arasında direction vectoru bulunup bu iki vector arasındaki açı dot product ile bulundu(Unity dot product' ı yapıyor). Bulunan rotasyon açısı kullanılarak silindir, yeni hesaplanan origin eksenini etrafında döndürüldü. Silindirin project ettiği nokta bulunup silindir bu noktaya taşındı. Sonuçlara bakıldığında scale ve pozisyonun uygun olduğu ve rotasyonun çoğu resimde hemen hemen uygun olduğu görüldü. Rotasyon hesabında yapıldığı düşünülen hesap hatası bazı resimlerde açı fazla olduğundan meydana gelmiş olabilir. İmplementason Part2 klasöründe 2 isimli sahnede bulunmaktadır.



