University of Burgundy

Masters in Computer Vision

MSFT Module - SFM + IMU

Report on 4 Points vs 2+1 Points Algorithms

by

Tsagkatakis Ioannis

Under the supervision of

Dr. Cédric Demonceaux

January 13, 2019

1 Introduction

When we are estimating the homography, at least 8 points are needed, because after after normalization the 9 components of the essential matrix is reduced to 8. But if we know some prior information about the motion of the camera like rotations and angles we can decrease the number of points required. In this project i compare the classical linear 4 points algorithm and the linear 2 points, assumming that the vertical direction of the camera is known.

2 The 4 points algorith

When all the points belongs to the same plane, then we need only 4 point. The homography can be computing using the equation(1)

$$\mathbf{H} = \mathbf{R} - \frac{\mathbf{t} * \mathbf{N}^T}{d} \tag{1}$$

where, $\mathbf{H}$ the Homography, $\mathbf{R}$, the rotation between two camera views, $\mathbf{t}$ the translation between two camera views, $\mathbf{N}$ the normal vector of the plane of points and d the distance between the plane and the camera.

The points $\mathbf{p}$ and $\mathbf{p}'$ are colinear so $\mathbf{p} \cong \mathbf{p}'$ and

$$\mathbf{p}' \times \mathbf{H}\mathbf{p} = \mathbf{0} \tag{2}$$

The second equation equation(2) can be used for the verification of the computed Homography.

2.1 Matlab code and results

First 50 are randomly generated in a plane of equation $\mathbf{N}^T\mathbf{X}_w + d = 0$ in the world frame (O_w, X_w, Y_w, Z_w). The camera is calibrated with the rotation R_i and T_i of the world coordinate ($X_w = R_i X_{ci} + T_i$) and $\mathbf{N} = [0, 0, 1]$. The points are shown in figure 1.

The Theoretical Homography $\mathbf{H}$ is calculated

```
N = [0,0,1]';
d = (N'*T1-zposition);
```

Where `d` is the distnace between the 3D points located `zposition` away from the ground plane and the 1st camera. We multiply by the `norm(N)` of

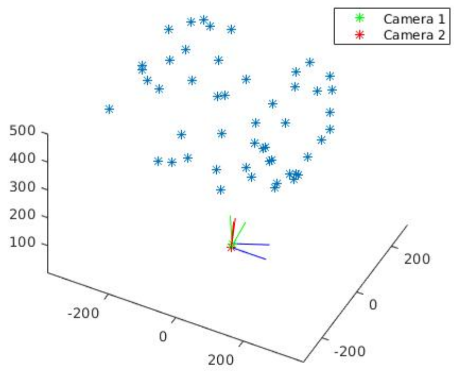

Figure 1: 50 random points located on a plane

the ground plane and and the translation `T1` of camera 1, to get the distance from where the 1st camera is moved in z-direction and then subtract the `zposition` from it.

```
### TEST 0 ##
The Theoretical Homography

H =

    0.8896   -0.4121    0.0270
    0.4111    0.8907    0.0190
   -0.0313    0.0046    1.0000

Estimated Homography (4 Point Algorithm)

H4pt =

    0.8896   -0.4121    0.0270
    0.4111    0.8907    0.0190
   -0.0313    0.0046    1.0000
```

3 The 2 points Algorithm

If the points are on the same plane, and this plane is vertical, then the 4 points can then be reduced to just 2 points. Then the homography is given by equation (3)

$$H = R_y + [t_x, t_y, t_t]^T [n_x, 0, n_z] \tag{3}$$

The (3) is exapnded in only 4 equations and the system is underdetermined. By using SVD:

$$Ah = b \tag{4}$$
$$A = UDV^T \tag{5}$$
$$h = Vy + w\upsilon \tag{6}$$
$$\upsilon = U^T b/D \tag{7}$$

where υ is the last column of vector V and from $\left|H^T H - \mathbf{I}\right| = 0$ we get w as the solution of a 4th order polynomial.

```
Theoretical Homography (2 angles known)

HV2 =

    0.9063   -0.4226    0.0007
    0.4226    0.9063    0.0083
         0         0    1.0208

Verification of theoretical Homography (2 angles known)

ans =

    0.2219
    0.1330
    0.9660


GroundTruth =

    0.2219
    0.1330
```

```
    0.9660

Estimated Homography (2+1 Point Algorithm)

EstimatedH =

    0.8878   -0.4140    0.0007
    0.4140    0.8878    0.0081
         0         0    1.0000

Verification of Yaw Angle

Yaw =

    0.4363


YawGroundTruth =

    0.4363

calculated and Theoretical translational vector

T2E =

   -0.0324
   -0.3700
   -0.9285


T2t =

    0.3486
    3.9848
   10.0000
```

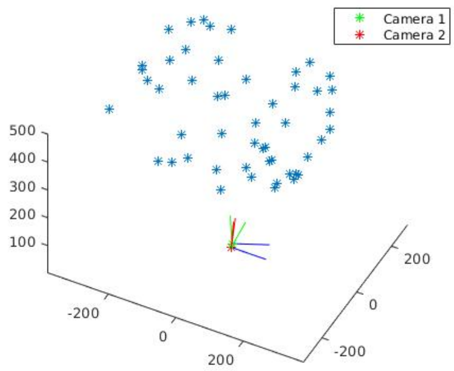

Figure 2: Test1 different data

4 Matlab Tests

4.1 Test1

Example with different datas, propose a test with different positions of the second camera ($R1 = I, T1 = 0$) with angles of rotation between 0° and 45 and translation of 0 to 100. The points are shown in figure 2.

4.1.1 The 4 points algorithm

```
### TEST 1 ##
The Theoretical Homography

The Theoretical Homography

H =

    1.4215    0.2506   -1.4480
    0.7545    1.5987    0.9532
    1.2558   -1.2442    1.0000

Estimated Homography (4 Point Algorithm)

H4pt =

    1.4215    0.2506   -1.4480
```

```
    0.7545    1.5987    0.9532
    1.2558   -1.2442    1.0000
```

4.1.2 The 2 points algorithm

Theoretical Homography (2 angles known)

```
### TEST 1 ##
Theoretical Homography (2 angles known)

HV2 =

    0.9848    0.1736   -0.0232
   -0.1736    0.9848   -0.0162
         0         0    0.9800

True Theoretical Homography (2 angles known)

TrueHomography =

    1.0049    0.1772   -0.0236
   -0.1772    1.0049   -0.0166
         0         0    1.0000

Estimated Homography (2+1 Point Algorithm)

EstimatedH =

    1.0049    0.1772   -0.0236
   -0.1772    1.0049   -0.0166
```

4.2 Test2

Example with noise, propose a test with different camera positions ($R1 = I, T1 = 0$) with angles of rotation between 0 and 45 and translation of 0 to 100 AND white noise in image points of camera 2 between 0 to 1 pixel std (use RANSAC functions).

The camera location and the points are shown in figure 3. Because with 4-Point Algorithm, the solution of AE $= 0$ is a 2-dimentional space, so we will take only x and y in our case.

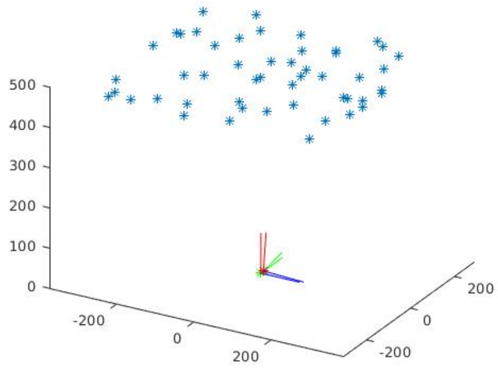

Figure 3: Test points with noise and cameras for test 2

4.2.1 The 4 points algorithm

```
### TEST 2 ##
The Theoretical Homography

H =

    1.0204    0.0893   -0.1072
   -0.0848    1.0233    0.0665
    0.0938   -0.0458    1.0000

Estimated Homography (4 Point Algorithm)

H4pt =

    1.0200    0.0894   -0.1066
   -0.0847    1.0228    0.0669
    0.0937   -0.0457    1.0000

Error(SSD) in 4 point estimation, with noise is : 9.8211e-07
```

4.2.2 The 2 points algorithm

Theoretical Homography (2 angles known)

```
### TEST 2 ##
Theoretical Homography (2 angles known)
```

```
HV2 =

    0.9962    0.0872   -0.0189
   -0.0872    0.9962    0.0137
         0         0    0.9800

True Theoretical Homography (2 angles known)

TrueHomography =

    1.0165    0.0889   -0.0193
   -0.0889    1.0165    0.0140
         0         0    1.0000

Estimated Homography (2+1 Point Algorithm)

EstimatedH =

    1.0204    0.0893   -0.1072
   -0.0848    1.0233    0.0665
    0.0938   -0.0458    1.0000

Error(SSD) in 2 point estimation, with noise is : 0.021467
```

4.3 Test3

Example with noise on IMU informations, propose a test with different camera positions ($R1 = I, T1 = 0$) with angles of rotation between 0 and 45 and translation of 0 to 100 AND white noise in image points of camera 2 between 0 to 1 pixel std AND white noise in IMU between 0 and 2 (use RANSAC functions).

The camera location and the points are shown in figure 4.

4.3.1 The 4 points algorithm

```
### TEST 3 ##
The Theoretical Homography

H =
```

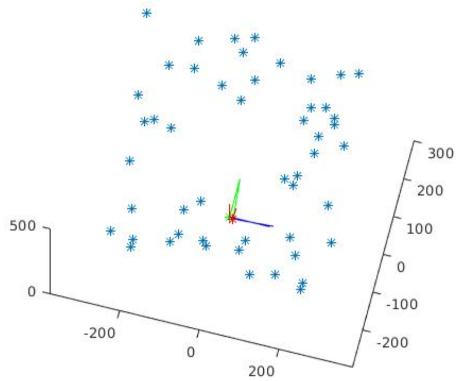

Figure 4: Test points with noise and noise in IMU for test 3

```
    1.0201    0.0952   -0.1095
   -0.0905    1.0231    0.0690
    0.0968   -0.0475    1.0000

Estimated Homography (4 Point Algorithm)

H4pt =

    1.0195    0.0951   -0.1089
   -0.0903    1.0226    0.0695
    0.0971   -0.0470    1.0000

Error(SSD) in 4 point estimation, with noise is : 1.5969e-06
```

4.3.2 The 2 points algorithm

Theoretical Homography (2 angles known)

```
### TEST 3 ##
Theoretical Homography (2 angles known)

HV2 =

    0.9957    0.0930   -0.0188
   -0.0930    0.9957    0.0138
         0         0    0.9800
```

```
True Theoretical Homography (2 angles known)

TrueHomography =

    1.0160    0.0949   -0.0192
   -0.0949    1.0160    0.0141
         0         0    1.0000

Estimated Homography (2+1 Point Algorithm)

EstimatedH =

    1.0201    0.0952   -0.1095
   -0.0905    1.0231    0.0690
    0.0968   -0.0475    1.0000

Error(SSD) in 2 point estimation, with noise is : 0.02289
```