Photogrammetric Computer Vision
WiSe2021

Final Project

Submitted by:

Group F

Naga Padma Veerla : 124268

Harshadeep Gelli : 124303

Zafrul Islam Shaik : 124302

Submitted to:

Professor Volker Rodehorst

Mariya Kaisheva

Christian Benz

Bauhaus University,Weimar

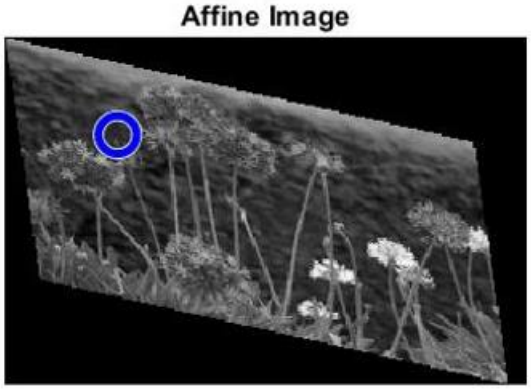
Photogrammetric Computer Vision
WiSe2021

1)

Selected image:

Grey Scale Image:

Affine Image:



The H matrix is arbitrarily chosen for affine transformation

```
H =

    0.8000    0.2000         0
    0.1000    0.7000         0
         0         0    1.0000
```

The above six parameters transform the distorted image **back to its initial state(Original Image)**.

2) Implementation of a MATLAB function LSM.m for a least-squares correlation.

```
%  function for a least squares correlation
function end_value = LSM (start_img,width, height)

% Calculating the Least Squares Correlation

% for initial_img
[x, y] = meshgrid(width:height,width:height);

ref_window = start_img(width:height,width:height);

[fx, fy] = gradient(ref_window,2);

% Obtaining A matrix from LSM Calcuation
A= [sum(sum(fx.^2.*(x.^2))) sum(sum(fx.^2.*(x.*y))) sum(sum(fx.*fy.*(x.*x))) sum(sum(fx.*fy.*(x.*y))) ;
sum(sum(fx.^2.*(x.*y))) sum(sum(fx.^2.*(y.^2))) sum(sum(fx.*fy.*(x.*y))) sum(sum(fx.*fy.*(y.^2)));
    sum(sum(fx.*fy.*(x.^2))) sum(sum(fx.*fy.*(x.*y))) sum(sum(fy.^2.*(x.^2))) sum(sum(fy.^2.*(x.*y)));
    sum(sum(fx.*fy.*(x.*y))) sum(sum(fx.*fy.*(y.^2))) sum(sum(fy.^2.*(x.*y))) sum(sum(fy.^2.*y.^2))];
% B Matrix Calculation
phi = deg2rad(90);
B=[cos(phi) -sin(phi) 0 0; sin(phi) cos(phi) 0 0 ;0 0 0 0;0 0 0 1];

% Obtaining the value of D by using Moore-Penrose Pseudo inverse to matrix A
d = pinv(A).*B;

end_value = [1+d(1), d(2), 0; d(3), 1 + d(4), 0; 0, 0, 1];
```

Photogrammetric Computer Vision
WiSe2021

3)

When we run the files "**AFFINE.m**" with "**LSM.m**"

Qualitative analysis:

The image that we got at the 72nd iteration is similar to the original image
But it does not have the exact quality.

Quantitative analysis:

There is a change in the deformation parameters compared to the initial parameters though some of the parameters are the same.

```
% for initial_img
[x, y] = meshgrid(width:height,width:height);

ref_window = start_img(width:height,width:height);

[fx, fy] = gradient(ref_window,2);
```

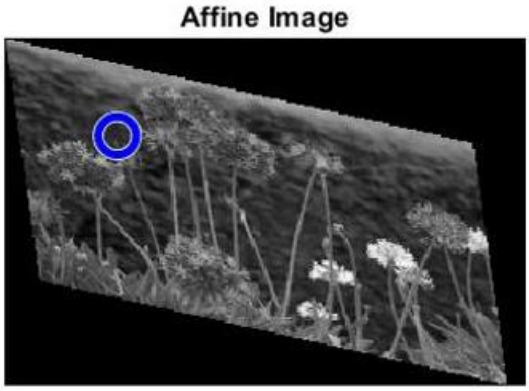
- The grid size for the start image is taken as (width: height, width: height)
- The Reference window for the start image is (width: height, width: height).
- The search window for the start image is (width: height, width: height).

From here, we get the affined start image which looks almost equivalent to the final image after performing multiple iterations.

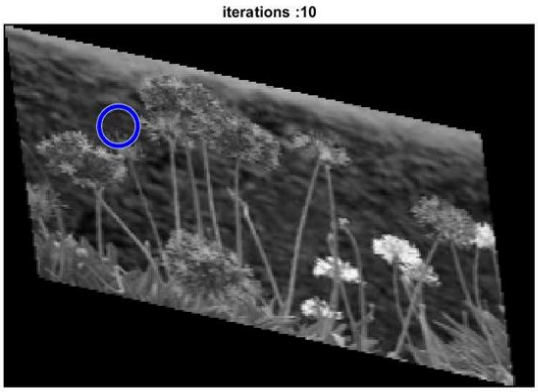
Photogrammetric Computer Vision
WiSe2021

Original Image:

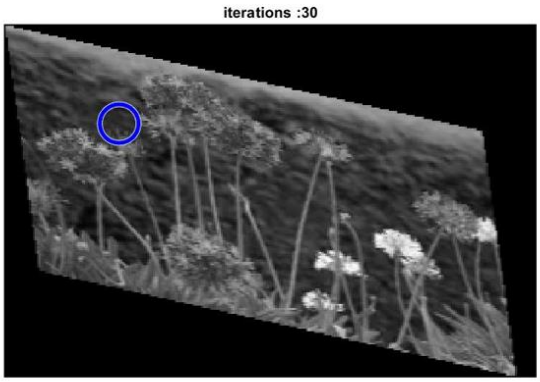
Affine Image:



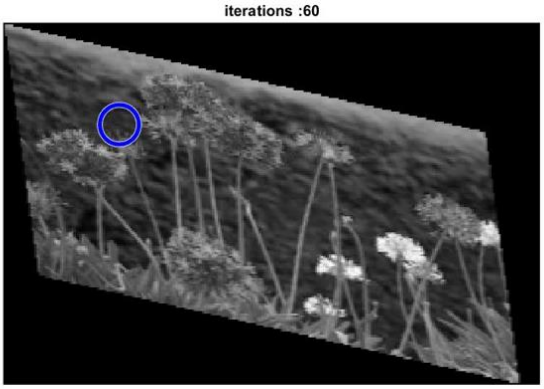
Iteration 10:



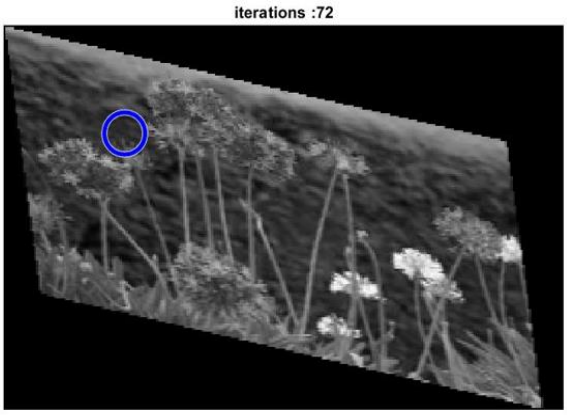
Iteration 30:



Iteration 60:

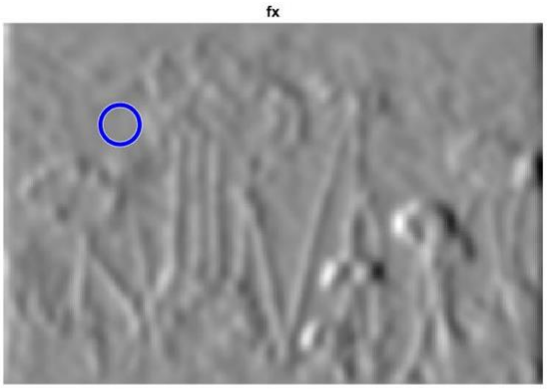


Iteration 72:

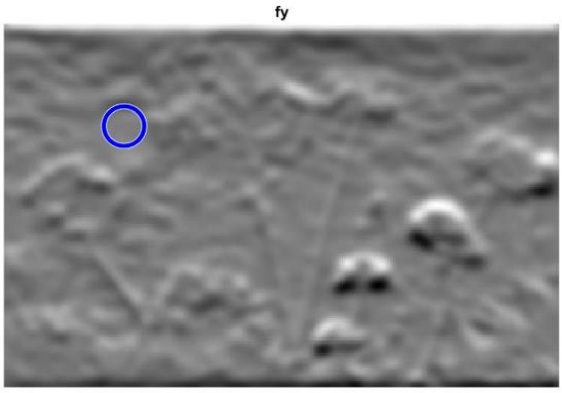


Photogrammetric Computer Vision
WiSe2021

Horizontal greyscale value:



Vertical greyscale value:



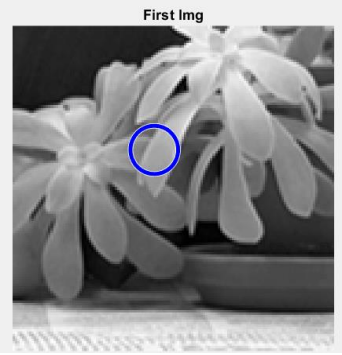
Photogrammetric Computer Vision
WiSe2021

4)

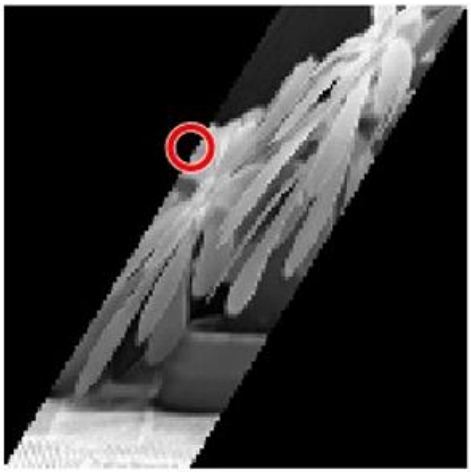
```
% To read the three distorted images
A_distorted_image = double (imread('distorted_img_A.png'));
B_distorted_image = double (imread('distorted_img_B.png'));
C_distorted_image = double (imread('distorted_img_C.png'));
```

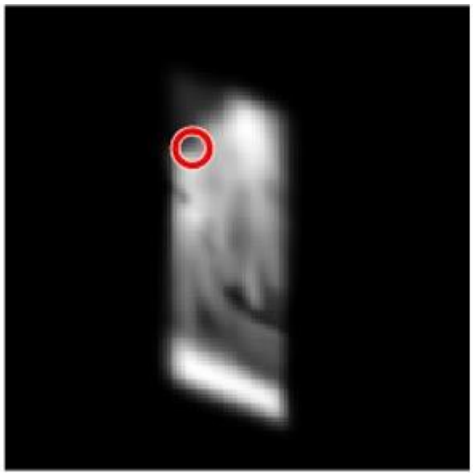
The number of iterations we did for

1. for distorted image A is 46 (n=46),
2. for distorted image B is 1 (n=1)
3. for distorted image C is 54 (n=54).



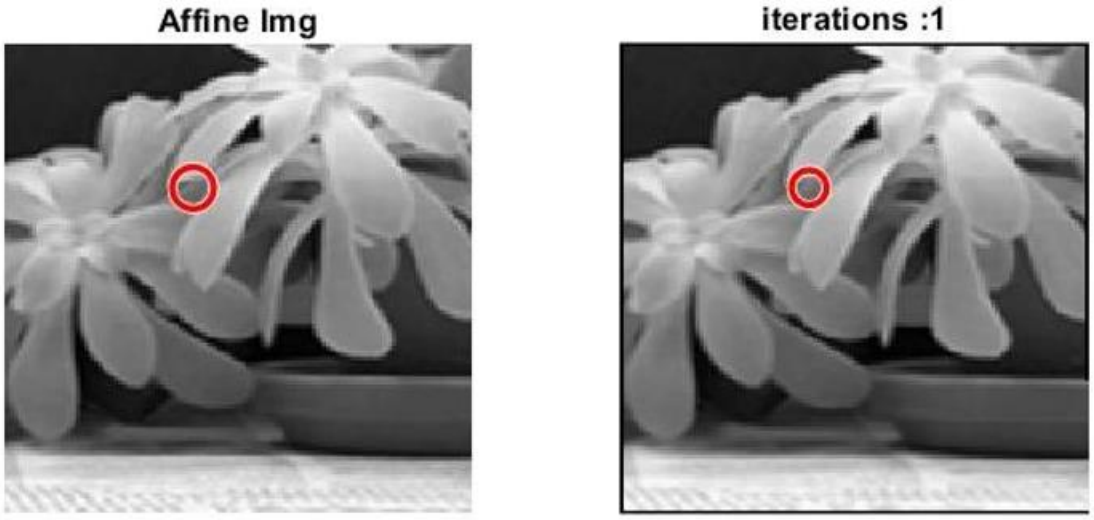
Distorted Image A:
iterations = 46





We can observe from the above images that, Distorted image A is not transformed totally and the number of iterations shown above is the nearest image to the target image after 46 iterations.

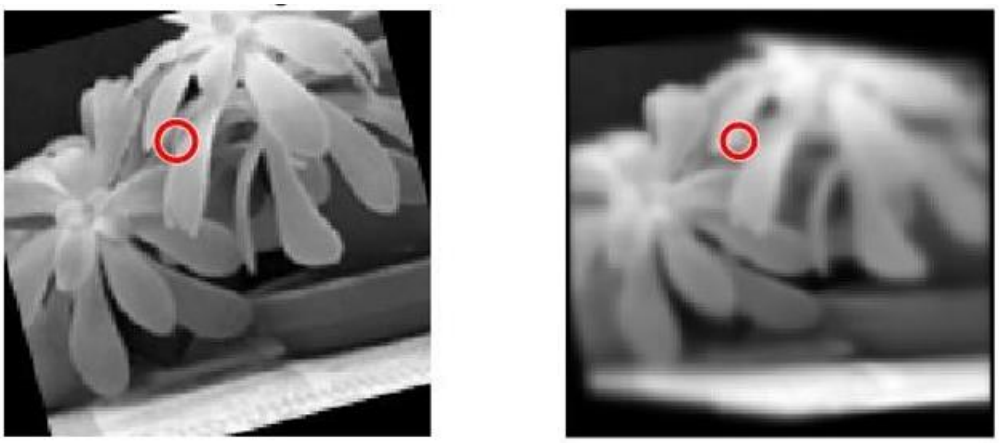
Distorted Image B:
Iterations = 1



We can observe from the above images that, Distorted image B is transformed totally and the number of iterations shown above is the image to achieve the target image after 1 iteration.

Distorted Image C:

Iterations = 54



We can observe from the above images that, Distorted image C is not transformed totally and the number of iterations shown above is the nearest image to the target image after 54 iterations. We can see some noise in the image.

The results are acceptable for Distorted images A, B, and C. The iterations count has varied for all the 3 distorted images and they are:

For Distorted image, A is 50 Iterations

for Distorted image, B is 1

Iterations number for Distorted image C is 54.

We have observed that,

- When there is an increase in mean difference, the noise level increases and the affine transformation is not in complement with the target image.
- When there is a small change in the mean difference, we can notice minimum distortions.

Photogrammetric Computer Vision
WiSe2021

5)

Least Squares Matching (LSM):

1. It is the method used when Sum of squared differences, Normalized cross-correlation are unsuitable for geometric deformations in close range.
2. The idea of LSM is to Compensate the differences of the distortions before the comparison
3. The task is to Search a transformation, which bring two stereo image pattern as well as possible in agreement
4. We estimate the 6 parameters i.e, 4 deformation parameters of D and 2 for the translation d

5. Then build a TAYLOR linearized inhomogeneous equation system Az=b for centered nxn image window.

$$\mathbf{A} = \left[\begin{array}{cccc:cc} \Sigma f_x^2x^2 & \Sigma f_x^2xy & \Sigma f_xf_yx^2 & \Sigma f_xf_yxy & \Sigma f_x^2x & \Sigma f_xf_yx \\ \Sigma f_x^2xy & \Sigma f_x^2y^2 & \Sigma f_xf_yxy & \Sigma f_xf_yy^2 & \Sigma f_x^2y & \Sigma f_xf_yy \\ \Sigma f_xf_yx^2 & \Sigma f_xf_yxy & \Sigma f_y^2x^2 & \Sigma f_y^2xy & \Sigma f_xf_yx & \Sigma f_y^2x \\ \Sigma f_xf_yxy & \Sigma f_xf_yy^2 & \Sigma f_y^2xy & \Sigma f_y^2y^2 & \Sigma f_xf_yy & \Sigma f_y^2y \\ \hdashline \Sigma f_x^2x & \Sigma f_x^2y & \Sigma f_xf_yx & \Sigma f_xf_yy & \Sigma f_x^2 & \Sigma f_xf_y \\ \Sigma f_xf_yx & \Sigma f_xf_yy & \Sigma f_y^2x & \Sigma f_y^2y & \Sigma f_xf_y & \Sigma f_y^2 \end{array}\right]$$

$$\mathbf{b} = (f-g)\cdot\left(\Sigma f_x x, \Sigma f_x y, \Sigma f_y x, \Sigma f_y y \,\vdots, \Sigma f_x, \Sigma f_y\right)^T$$

$$\mathbf{z} = (d_1, \ldots, d_4 \,\vdots, d_5, d_6)^T$$

6. A is the design matrix and Z consists of variables d1,…,d6.

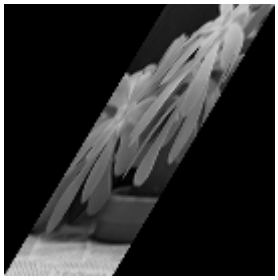
7. The intensity difference between the reference window and the search window gives the pixel position that is in relation to each other at a point of iteration.

Photogrammetric Computer Vision
WiSe2021

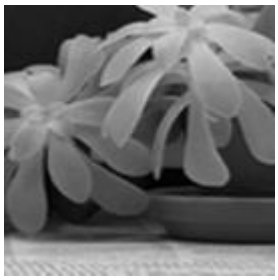
6)

- d1: Rotation
- d2: Shear and Rotation
- d3: Rotation
- d5 and d6: Translation of image.

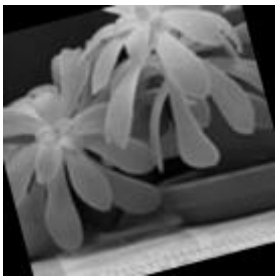
The distortion image A is **sheared** compared to the target image.



The distortion image B is **Scaled** compared to the target image.



The distortion image C is **Rotated and Translated** compared to the target image.



Photogrammetric Computer Vision
WiSe2021

References:

1. Photogrammetric Computer Vision Lecture notes - Bauhaus University, Weimar.
2. https://github.com/zxyctn/Least-Squares-Correlation
3. (Image source Link) https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.ricoh-imaging.co.jp%2Fenglish%2Fproducts%2Flens%2Fk%2Ftelephoto%2Fhdpentax-dfa-150-450%2F&psig=AOvVaw2uLRmPY-pAYRUrmtW1Bu9g&ust=1647248195429000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCIj5y_DbwvYCFQAAAAAdAAAAABAK