Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\F\_archive\easystore\Python\_Programs\panorama-stitching\proximity\_stitcher.py

Please Browse for first image to be stitched

I got 738884 bytes from this file.

Please Browse for second image to be stitched

I got 754095 bytes from this file.

Please select directory to hold pickle file:

You chose F:/Adams\_Farm/Rescue\_2/some\_images\_reformatted

pickle file F:/Adams\_Farm/Rescue\_2/some\_images\_reformattedrespinse\_surface\_plot.pickle

Enter altitude\_bias for left image: 320.

Enter altitude\_bias for right image: 320.

offset distance between images = [-0.5573636772600101, 24.96302048998641] feet

pixels size A (feet) 0.00988324316051075

pixels size B(feet) 0.00988324316051075

Mb

[[ 1.00e+00 0.00e+00 2.58e+03]

[ 0.00e+00 1.00e+00 -3.46e+02]

[ 0.00e+00 0.00e+00 1.00e+00]]

&\*&\* Best e\_loss &\*&\* 0.839386817578747

&\*&\* Best Mb &\*&\*

[[ 1.00e+00 0.00e+00 2.88e+03]

[ 0.00e+00 1.00e+00 -4.60e+01]

[ 0.00e+00 0.00e+00 1.00e+00]]

lenght of e\_losses 40

e\_losses:

[[0.93840719 0.96295264 1.00145505 ... 1.54873777 1.53345379 1.5171806 ]

[0.97119306 0.99584168 1.03432195 ... 1.56520373 1.55016121 1.5337469 ]

[0.96847226 0.99304346 1.03141925 ... 1.56315405 1.5481173 1.53152473]

...

[0.84466069 0.87476774 0.91983949 ... 1.50373642 1.48811197 1.47237549]

[0.84175218 0.87202692 0.91711706 ... 1.50301945 1.48808212 1.47239831]

[0.83938682 0.86967694 0.91482915 ... 1.50021631 1.48513683 1.46897964]]

Reset Mb

[[1.00000000e+00 0.00000000e+00 2.52579240e+03]

[0.00000000e+00 1.00000000e+00 5.63948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Enter maximum number of epochs: 4

e\_loss 1.0772560504952344

grad

[[0. 0. 0.00493836]

[0. 0. 0.00522619]

[0. 0. 0. ]]

Before update self.Mb[ 0 ]

[[1.00000000e+00 0.00000000e+00 2.53079240e+03]

[0.00000000e+00 1.00000000e+00 6.13948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

After update self.Mb[ 0 ]

[[1.00000000e+00 0.00000000e+00 2.53074302e+03]

[0.00000000e+00 1.00000000e+00 6.13425539e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

e\_loss 1.1033869844061233

grad

[[0. 0. 0.00360437]

[0. 0. 0.00422004]

[0. 0. 0. ]]

self.Mb\_last

[[1.00000000e+00 0.00000000e+00 2.53579240e+03]

[0.00000000e+00 1.00000000e+00 6.63948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

grad\_last

[[0. 0. 0.00360437]

[0. 0. 0.00422004]

[0. 0. 0. ]]

denominator 0.0

Mb\_Mb\_last

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

dot product

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

norm 0.0

(grad-grad\_last) [[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

Before update self.Mb[ 1 ]

[[1.00000000e+00 0.00000000e+00 2.53579240e+03]

[0.00000000e+00 1.00000000e+00 6.63948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

After update self.Mb[ 1 ]

[[1.00000000e+00 0.00000000e+00 2.53575636e+03]

[0.00000000e+00 1.00000000e+00 6.63526153e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

e\_loss 1.12448720066332

grad

[[0. 0. 0.00236098]

[0. 0. 0.00313506]

[0. 0. 0. ]]

self.Mb\_last

[[1.00000000e+00 0.00000000e+00 2.54079240e+03]

[0.00000000e+00 1.00000000e+00 7.13948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

grad\_last

[[0. 0. 0.00236098]

[0. 0. 0.00313506]

[0. 0. 0. ]]

denominator 0.0

Mb\_Mb\_last

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

dot product

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

norm 0.0

(grad-grad\_last) [[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

Before update self.Mb[ 2 ]

[[1.00000000e+00 0.00000000e+00 2.54079240e+03]

[0.00000000e+00 1.00000000e+00 7.13948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

After update self.Mb[ 2 ]

[[1.00000000e+00 0.00000000e+00 2.54076879e+03]

[0.00000000e+00 1.00000000e+00 7.13634652e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

e\_loss 1.1401624825887768

grad

[[0. 0. 0.0020837 ]

[0. 0. 0.00290146]

[0. 0. 0. ]]

self.Mb\_last

[[1.00000000e+00 0.00000000e+00 2.54579240e+03]

[0.00000000e+00 1.00000000e+00 7.63948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

grad\_last

[[0. 0. 0.0020837 ]

[0. 0. 0.00290146]

[0. 0. 0. ]]

denominator 0.0

Mb\_Mb\_last

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

dot product

[[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

norm 0.0

(grad-grad\_last) [[0. 0. 0.]

[0. 0. 0.]

[0. 0. 0.]]

Before update self.Mb[ 3 ]

[[1.00000000e+00 0.00000000e+00 2.54579240e+03]

[0.00000000e+00 1.00000000e+00 7.63948158e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

After update self.Mb[ 3 ]

[[1.00000000e+00 0.00000000e+00 2.54577157e+03]

[0.00000000e+00 1.00000000e+00 7.63658011e+01]

[0.00000000e+00 0.00000000e+00 1.00000000e+00]]

X\_gd:

[2525.79240281 2530.74301925 2535.75635908 2540.76879299]

Y\_gd:

[56.39481577 61.3425539 66.35261534 71.36346521]

Z\_gd:

[1.10338698 1.1244872 1.14016248 1.1546698 ]

SSE\_loss 5210672.571356771

Do you want to show solution (y/n) y

Enter any key to continue:

do you want to save cropped images (y/n)? y

>>>

======== RESTART: F:\Python\_Programs\panorama-stitching\my\_stitch.py ========

Please Browse for first image to be stitched

I got 738884 bytes from this file.

Please Browse for second image to be stitched

I got 754095 bytes from this file.

H

[[ 1.06154723e+00 1.55683990e-02 2.18492915e+03]

[-1.06754542e-02 1.03714118e+00 -1.99607098e+02]

[-4.83894518e-07 -7.92025866e-07 1.00000000e+00]]

Do you waant to show input images (y/n) y

Do you waant to show matches (y/n) y

>>>