Project 2: Panorama Mosaic Stitching

Wenzhen Zhu

Oct 5th, 2016

Mathematica Script to Import and Resize

Import

In[332]:=
```
dir = NotebookDirectory[];
```

In[382]:=
```
fullDir1 = dir <> "campus_00" <> ToString[#] <> ".JPG" & /@ (Range[10] - 1)
fullDir2 = dir <> "campus_0" <> ToString[#] <> ".JPG" & /@ (9 + Range[7])
```

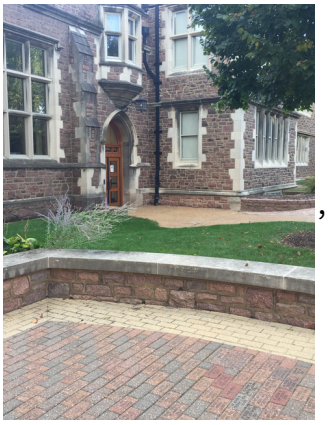
```
{/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_000.JPG,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_001.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_002.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_003.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_004.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_005.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_006.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_007.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_008.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_009.JPG}
{/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_010.JPG,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_011.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_012.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_013.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_014.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_015.JPG
 ,
 /Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_016.JPG
 }
```

```
part1 = Import[#] & /@ fullDir1;
part2 = Import[#] & /@ fullDir2;
imgList = Join[part1, part2];
```

Resize

```
resized = ImageResize[#, {480, 640}] & /@ imgList
```

Out[338]= {  ,  ,

 ,  ,  ,

 ,  ,  ,


,

,

,

,

,

,

,

,

}

Export

In[347]:=
```
MapThread[Export[dir <> "campus_00" <> ToString@#1 <> ".tga", #2] &, {(Range[10] - 1), resized[[1 ;; 10]]}]
MapThread[Export[dir <> "campus_0" <> ToString@#1 <> ".tga", #2] &, {(9 + Range[7]), resized[[11 ;; 17]]}]
```

Out[348]= {/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_010.tga,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_011.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_012.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_013.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_014.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_015.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_016.tga
}

Out[347]= {/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_000.tga,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_001.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_002.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_003.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_004.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_005.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_006.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_007.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_008.tga
,
/Users/wenzhen/Dropbox/2_Society/WUSTL/cse559a/projects/PackageProject_2/test_sets/myTest4/campus_009.tga
}

To Do

Steps

1. Take pictures on a tripod (or handheld)
2. Warp to spherical coordinates (Panorama.exe)
3. Extract features (Features.exe)
4. Match features (Features.exe)
5. Align neighboring pairs using RANSAC (Panorama.exe)
6. Write out list of neighboring translations (Panorama.exe)
7. Correct for drift (Panorama.exe)
8. Read in warped images and blend them (Panorama.exe)
9. Crop the result and import into a viewer

Compute features in the warped images

I used SIFT features by SIFT Keypoint Detector.

Out[359]= http://www.cs.ubc.ca/~lowe/keypoints/

Warp each image into spherical coordinates. (file: WarpSpherical.cpp, routine: warpSphericalField)

Compute the inverse map to warp the image by filling in the skeleton code in the warpSphericalField routine to:
Convert the given spherical image coordinate into the corresponding planar image coordinate using the coordinate transformation equation from the lecture notes
Apply radial distortion using the equation from the lecture notes

Devices

I used iPhone 6 Plus. From the photo info, I got $d = 4.15$. From online resource, I got $w_s = 4.8$ mm, image is resized to

Sensor size Information about the dimensions of the image sensor used in the device. Usually cameras with larger sensors and less pixel density tend to provide better image quality despite the lower resolution.	4.8 x 3.6 mm (millimeters) 0.24 in (inches)

In[343]:=
```
d = 4.15;
wp = 480;
ws = 4.8;
f = d wp/ws
```

Out[346]=
```
415.
```

I applied radio distortion.

Left image is the resized original image. Right image is the warped to spherical coordinate without radial distortion.

Out[379]=

Here is a comparison. Left is warped image without radial distortion and right is with radial distortion.

Out[381]=

Compute the alignment of two images.
(file: FeatureAlign.cpp, routines: alignPair, countInliers, and leastSquaresFit)

To do this, you will have to implement a feature-based translational motion estimation. The skeleton for this code is provided in FeatureAlign.cpp. The main routines that you will be implementing are:

```
int alignPair(const FeatureSet &f1, const FeatureSet &f2, const vector &matches,
MotionModel m, float f, int nRANSAC, double RANSACthresh, CTransform3x3& M);
```

```
int countInliers(const FeatureSet &f1, const FeatureSet &f2, const vector &matches,
MotionModel m, float f, CTransform3x3 M, double RANSACthresh, vector &inliers);

int leastSquaresFit(const FeatureSet &f1, const FeatureSet &f2, const vector &matches,
MotionModel m, float f, const vector &inliers, CTransform3x3& M);
```

Stitch and crop the resulting aligned images.

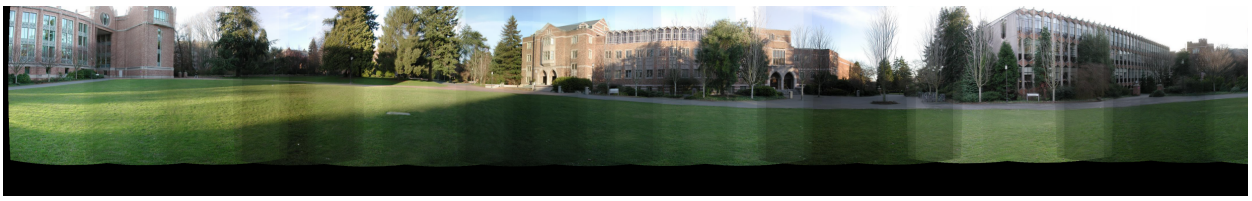
Given the warped images and their relative displacements, figure out how large the final stitched image will be and their absolute displacements in the panorama
Then, resample each image to its final location and blend it with its neighbors (AccumulateBlend, NormalizeBlend). Try a simple feathering function as your weighting function (see mosaics lecture slide on “feathering”) (this is a simple 1-D version of the distance map described in Szeliski & Shum ‘97). For extra credit, you can try other blending functions or figure out some way to compensate for exposure differences. In NormalizeBlend, remember to set the alpha channel of the resultant panorama to opaque!
Crop the resulting image to make the left and right edges seam perfectly (BlendImages). The horizontal extent can be computed in the previous blending routine since the first image occurs at both the left and right end of the stitched sequence (draw the “cut” line halfway through this image). Use a linear warp to the mosaic to remove any vertical “drift” between the first and last image. This warp, of the form $y' = y + ax$, should transform the y coordinates of the mosaic such that the first image has the same y-coordinate on both the left and right end. Calculate the value of a needed to perform this transformation.

For feathering, I only did naively piecewise horizontal hat function

```
from 0 to blendWidth/2-1: accending
from (blendWidth/2) to (sh.width-blendWidth/2-1): flat
from (sh.width-blendWidth/2) to (sh.width-1): descending
```

Demo

Campus



What makes a horrible panorama

1. drifting too much
2. don’t have 50% overlapping, this will result blurry part.
3. Person move in adjacent pictures, this will also result blurry part.

A good panorama

Extra credit

1. I did support radial distortion correction.
2. Try a sequence in which the same person appears multiple times, as in this example.