# A04 Pano

By Kaden Ramirez

# Input Images

Image1 (left)

Image2 (middle)

Image1 (right)

# Code

```
def matchMe(img1,img2):
        orb = cv2.ORB_create()
        kp1 = orb.detect(img1)
        kp2 = orb.detect(img2)
        kp1,des1 = orb.compute(img1,kp1)
        kp2,des2 = orb.compute(img2,kp2)
        bf = cv2.BFMatcher()
        matches = bf.knnMatch(des1,des2,k=2)
        good = []
        for m,n in matches:
                if m.distance < 0.9*n.distance:
                        good.append(m)
        for m in good:
                pt1=kp1[m.queryIdx].pt
                pt2=kp2[m.trainIdx].pt
                points1.append(pt1)
                points2.append(pt2)
```

```
def homographyFind(img1,img2,points1,points2):
        H,_=cv2.findHomography(np.float32(points2), np.float32(points1),
                cv2.RANSAC, 5.0)
        HI=np.linalg.inv(H)
```

```
def eroder(img1,img2,mask,transition_zone=129):
        show(mask)
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,
                (transition_zone,transition_zone))
        mask = cv2.erode(mask,kernel,iterations = 1)
```

```
def blurMask(out1,out2,mask,transition_zone=129):
        mask=cv2.blur(mask,(transition_zone,transition_zone))
        out=mask/255*out1+(1-mask/255)*out2
```

# Code Cont… (Pano Translation)

```
h,w,=img1.shape[:2]
points=np.float64([[[0,0],[w-1,0],[0,h-1],[w-1,h-1]]])
points=np.hstack((points,cv2.perspectiveTransform(points, H)))

minx=np.min(points[0,:,0])
maxx=np.max(points[0,:,0])
miny=np.min(points[0,:,1])
maxy=np.max(points[0,:,1])
target_size=(int(maxx-minx),int(maxy-miny))
translate=np.float64([[1,0,-minx],[0,1,-miny],[0,0,1]])
out1=cv2.warpPerspective(img2,translate.dot(H),target_size)
mask=cv2.warpPerspective(np.uint8(img2*0+255),translate.dot(H),target_size)
out2=cv2.warpPerspective(img1,translate,target_size)
```

# First Pano with Matches

Matches

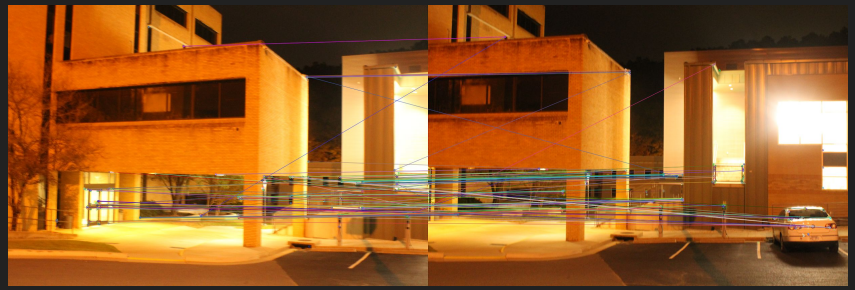


Image1 (left)

Image2 (middle)

Image1 and Image2 Pano

# Second Pano with Matches

Matches

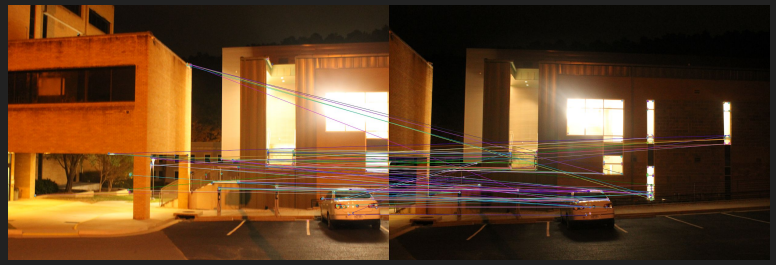


Image2 (middle)

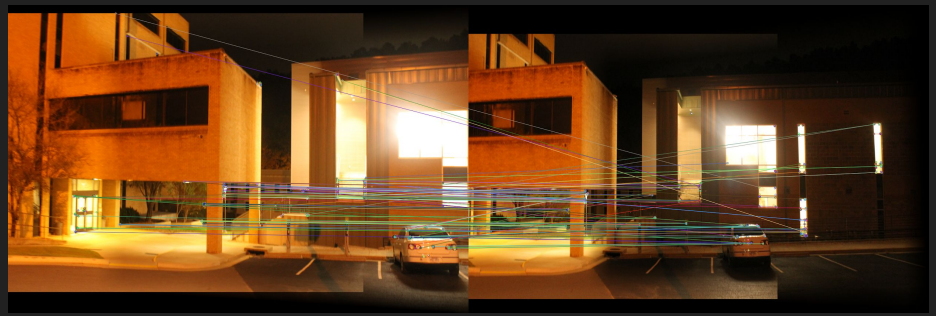
Image1 (right)

Image2 and Image3 Pano

# Final Pano With Matches and Cropped

Cropped Pano

Matches



Pano