CS6476 Computer Vision

Assignment – Image Stitching (Extra credit)

In this assignment, you'll implement an image stitching algorithm.

Objective:

The objective of this assignment is to explore and implement advanced techniques in image stitching to create seamless panoramic images from a set of overlapping images. You will need to learn and understand key concepts such as correspondence estimation, homography estimation, and image stitching algorithms to achieve this goal.

Understanding Image Stitching:

1.) Correspondence estimation or keypoint matching:

Correspondence estimation, also known as keypoint matching involves identifying distinctive points or features in different images that correspond to the same physical point in the scene. These keypoints serve as reference points for aligning and blending the images together.

There are many feature detectors and descriptors available like SIFT, SURF, ORB, BRISK, etc. You can choose any of them for this assignment based on their properties.

Resources:

- Blog Post: "A Gentle Introduction to Keypoint Detection and Matching" by PyImageSearch. [Link]
- Paper: "SIFT: Scale-Invariant Feature Transform" by David G. Lowe. This paper introduces the Scale-Invariant Feature Transform (SIFT) algorithm, one of the most widely used techniques for keypoint detection and matching. [Link]

2.) Homography estimation:

Homography estimation is a critical step in image stitching, where we calculate a transformation matrix (homography matrix) that describes the geometric relationship between two images. It allows us to align different images so that they can be seamlessly stitched together to create a panoramic view.

Resource:

- Blog Post: "Homography Estimation" by OpenCV. This detailed blog post explains the concept of homography estimation and provides code examples using OpenCV. [Link]

- Paper: "Homography Estimation" by Richard Hartley and Andrew Zisserman. Chapter 2 of the book "Multiple View Geometry in Computer Vision" provides an in-depth explanation of homography estimation. [Link]

3.) Stitching image using estimated homography:

Once the homography matrix is estimated, it can be used to warp and align images. This alignment is essential for stitching images together to create a seamless panoramic view. Image stitching algorithms apply techniques like blending to ensure a smooth transition between the overlapping regions.

Resources:

- Paper: "Automatic Panoramic Image Stitching using Invariant Features" by Matthew Brown and David G. Lowe. This paper discusses the concept of automatic panoramic image stitching using invariant features and provides insights into the stitching process. [Link]

Assignment Instructions:

1.) Implement correspondence/keypoint matching between 2 images.

Left Image

Right Image

Source: Unknown

Head over to **stitch.py** and write your code in **get_correspondence()**

Note:

- We have provided a visualization function - **visualize_keypoints_and_correspondences()** to visualize your matched keypoints. Feel free to edit or write your own function to incorporate other visualization functionalities if necessary.
- You can use any of the above-mentioned feature detector/descriptor and use of inbuilt openCV functions is allowed.

2.) Compute homography.

Complete the **get_homography()** in **stitch.py**

Note: Write your own function to compute homography and **do not** use inbuilt openCV functions to compute homography.

3.) Stitching both the images using the estimated homography

Complete **stitch()** in **stitch.py**

Run **stitch.py** to generate your stitched image - **output.jpg**

Note:

- All the above functions are already called in the main(). Feel free to edit/change anything if necessary.
- **Do not** use openCV stitcher class to stitch the image. Write your own stitch(). Points **will not be awarded** if you use inbuilt stitcher() class.

4.) Analyze variation to properties (rotation, scale and illumination) of your descriptor. Comment on its applicability in a real-world setting.

- **Rotation:** Rotate your right image by 30 degrees from 0 to 360 and match keypoints with the rotated right image and the original left image. Include qualitative results of matched keypoints b/w rotated right and left image, plot the histogram of no. of correspondences VS angle, and comment if its variant or invariant.

Note: Include qualitative result of matched keypoints b/w rotated right and left image along with the histogram plot (no. Of correspondences vs angle), for each rotation. Comment if its variant or invariant.

- **Scale:** Scale your right image by downsampling or upsampling, and report how the no. of correspondences vary qualitatively and quantitatively between the scaled right image and original left image. Try atleast 3-4 scaling factors, and comment if its variant or invariant.

Note: Include qualitative result of matched keypoints b/w scaled right image and original left image along with their no. Of correspondences, for every scaling factor. Comment if its variant or invariant.

- **Illumination:** Increase/Decrease the brightness/illumination of the right image, and report how the no. of correspondences vary qualitatively and quantitatively between the modified right and original left image. Try atleast 3-4 different illumination levels, and comment if its variant or invariant.

Note: Include qualitative result of matched keypoints b/w modified right image and original left image as well as their no. Of correspondences, for every illumination level. Comment if its variant or invariant.

- Based on the above 3 properties, comment on the applicability of your descriptor in a real-world setting.

Grading Schema:

- Qualitative result of keypoint matching (use visualize_keypoints_and_correspondences() or your own visualization function) - [1 point]
- Homography estimation [3 points]
- Qualitative result of stitched image – [3 points]
- Analyze properties (rotation, scale, illumination) of your descriptor based on the above instructions [3 points – 1 points per each property]

Submission:

- Submit your code – **stitch.py**
- Submit a PDF describing your results
    - Qualitative result - matched keypoints between 2 images
    - Final stitched image.
    - Properties of your descriptor (as mentioned above in Assignment Instructions)
        - Rotation
        - Scale:
        - Illumination:
        - Applicability in a real-world setting