Introduction to Computer Vision Assignment #2

Due Nov. 2(Mon)

## Homography and Image Stitching

In this project you will implement an image stitching algorithm that automatically stitches multiple images of a scene into a single panoramic image using image warping and homographies.

### 1. Implementation

- **Taking panoramic pictures**
  - Obtain your own images (5 images, $I_1$, $I_2$, $I_3$, $I_4$, $I_5$ ) by taking pictures of a scene in different views by rotating your smart phone camera. Be sure to maintain the same center of projection while taking pictures and adequate overlap between neighboring views are required. To avoid lens distortion, do not use wide-angle lens or mode of your camera.
  - Resize your images into 256 x 256.

- **Feature Extraction**
  - Detect feature points in each image.
  - You need to convert color images to grey images (`rgb2gray`)
  - You can use SIFT feature and its MATLAB code vl_sift, or its C++ version (set the threshold to extract about 200-300 points in each image).
  - You have to make the feature points be evenly distributed over each image and avoid multiple detections at the same point. (Think about how to get features extracted evenly across an image).

- **Feature Matching**
  - Find the putative matches between images $I_i$ and $I_{i+1}$.
  - For measuring the uniqueness of the correspondences, use the ratio of the distance between the best matching keypoint and the distance to the second best one. You can use the code vl_ubcmatch.
  - For speed up the feature matching, use the kd-tree library such as ANN (http://www.cs.umd.edu/~mount/ANN/ ). Plot the detected key points and the correspondences between adjacent images
  - Display the detected correspondences using lines or vectors overlaid on the images.

- **Homography Estimation using RANSAC**
  - Using the detected putative correspondences between $I_i$ and $I_j$, you can estimate the 3x3 Homography matrix $H_{ij}$ between two images using RANSAC and DLT method.

**Objective**

Compute homography between two images

**Algorithm 1 Automatic Estimation of H using RANSAC**

(i) **Interest points:** Compute interest points in each image

(ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure

(iii) **RANSAC robust estimation:** Repeat for $N$ samples

(a) Select 4 correspondences and compute **H** using **DLT** method

(b) Calculate the distance $d_\perp$ for each putative match

(c) Compute the number of inliers consistent with **H** ($d_\perp < t$)

Choose **H** with most inliers

(iv) (Option) **Optimal estimation:** re-estimate **H** from all inliers by minimizing ML cost function with DLT or Levenberg-Marquardt Algorithm

(v) (Option) Guided matching: Determine more matches using prediction by computed **H**

Optionally iterate last two steps until convergence

**Objective**

Given $n \geq 4$ 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}$, determine the 2D homography matrix **H** such that $\mathbf{x}_i' = \mathbf{H}\mathbf{x}_i$

**Algorithm 2 DLT**

(i) For each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ compute $\mathbf{A}_i$. Usually only two first rows needed.

(ii) Assemble $n$ 2x9 matrices $\mathbf{A}_i$ into a single $2n$x9 matrix **A**

(iii) Obtain SVD of **A**. Solution for **h** is last column of **V**

(iv) Determine **H** from **h**

**Algorithm 3** Adaptive determination of the # of samples for **RANSAC**

(i) $N = \infty$, $sample\_count = 0$.

(ii) While $N > smaple\_count$ Repeat

(a) Choose a sample and count the number of inliers.

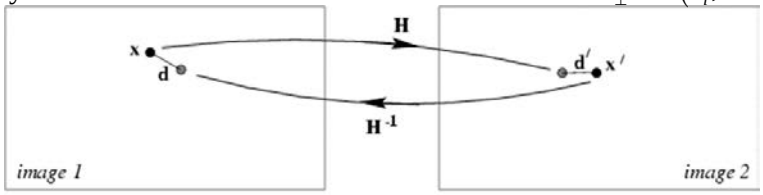
(b) Set $\varepsilon = 1 - \frac{number\ of\ inliers}{total\ number\ of\ points}$

(c ) Set $N$ from $\varepsilon$ and $N = \log(1-p)/\log\left(1-(1-\varepsilon)^s\right)$ with $s = 4, and\ \ p = 0.99$.

(d) Increment the *sample_count* by 1.

(iii) Terminate

- Implement a homography estimation function using RANSAC, `H_ij = HbyRANSAC(x_i, x_j)` for two vectors of corresponding features in `I_i` and `I_j`, respectively (four point features each).
- Use the symmetric transfer error for the distance measure $d_\perp = d\left(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}_j\right)^2 + d\left(\mathbf{x}_j, \mathbf{H}\mathbf{x}_i\right)^2$




- Use DLT for determining $H$ for a sampled 4 correspondences. For randomly sampling matches, you can use the `randperm` or `randsample` functions. The solution to the homogeneous least squares system $\boldsymbol{Ah} = 0$ is obtained from the SVD of $\mathbf{A}_{n\text{x}9}$ by the singular vector corresponding to the smallest singular value:

```
[U,S,V]=svd(A); X = V(:,end);
```

- Set the RANSAC parameter values: *t=1.25, p=0.99*
- Use the adaptive determination method for the # of samples *N*.
- Determine the homographies $H_{12}, H_{23}, H_{34}, H_{45}$ using this function.
- **Reference**: CH4. of "Multiple View Geometry in Computer Vision" by Richard Hartley and Andrew Zisserman, 2nd Ed., Cambridge University Press, March 2004

- **Warping Images**
  - Implement a warping function that transform image $I_1$ to the other image $I_2$ using the homography $H_{12}$.
  - Mosaic the five images by warping other images to the plane of the center image $I_3$. (Backward warping using inverse homography matrix is recommended for better visualization). You can use the `maketform` and `imtransform` functions.
  - For color images, you can use the same homography for each color channel, and warp each RGB channel separately and then stack together to form the output.

- **Result image**
  - Report the final homography estimation results and the total symmetric transfer error for each H.
  - Display the intermediate pairwise matching results, and the final mosaic image.
  - Discuss any issues or artifacts that may be evident in your output.

- **Extra credit**
  - Apply the iterative refinement steps (Steps (iv) and (v) in Algorithm 1) using DLT for finding more accurate homographies, and compare the results with and without these steps.

**Submission instructions: what to hand in:**

- Upload the electronic file that includes the report, source code, and data in a single zip format with the name "**ICV_assignment#2_yourname.zip**" on the ETL class homepage.
- The report should include the brief description of the problems, results, and discussions

**Note:** All works should be individual-based. NO copy is allowed.