EE5490: Image Signal Processing
Lab-3

Image mosaicing

1 Problem statement

Image mosaicing is the alignment and stitching of a collection of images having overlapping regions into a single image. In this assignment, you have been given three images which were captured by panning the scene left to right. These images (`img1.pgm`, `img2.pgm` and `img3.pgm`) capture overlapping regions of the same scene from different viewpoints. The task is to determine the geometric transformations (homographies) between these images and stitch them into a single image.

2 Steps

1. Take `img2.pgm` as the reference image.
2. Determine homography H_{21} between $I_2 =$ `img2.pgm` and $I_1 =$ `img1.pgm` such that $I_1 = H_{21} I_2$.
3. Determine homography H_{23} between $I_2 =$ `img2.pgm` and $I_3 =$ `img3.pgm` such that $I_3 = H_{23} I_2$.
4. Create an empty canvas. For every pixel in the canvas, find corresponding points in I_1, I_2 and I_3 using H_{21}, identity matrix and H_{23} respectively (target-to-source mapping). Blend the three values by averaging them. Employ the values in blending only if it falls within the corresponding image bounds. Choose the origin so as to get a full mosaic.

2.1 Determining homography between two images

1. Determine SIFT features of the two images and determine correspondences between them. File `sift_corresp.m` returns the SIFT correspondences between two images (see Section 2.3). Now to find H such that:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

2. Run RANSAC on matched points (correspondences) to remove outliers (wrong matches), and find the homography between the two images.

   (a) Input: Matched points (x_i, y_i) and (x'_i, y'_i) with $i \in \mathcal{M}$.

   (b) Randomly pick four correspondences (so that we can form eight equations), i.e. (x_i, y_i) and (x'_i, y'_i) with $i \in \mathcal{R} \subset \mathcal{M}$ and $|\mathcal{R}| = 4$, where $|\cdot|$ denotes the cardinality of the set.

   (c) Calculate the homography H using the above four correspondences (see Section 2.2).

   (d) For each of the remaining correspondences (x_i, y_i) and (x'_i, y'_i) with $i \in \mathcal{P} = \mathcal{M} \backslash \mathcal{R}$, check whether they satisfy the homography (within an error bound). If yes, add the index of that correspondence to the consensus set.

$$\begin{bmatrix} x''_i \\ y''_i \\ z''_i \end{bmatrix} \leftarrow H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \text{ and normalize so that } z''_i = 1,$$

   i.e. $x''_i \leftarrow x''_i / z''_i$ and $y''_i \leftarrow y''_i / z''_i$

   If $\sqrt{(x'_i - x''_i)^2 + (y'_i - y''_i)^2} < \epsilon(= 10)$, then update consensus set $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$.

   (e) If the consensus set is large enough i.e. if $|\mathcal{C}| > d(= 0.8|\mathcal{P}|)$, then return this homography H, else go to step (b).

   (f) Output: Homography H.

2.2 Calculating homography

1. Consider a correspondence (x, y) and (x', y'),

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

   Upon normalizing z',

$$x' = h_1 x + h_2 y + h_3 / h_7 x + h_8 y + h_9,$$
$$y' = h_4 x + h_5 y + h_6 / h_7 x + h_8 y + h_9.$$

   Form two equations for each correspondence (corresponding to two rows of matrix A).

$$(x)h_1 + (y)h_2 + (1)h_3 + (0)h_4 + (0)h_5 + (0)h_6 - (x'x)h_7 - (x'y)h_8 - (x')h_9 = 0$$
$$(0)h_1 + (0)h_2 + (0)h_3 + (x)h_4 + (y)h_5 + (1)h_6 - (y'x)h_7 - (y'y)h_8 - (y')h_9 = 0$$

2. Solve the system,

$$A_{8\times 9}\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_9 \end{bmatrix} = 0$$

i.e., find the null space of A.

3. Homography matrix

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}.$$

Normalize H so that $h_9 = 1$.

2.3 Using SIFT

Files needed:

1. `sift_corresp.m`
2. `sift.m` and
3. `sift` (for GNU/Linux) or `siftWin32.exe` (for Windows).

Copy the above three files to the working directory. In GNU/Linux, check that the file `sift` has executable permission. If not, run `chmod +x sift` in a terminal.

Usage: `[corresp1, corresp2] = sift_corresp('img1.pgm','img2.pgm')`
Check the comment lines in the file `sift_corresp.m` for X and Y axes convention.

2.4 Creating the mosaic

Pseudo code:

- Create a large empty canvas C three times the size of the individual images.
- For every pixel (x, y) in the canvas
  $\tilde{x}_1 = x - x_{offset}$, $\tilde{y}_1 = y - y_{offset}$
  $temp = H_{21} * [\tilde{x}_1 \quad \tilde{y}_1 \quad 1]'$
  $x_1 = temp(1)/temp(3)$, $y_1 = temp(2)/temp(3)$
  Obtain the intensity value from $I_1(x_1, y_1)$ using bilinear interpolation, and also return a flag based on whether the point (x_1, y_1) lies inside (flag=1) or outside (flag=0) the boundary of the image I1.

$[int1, flag1] = BilinearInterp(x_1, y_1, I_1)$
Choose x_{offset}, y_{offset} appropriately so as to get a full mosaic.

- Similarly, find the intensity values and the corresponding flags for I2 (using identity matrix), and I3 (using H_{23}).
  $[int2, flag2] = BilinearInterp(x_2, y_2, I_2)$
  $[int3, flag3] = BilinearInterp(x_3, y_3, I_3)$
- $C(x, y) = (int1 + int2 + int3)/(flag1 + flag2 + flag3)$
- end for

–end–