EE5175: Image Signal Processing
Lab-2

Occlusion Detection & Image mosaicing

1 Occlusion Detection

In this problem, we are going to revisit the Occlusion Detection task from Lab-1. The task is to estimate the homography matrix using SIFT based features, assuming that you don't know the underlying transformations and the point correspondences.

1.1 Steps

- Taking `IMG2.pgm` as the reference image, determine homography H between $I_2 =$ `IMG2.pgm` and $I_1 =$ `IMG1.pgm` such that $I_1 = H \times I_2$.
- Transform I_2 using the estimated homography H (target-to-source mapping) and determine the changes between the two images.

Verify that the underlying transformations consist of in-plane rotation and translation. And also that the values matches with those computed in Lab-1.

1.2 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 1.4). 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 $i \in \mathtt{M}$.
   (b) Randomly pick four correspondences (so that we can form eight equations), i.e. (x_i, y_i) and (x_i', y_i') with $i \in \mathtt{R} \subset \mathtt{M}$ and $|\mathtt{R}| = 4$, where $|\cdot|$ denotes the cardinality of the set.

(c) Calculate the homography H using the above four correspondences (see Section 1.3).

(d) For each of the remaining correspondences (x_i, y_i) and (x'_i, y'_i) with $i \in \mathtt{P} = \mathtt{M} \backslash \mathtt{R}$, 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 $\mathtt{C} \leftarrow \mathtt{C} \cup \{i\}$.

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

(f) Output: Homography H.

1.3 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).

$$\begin{aligned} (x)h_1 + (y)h_2 + (1)h_3 + (0)h_4 + (0)h_5 + (0)h_6 \\ - (x'x)h_7 - (x'y)h_8 - (x')h_9 = 0 \\ (0)h_1 + (0)h_2 + (0)h_3 + (x)h_4 + (y)h_5 + (1)h_6 \\ - (y'x)h_7 - (y'y)h_8 - (y')h_9 = 0 \end{aligned}$$

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

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

1.4 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')`

2 Image mosaicing

2.1 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 (`mosaic1.pgm`, `mosaic2.pgm` and `mosaic3.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.

2.2 Steps

1. Take `mosaic2.pgm` as the reference image.
2. Determine homography H_{21} between I_2 = `mosaic2.pgm` and I_1 = `mosaic1.pgm` such that $I_1 = H_{21} I_2$.
3. Determine homography H_{23} between I_2 = `mosaic2.pgm` and I_3 = `mosaic3.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.

2.2.1 Creating the mosaic

Pseudo-code:

```
canvas = zeros(NumCanvasRows,NumCanvasColumns);
for jj = 1:NumCanvasColumns
  for ii = 1:NumCanvasRows

   i = ii - OffsetRow;
   j = jj - OffsetColumn;

   tmp = $H_{21}$ * [i;j;1];
   i1 = tmp(1) / tmp(3);
   j1 = tmp(2) / tmp(3);

   tmp = $H_{23}$ * [i;j;1];
   i3 = tmp(1) / tmp(3);
   j3 = tmp(2) / tmp(3);

   v1 = BilinearInterp(i1,j1,$I_1$);
   v2 = BilinearInterp(i,j,$I_2$);
   v3 = BilinearInterp(i3,j3,$I_3$);

   canvas(ii,jj) = BlendValues(v1,v2,v3);

  end
end
```

where (`OffsetRow, OffsetColumn`) can be chosen such that the final mosaic will fit within the canvas.

2.3 Capturing your own data

Use your mobile phone camera to capture images (three or more), and run your code to generate the mosaic. Ensure that there is adequate overlap between successive images, and the camera is imaging a far-away scene (think why). Bring down the resolution of the input images such that the $width < 1000$ pixels, and convert them to grayscale before using them for mosaicing. (In MATLAB, use '`imresize`' to reduce the image resolution, and '`rgb2gray`' for conversion to grayscale)

–end–