Project 2 Content Based Image Retrieval Using Global and Local Features

Daniel Bryan and Olmo Zavala
Florida State University

November 11, 2014

0.1 Matlab interface

We created a useful interface to define the corresponding points between two images. Figure 1 shows a screenshot of this interface. The user needs to select corresponding points between images and it doesn't have to be in order (one point in one image and the next point in the following image). The interface also provides a button to reset all the points as well as to remove the last added point. These two buttons are very handy when selecting points. Once all the desired buttons have been selected the resulting points are saved in at matlab file that can be stored for later compuatation using the 8-point algorithm.

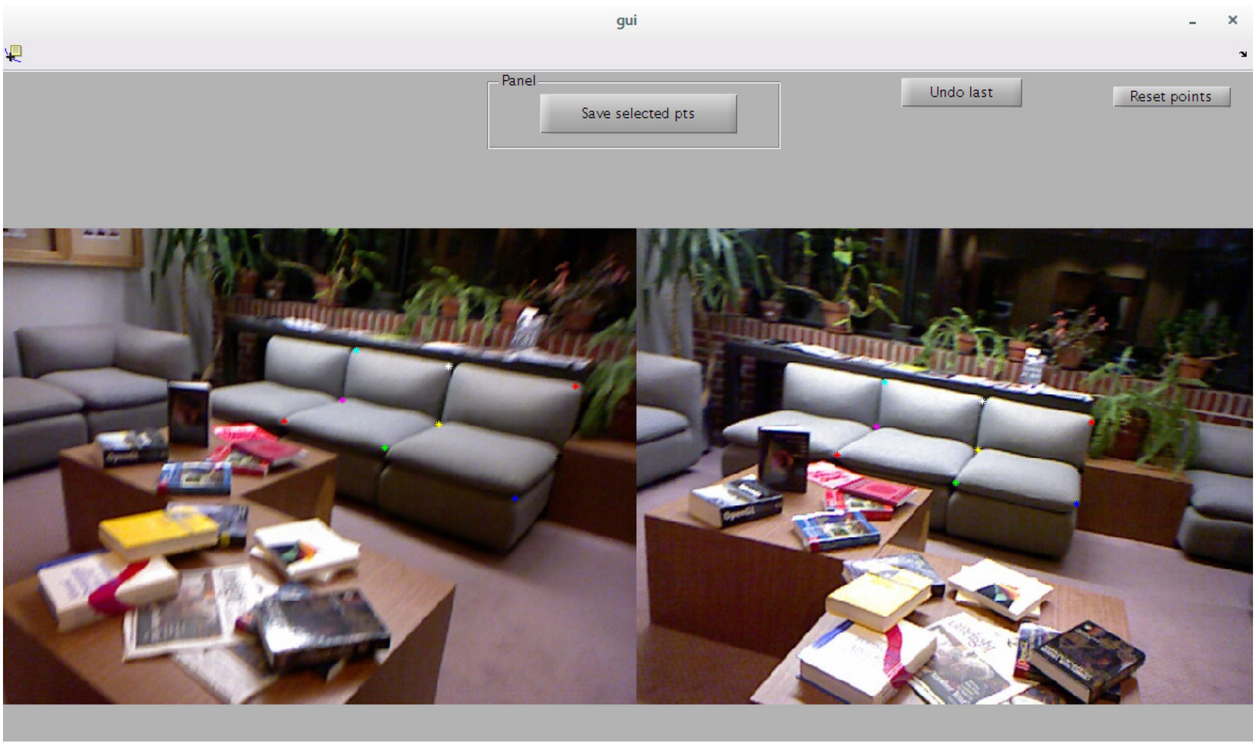

Figure 1: Matlab interface to select corresponding points between two images. Example of selecting 9 points using the Matlab interface.

Each corresponding point is displayed with the same color, this makes it easy to see wich points correspond in both of the images. The coordinates obtained from the selection of points displayed at figure 1 are displayed in figure 2.

In our results we use the example provided by the professor to compare the results.

```
x_l =

  285.8667  195.6923
  387.2000  222.0659
  518.4000  272.7033
  563.2000  230.5055
  441.6000  198.8571
  344.5333  173.5385
  358.4000  123.9560
  451.2000  139.7802
  579.2000  159.8242


x_r =

  202.6667  229.4505
  322.1333  256.8791
  443.7333  277.9780
  448.0000  245.2747
  344.5333  224.1758
  242.1333  200.9670
  250.6667  155.6044
  348.8000  174.5934
  458.6667  196.7473
```

Figure 2: Obtained coordinates from the selected in figure 1.

0.2 Eight-point algorithm

- Normalize image points
  - **Centroid is at the origin**. We create the matrix T_{trans} for each camera like this:

$$\begin{bmatrix} 1 & 0 & -\mu_x \\ 0 & 1 & -\mu_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

    And we multiply each point of the cameras to they corresponding T matrix like this: Tx_i.
  - **RMS distance from the origin is $\sqrt{2}$.** First compute the RMS of the available points:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_i-\mu_x)^2+(y_i-\mu_y)^2))} \tag{2}$$

    Then create T_{scale} and multiply it to each point in the camera. T is:

$$T_s = \begin{bmatrix} \sqrt{2}/RMS & 0 & 0 \\ 0 & \sqrt{2}/RMS & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

- **Multiply each point by $T_n = T_sT_t$ like this $[u\ v\ 1]' = T_nx$. Do it for each camera.**
- **Solve** $x_n'Fx_n = 0$. To do this we need to form the system $Af = 0$ and solve for f. The

matrix A is:

$$A = \begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \\ u'_2u_2 & u'_2v_2 & u'_2 & v'_2u_2 & v'_2v_2 & v'_2 & u_2 & v_2 & 1 \\ u'_3u_3 & u'_3v_3 & u'_3 & v'_3u_3 & v'_3v_3 & v'_3 & u_3 & v_3 & 1 \\ & & & & \vdots & & & & \\ u'_nu_n & u'_nv_n & u'_n & v'_nu_n & v'_nv_n & v'_n & u_n & v_n & 1 \end{bmatrix} F = 0 \tag{4}$$

- **Find least square solution of** $Af = 0$.
  - First find SVD of A $USV = A$.
  - Choose F_{Norm} to be the last column of V
- **Enforcing Singularity**
  - For $F_{Norm} = USV^T$
  - Set S_3=0 for

$$F_{norm} = U \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix} V^T \tag{5}$$

- **Denormalisation**

$$F = T_{NormLeft} * F_{Norm} * T_{NormRight} \tag{6}$$

1 Description

1.1 Color histogram

1.2 Computation time

2 Results