16720 Computer Vision Assignment2

Yu Fang Chang, Robotics Institute
yufangc@andrew.cmu.edu
September 12, 2015

1 Keypoint Detector

1.1 Gaussian Pyramid

Figure 1: Gaussian pyramid for model model_chickenbroth.jpg

1.2 The DoG Pyramid

Figure 2: DoG pyramid for model model_chickenbroth.jpg

1.3 Edge suppression

The DoG function above will have a strong response along edges and will have a large principle curvature across the edge if is a poorly defined peak. Thus, we would compute the Hessian matrix and only concern about the ratio, Trace and Det.

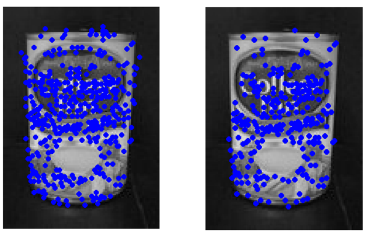

Figure 3: Interest Point (keypoint) Detection without and with edge suppression for model_chickenbroth.jpg.

1.4 Detecting Extrema

In this part, we first checked the local extrema, finding the maximum and minimum. We compare each center point with the 8 neighorhood points in the same level and 9 points in upward, downward level respectively.

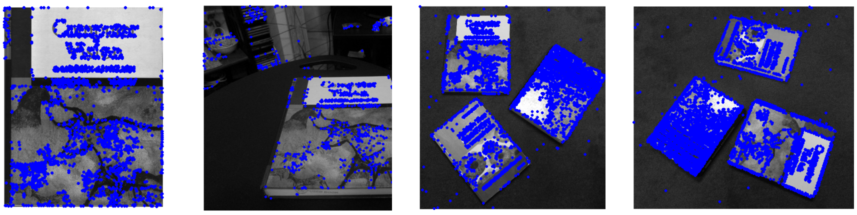

Figure 4: Extrema detection from CV textbook series.

2 BRIEF Descriptor

2.1 Creating a Set of BRIEF Tests

In the provided reading reference, it has experimented few testing sets. The gaussian distribution $(\mathbf{X}, \mathbf{Y}) \sim$ i.i.d. Gaussian$(0, \frac{1}{25}S^2)$ has a small advantage over the other samplings. I implemented a test pattern of center with the most probability. The sampling pairs are chosen randomly, initialized only once and used for each image and local patch.

2.2 Compute the BRIEF Descriptor

After we create the test pattern, we can derive the descriptor of each interesting points. Each has $N = 256$ binary number regarding to the comparison between the test pattern and interesting point's intensity. Since generating the descriptor is working on a patch with specified patch width, if the interesting points is near the border, we may discard the feature at this point.

2.3 Putting it all Together

Simply combine DoGdetector and computeBrief

2.4 Check Point: Descriptor Matching

The matches are shown as Figure 5.

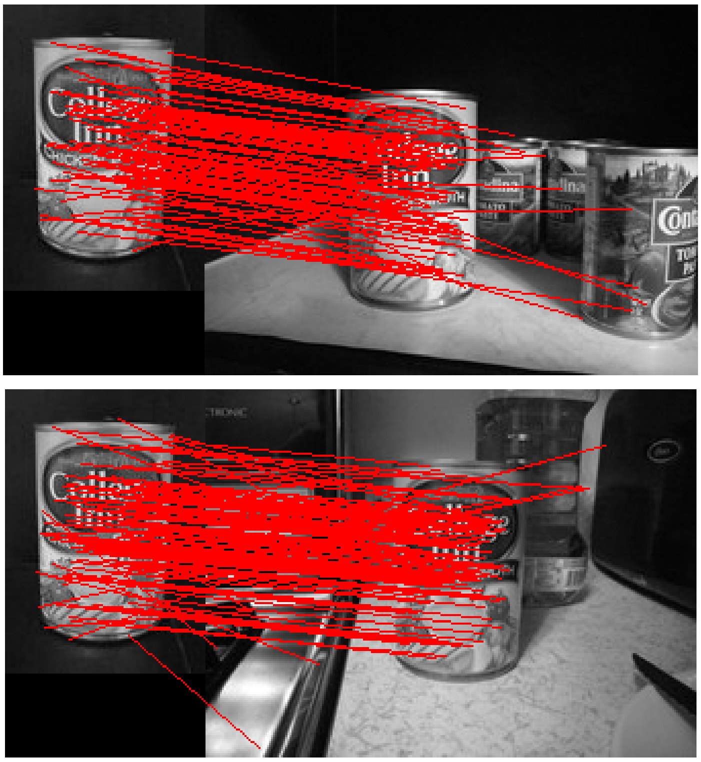

Figure 5: Testing matches on chickenbroth.jpg. We can see that most of the matches are reasonable. The upper figure has stack of same models with different orientaion. The most matches are on the same orientation with the model.

2.5 BRIEF and rotations

Since the descriptor is operated by comparing the same set of smoothed pixel pairs for each local patch, the images will have distinct description with different orientation. Each bit in the discriptor is computed by comparing the pairs created at first. Once the image rotates, the fixed pixel pairs will definitely derive different description. Hence, the BRIEF discriptor is sensitive to in-plane rotation. The testing result is listed as Figure 6. And the degree, match numbers are also indicated as Figure 7. We can found that matching performance of BRIEF falls off sharply for in-plane rotation.

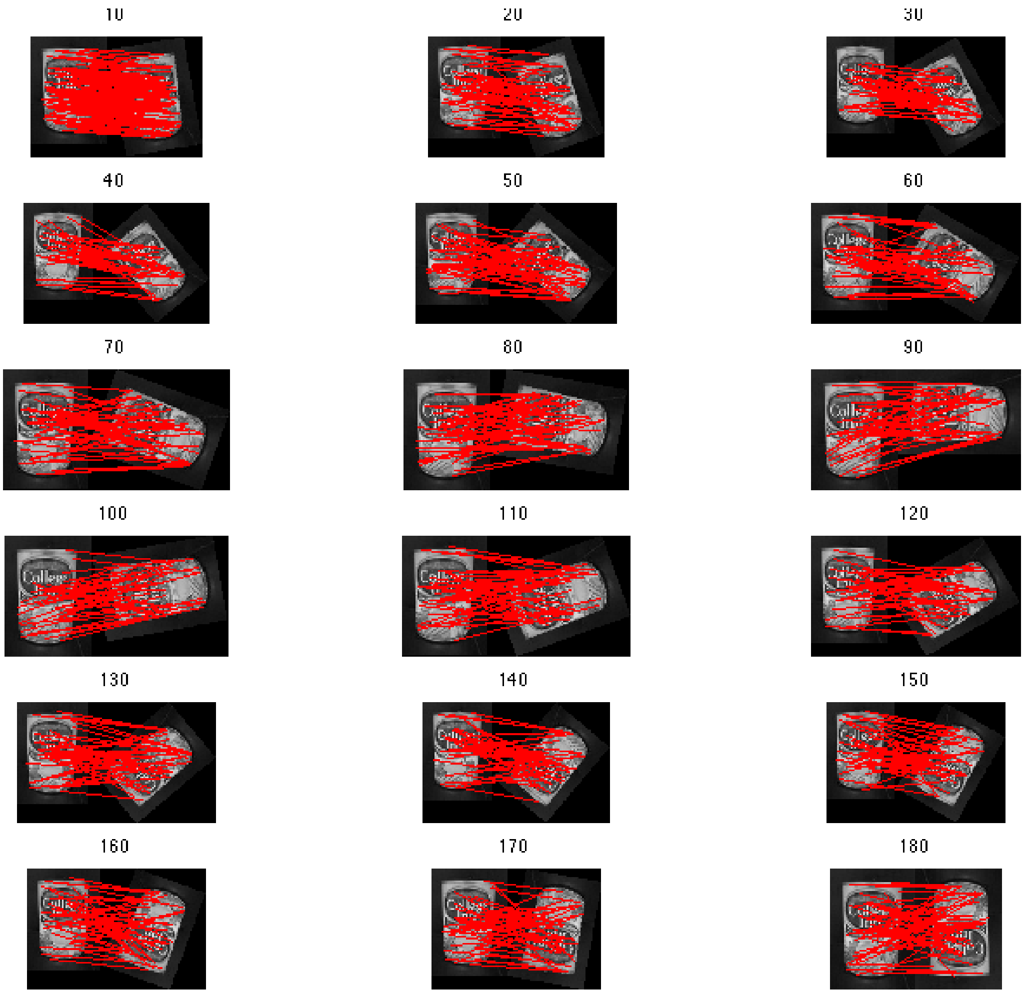

Figure 6: The title above each figure is the rotation degree

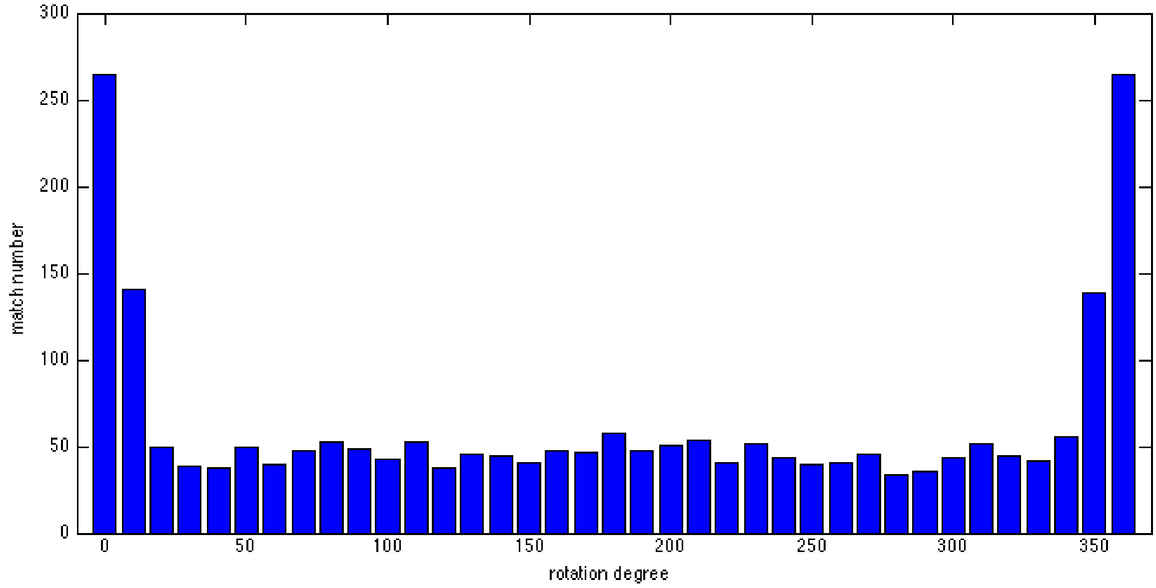

Figure 7: A bar graph showing rotation angle vs the number of correct matches, rotation degree ranging from 10 to 360.

3 Improving Performance

3.1 Rotation invariant on BRIEF

We would like to add some orientation information to our descriptor. The idea is to create more testing sets with different orientation patches. From the initial testing pairs, we rotated the patch with several degrees and add the new pairs into the testing sample. Finally, we get the oriented description in the new descriptor and thus make rotation invariant on BRIEF.

4 Planar Homographies: Theory

(a) Since each point correspondence provides 2 equations, 4 correspondences are sufficient to solve for the 8 degrees of freedom of H. The restriction is that no 3 points can be collinear. Four $2x9A_i$ matrices (one per point correspondence) can be stacked on top of one another to get a single 89 matrix A. The 1D NULL space of A is the solution space for h. The derivation is computed as following.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$$

$$x_i = \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + h_{33}}, \quad y_i = \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + h_{33}}$$

$$
\begin{aligned}
(h_{31}u_i + h_{32}v_i + h_{33})x_i &= h_{11}u_i + h_{12}v_i + h_{13} \\
\rightarrow h_{11}u_i + h_{12}v_i + h_{13} - h_{31}u_i x_i - h_{32}v_i x_i - h_{33}x_i &= 0 \\
(h_{31}u_i + h_{32}v_i + h_{33})y_i &= h_{21}u_i + h_{22}v_i + h_{23} \\
\rightarrow h_{21}u_i + h_{22}v_i + h_{23} - h_{31}u_i y_i - h_{32}v_i y_i - h_{33}y_i &= 0
\end{aligned}
$$

Hense, a point correspondence provides two independent constraints. Given N correspon- dences, the set of 2N independent linear equations can be express as $A_{2N9}h = 0$

$$
A = \begin{bmatrix}
u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 & -x_1 \\
0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 & -y_1 \\
u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 & -x_2 \\
0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 & -y_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_N & v_N & 1 & 0 & 0 & 0 & -u_Nx_N & -v_Nx_N & -x_N \\
0 & 0 & 0 & u_N & v_N & 1 & -u_Ny_N & -v_Ny_N & -y_N
\end{bmatrix}
$$

(b) From above, there are 9 elements in **h**

(c) Since H is defined up to scale, that is, if a point P in a plane is represented by $x = [x_1, x_2, x_2]^T$, then any non-zero scalar multiple of x also represented the same point P. If **H** represents a homography, and $Hx = u$, then $(cH)x = cu$. cu represents the same point as u and thus cH represents the same homography as H. Finally, we can conclude that the degree of freedom of **H** is 8.
Since the homography matrix **H** has 8 degree of freedom, and each points can derive 2 equations (as above (a)), 4 corresponding pairs are enough to constrain the problem.

(d) In class, we have discussed that if we set h_{33} to 1, though we force the degree of freedom to 8, while we in fact has $h_{33} = 0$ then we would get the wrong answer. Instead, since h is only defined up to scale, a scale may be arbitarily chosen for h by insisting that $||h|| = 1$. To estimate h, we can define the problem as a least-square system: $min||Ah||^2 subjectto||h|| = 1$ and it can be written as

$$
\begin{aligned}
f(h) &= \frac{1}{2}(Ah)^T(Ah) \\
&= \frac{1}{2}h^TA^TAh
\end{aligned}
$$

To get the minumum, we take the derivative of f respect to h and set it to zero

$$
\begin{aligned}
0 &= \frac{1}{2}(A^TA + (A^TA)^T)h \\
&= A^TAh
\end{aligned}
$$

From above, we see that h should equal to the eigenvector of A^TA that has an eigenvalue of zero. This result is identical to the result obtained using SVD. SVD of A^TA is UDU^T. Let h be the column of U (unit eigenvector) associated with the smallest eigenvalue in D. (if only 4 points, that eigenvalue will be 0)

(e) Since homography is defined up to scale.

$$
\begin{aligned}
p &\sim \mathbf{K_1}[\mathbf{I} \quad 0]P \\
q &\sim \mathbf{K_2}[\mathbf{R} \quad 0]P \\
&\sim \mathbf{K_2 R K_2^{-1} K_1}[\mathbf{I} \quad 0] \\
&\sim \mathbf{K_2 R K_2^{-1}} x \\
&\sim \mathbf{H} x
\end{aligned}
$$

In this case $\mathbf{H} = \mathbf{K_2 R K_2^{-1}}$.

5 Planar Homographies: Implementation

5.1 Computing Homography

As analysis above, given more than 4 point correspondences, the linear system $Ah = b$ is over-determined. Hence we can solve it by minimizing the least square errors $h^* = minh||Ah||^2$ subject to $||h|| = 1$. The h can be found as the eigenvalues corresponding to the smallest eigenvalues. Hence we perform SVD on $A = U\Sigma V^T$, and since SVD has sorted singular values in diagonal matrix, we extract the last column of V as the resulting h^*.

6 Stitching it together: Panoramas

At first, we cannot get the proper warped image. As I visualized the matches of two incline images (figure 8), I found that there are several mismatches and once the homography computation take them into account, we would totally get the wrong homography matrix. And this may be improved by the RANSAC. The warped image is viewed as Figure 8.

From the result above, we found that some part of picture is clipped. To improve the panoroma effect, I compute the corner of the warped image first and specify my width to 1024. The M matrix has terms only for translation and scaling as following.

$$
M = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}
$$

Once getting the corner, we can compute the original paranorma width and scale it to 1024. $s_x = s_y = \frac{1024}{original\ width}$. If the upmost pixel is out of range (negative value v), I would translate the y axis with $s_y = -vxs_y$. The resulting paranorma is shown in figure 10.

7 RANSAC

Since I have applied RANSAC in the previous process, the results are visualized as above. Inlier matches in Figure 8, paranorma in Figure 10.

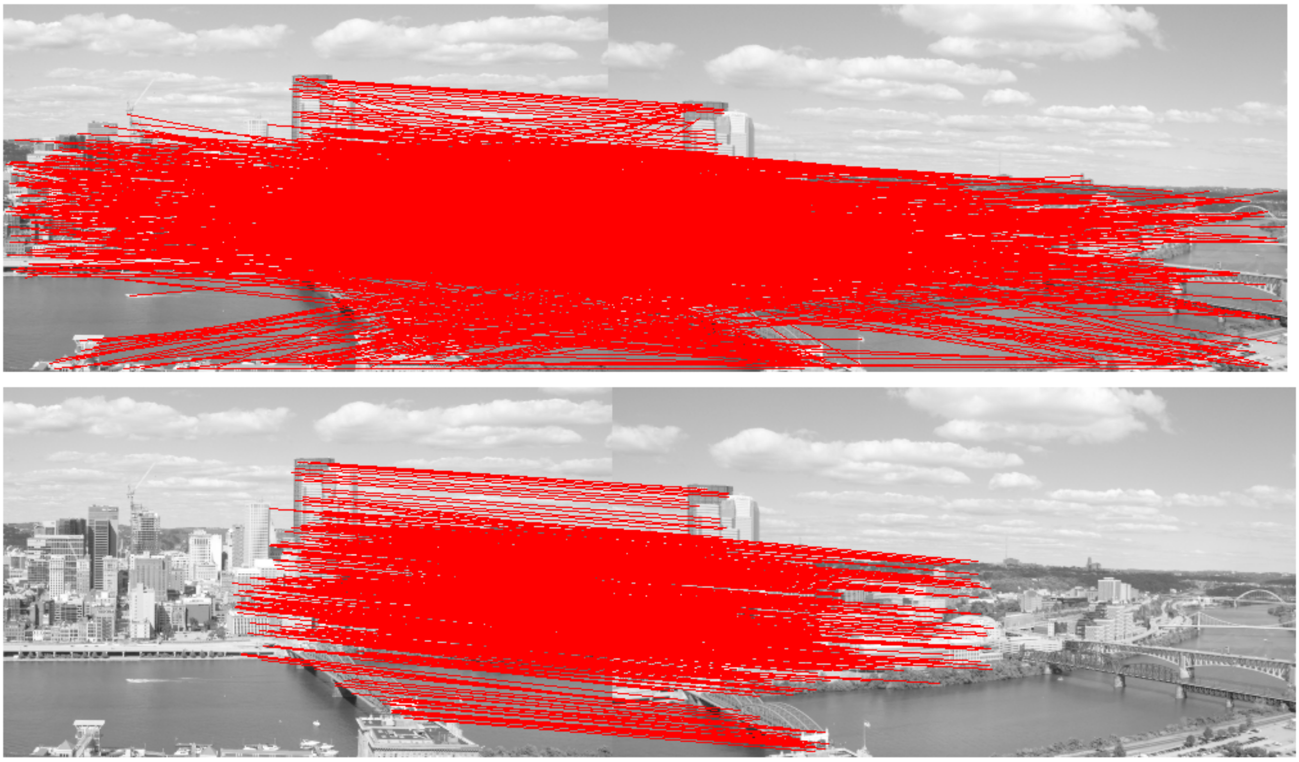

Figure 8: The upper picture visualized the matches without RANSAC, and we can found that there are many outliers and thus we might get the wrong homography. The buttom figure applied RANSAC algorithm with 1000 iterations and range tolerance 20. We get 594 matches with RANSAC while 1396 matches without RANSAC.

8 Extra Credit

I have taken some pictures from my undergrad and also apply some of techniques to improve the performance. With original stitching algorithm (result in Figure 11.), I found that the warpped image will result in great distortion. Thus, I apply Cylindrical Warping before stitching. Also, I derive my camera's focal length as the parameter of the cylinder's radius. The result is shown in Figure 12.

Figure 9: Warped image after applying RANSAC

Figure 10: Final paranorma without clipping the oringinal image

Figure 11: Paranorma with 3 input images

Figure 12: Paranorma with cylindrical warping

Figure 13: Feature detect after cylindrical warping

Figure 14: Final paranorma