ECE 8410: Computer Vision

Laboratory 4 Report: Image Matching Using SIFT

March 26, 2021

Mark Belbin (201618477)
Raylene Mitchell (201415247)
Rodney Winsor (200433886)

Part 1: Top 10 SIFT Matches

First we calculated all the SIFT features using the vl_feature SIFT function. Figure 1 shows an example of test image 1 with all its SIFT features displayed.

Figure 1: All detected SIFT features for test image 1.

Next, the features were matched between the test images and the reference image. The following figures 2-13 show each image adjacent to the reference image with its top 10 matches.

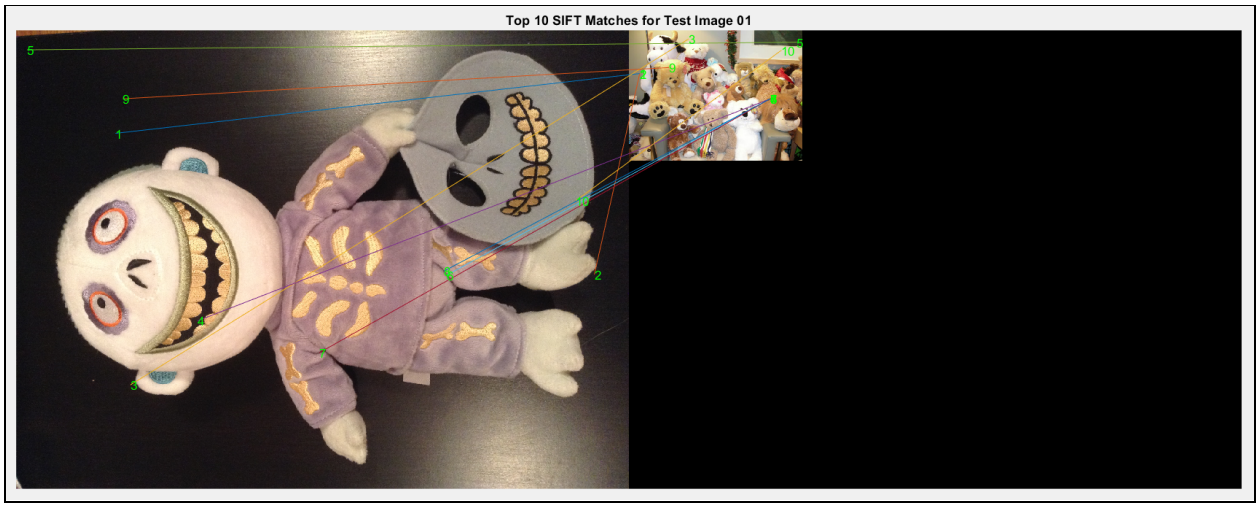

Figure 2: Top 10 SIFT feature matches between test image 1 and the ref image.

Figure 3: Top 10 SIFT feature matches between test image 2 and the ref image.

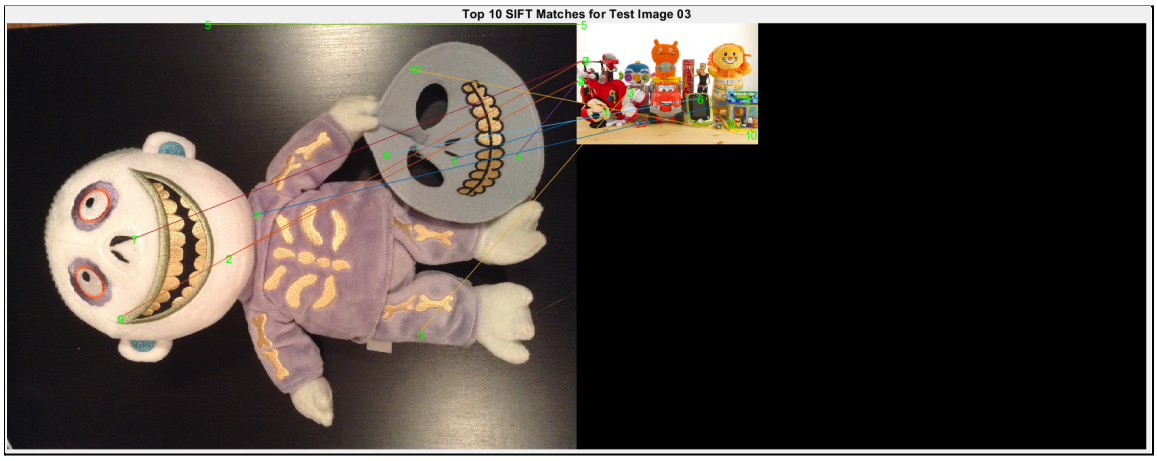

Figure 4 Top 10 SIFT feature matches between test image 3 and the ref image.

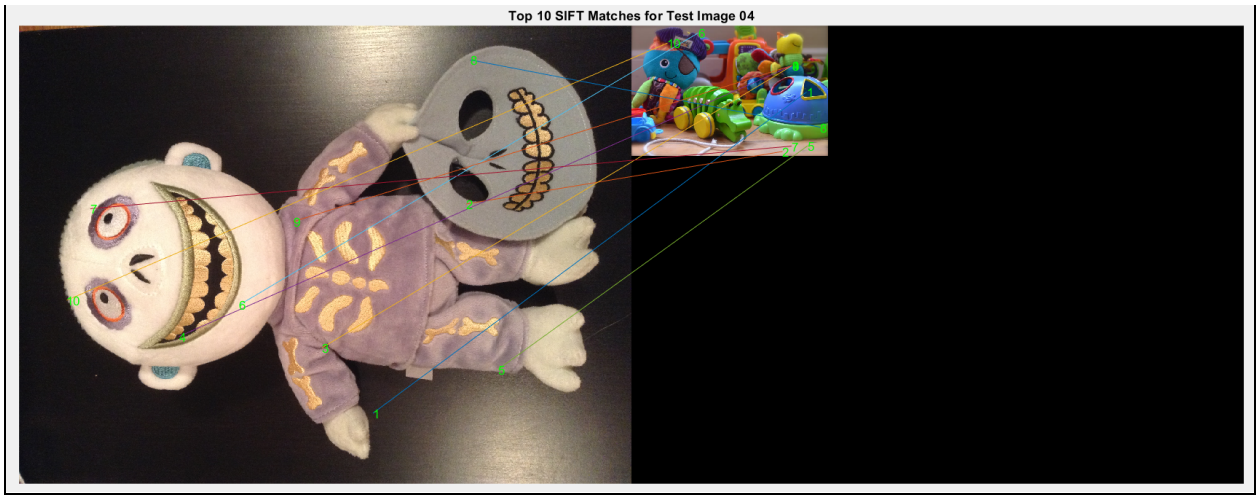

Figure 5: Top 10 SIFT feature matches between test image 4 and the ref image.

Figure 6: Top 10 SIFT feature matches between test image 5 and the ref image.

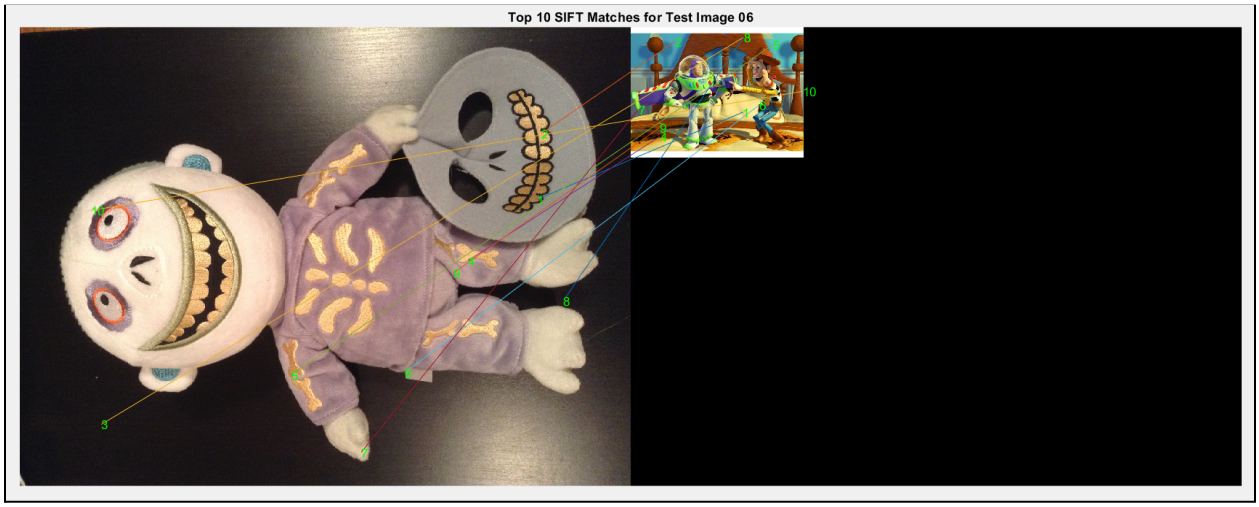

Figure 7: Top 10 SIFT feature matches between test image 6 and the ref image.

Figure 8: Top 10 SIFT feature matches between test image 7 and the ref image.

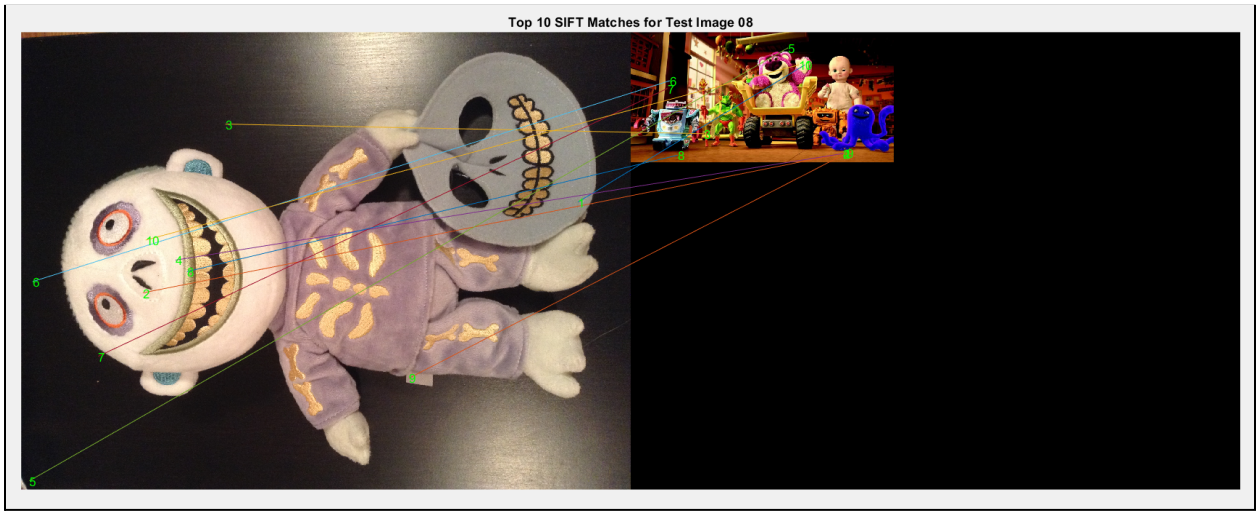

Figure 9: Top 10 SIFT feature matches between test image 8 and the ref image.

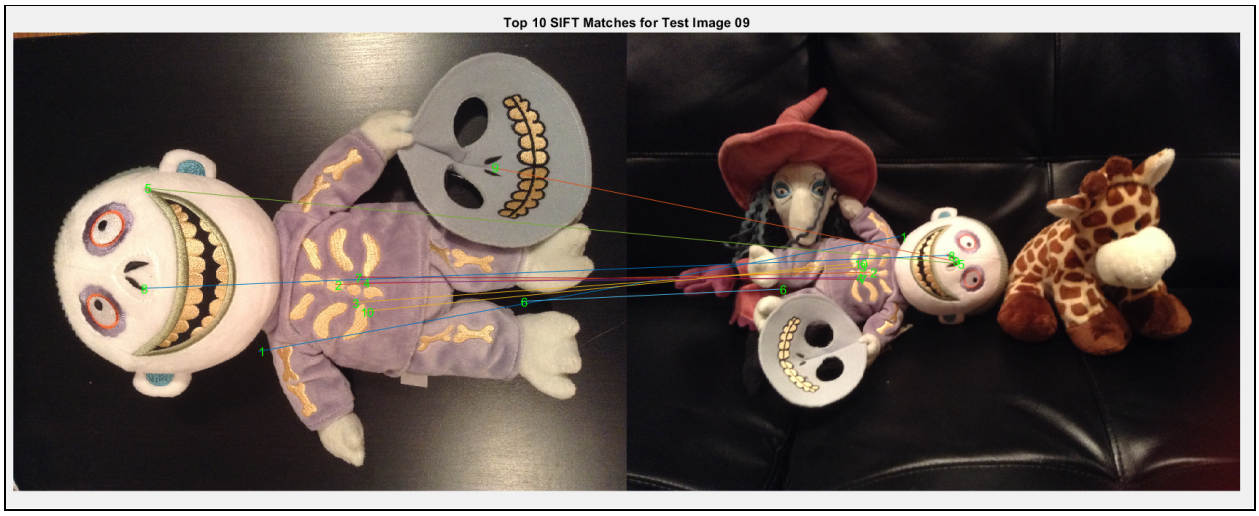

Figure 10: Top 10 SIFT feature matches between test image 9 and the ref image.

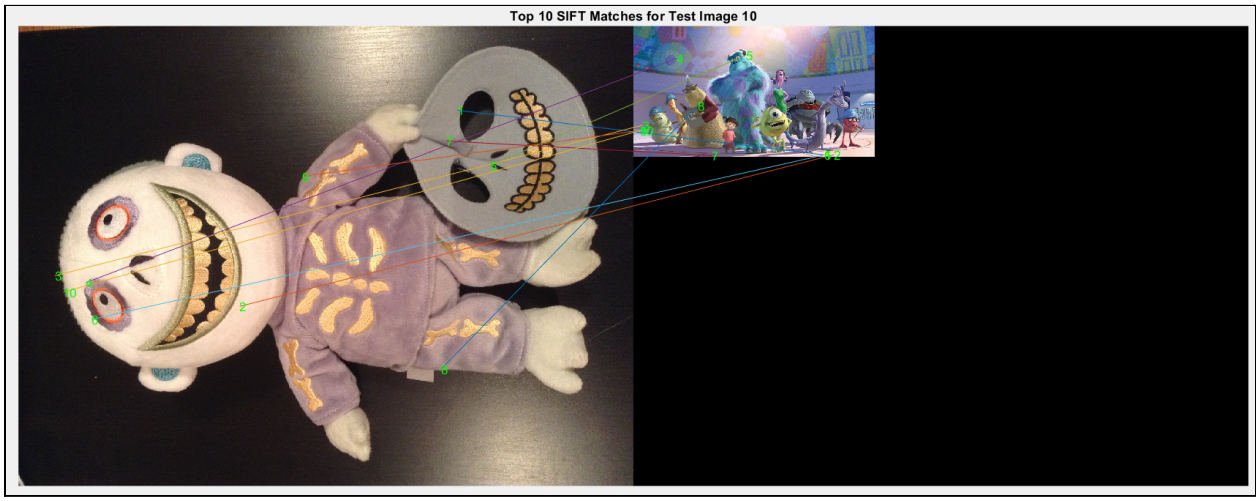

Figure 11: Top 10 SIFT feature matches between test image 10 and the ref image.

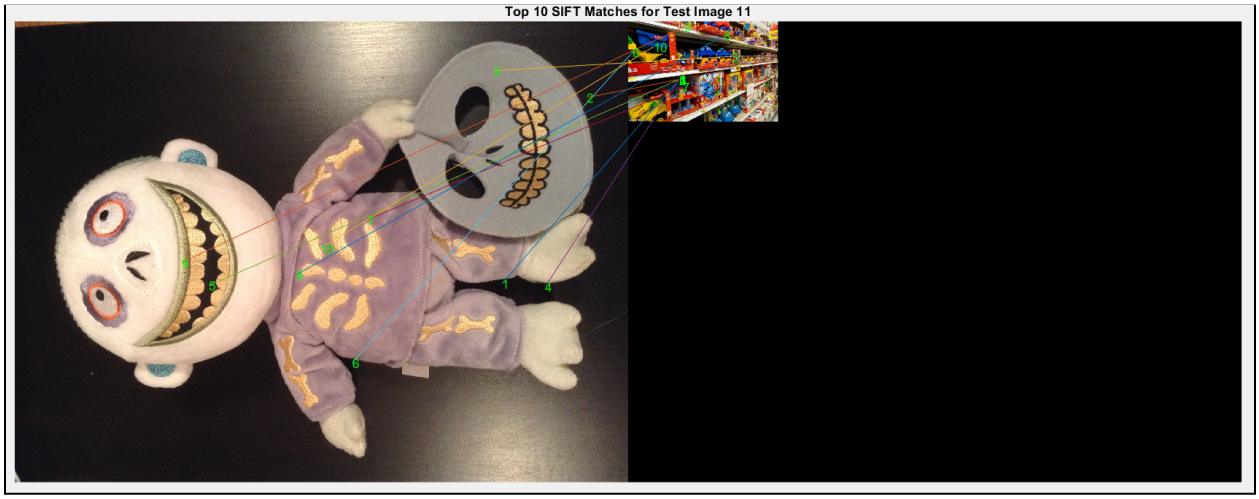

Figure 12: Top 10 SIFT feature matches between test image 11 and the ref image.

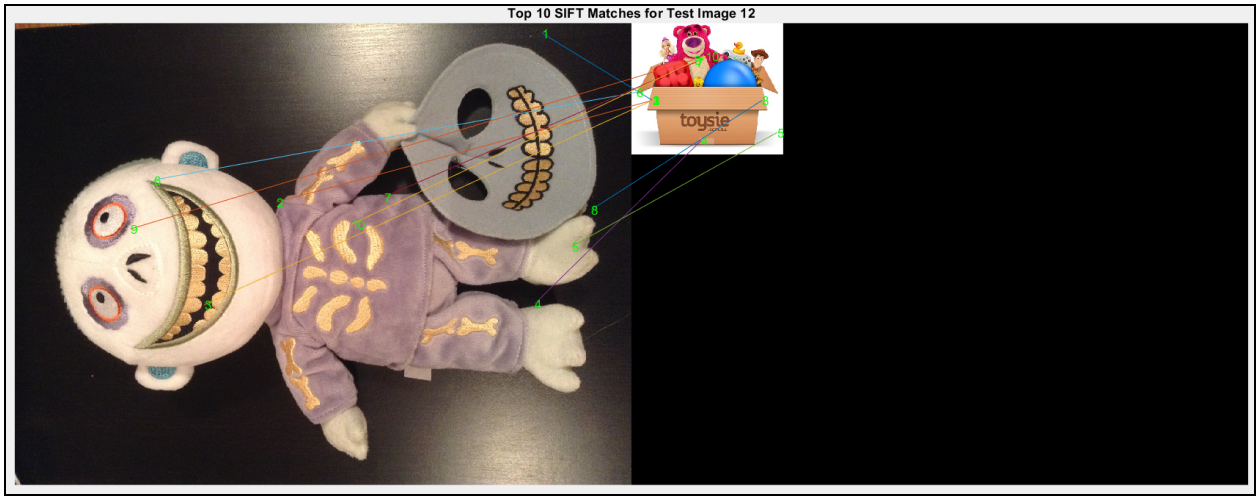

Figure 13: Top 10 SIFT feature matches between test image 12 and the ref image.

Part 2: Image Transformations

With only 3 matched points the best transform we can do is an affine transformation. We used the MATLAB function “estimateGeometricTransform” which uses RANSAC to find the best transform based on matched points. Figure 14 shows the transformation output for each test image.

Image 01 to Image Ref Transform

-0.1878	-0.49727	0
-0.35712	0.14479	0
2520.1	1028.1	1

Image 02 to Image Ref Transform

3.1961	-0.48711	0
3.0544	-0.2989	0
-40.848	1103.4	1

Image 03 to Image Ref Transform

-0.24848	-0.17461	0
0.41252	0.042835	0
1849.1	622.79	1

Image 04 to Image Ref Transform

1.2049	0.48562	0
-2.226	1.8972	0
1717.6	236.58	1

Image 05 to Image Ref Transform

-0.61058	-0.22302	0
0.81094	-5.5675	0
1807.1	878.12	1

Image 06 to Image Ref Transform

0.43017	-0.6618	0
1.9773	-0.19129	0
748.31	1883.3	1

Image 07 to Image Ref Transform

1.3422	-1.5	0
0.3579	0.78335	0
1751.6	961.75	1

Image 08 to Image Ref Transform

-0.43443	2.4467	0
-1.7061	1.5136	0
2008.7	-141.36	1

Image 09 to Image Ref Transform

-1.9474	-0.47276	0
0.48549	-2.0114	0
2871.4	3597.5	1

Image 10 to Image Ref Transform

0.98986	-0.31781	0
-1.1106	-0.30405	0
1657.5	775.65	1

Image 11 to Image Ref Transform

-0.068738	1.0016	0
1.1958	1.2398	0
695.54	747.65	1

Image 12 to Image Ref Transform

1.3364	0.37795	0
1.0267	0.50935	0
1287.8	815.45	1

Figure 14:The transformations based on the top 3 matched features of each respective image with the reference.

Part 3: Number of Inliers

To get the total number of inliers, all matched points were used to recalculate the transformations. The “estimateGeometricTransform” gives the number of inliers as one of its results. Figure 15 shows the code for finding the inliers.

However, there were multiple image pairs with the same number of inliers, and there seemed to be no way to weed out the best image or affect the feature matches in order to get one final image with the most inliers. However, from visual inspection, we knew the object in the reference image is the same as the one in test image 9, so that one was the one we used as our best match. As well, this image always tied for most number of inliers with another image, so it was definitely a good match.

```
% Estimate the affine transformation between f1match and f_refmatch
% using RANSAC
[transform, inliers] = estimateGeometricTransform2D(f1match_points, f_refmatch_points,...
    'affine', 'Confidence', 99.9, 'MaxNumTrials', 5000);

% Calculate amount of inliers
in_count = 0;
score_total = 0;
for n = 1:1:size(inliers,1)
    if inliers(n,1) == 1
        in_count = in_count + 1;
        score_total = score_total + matches(3, n);
    end
end
```

Figure 15: The strength of the match was determined by summing the score of the inlier matches.

Part 4: Best Image Match Transform

The final transform result of test image 9 is shown in Figure 16. A red rectangle is drawn on the chest of the doll in the reference image, and after transforming its coordinates, it roughly shows up on the same doll's chest in test image 9. The code for transforming and drawing the rectangles is shown in Figure 17.

Figure 16: A rectangle was superimposed on the reference image, and the transformation found through part 2 was used to generate a copy of it onto the image with the best match as determined in part 3.

```
%% Show transform result on best inlier image, test image 9
% Original rect coordinates
rect_coord = [1201 1735 1648 1122 1201;
              783 838 1295 1218 783;
              1 1 1 1 1];

% Transform coordinates based on SIFT feature matched transform
rect_coord_transform = inv(transform9.T)' * rect_coord;
o = size(img_ref,2);

% Show original and transformed rectangle
figure;
imshow([img_ref,im9],[])
hold on;
line(rect_coord(1,:),rect_coord(2,:),'Color', 'red', 'LineWidth', 2)
line(rect_coord_transform(1,:)+o,rect_coord_transform(2,:),'Color', 'red', 'LineWidth', 2)
```

Figure 17: This is the code that was used to superimpose the rectangle onto the matched images.