Machine Perception Assignment 1

Reece Jones   
Student ID: 19476111

# Task 1

## Harris Corner Detector

The Harris corner detector [1] is variant under scaling and invariant under rotation. Scale invariance is not achieved since the size or standard deviation of the window function is a fixed parameter of the algorithm; with images of different scale, the window will cover a different size image patch. I expect that the higher the ratio of window size to feature size, the lower the corner response, since the window will include a higher proportion of noncorner pixels. The experimental results are shown in Fig. 1. Interestingly, as the image size increases, the number of corners detected decreases, contrary to my hypothesis. I believe that in the process of resampling the images, the increase in the size of the edges has reduced the gradient, reducing the corner response. This effect is likely dependent on the resampling method used (preliminary testing with nearest-neighbour resampling indicates different results).

Rotational invariance is achieved because the corner response is calculated based on the eigenvalues of the autocorrelation matrix at a point [2]. These eigenvalues give the magnitudes of curvature of the image surface in the directions of most and least curvature [1]. That is, any rotation present has no effect on the eigenvalues, and therefore no effect on the corner response. The experimental results are shown in Fig. 2; as expected, the same corners are detected regardless of rotation.

The experimental results for the dugong image are similar and may be reproduced using the code in Appendix 1.

## Intensity Histograms

Image histograms are variant under scaling and may be variant under rotation (depending on the rotation). Scaling an image changes its resolution, and since histograms are based on pixel counts, the histogram will be affected. However, for small or moderate changes in scale, the relative ratios of intensities will be retained, i.e. the histogram will have approximately the same “shape” (if normalised histograms are used, then scale invariance is achieved). This is demonstrated in Fig. 3. Notice that the histogram peaks are vastly different, but the overall shape is almost identical.

Histograms may be invariant under rotation, depending on whether the rotation affects the image bounds. Typically, histograms are calculated across an image or bounding box, the content of which will change under most rotations (consider the 45-degree rotation of an entire image – some regions are cut off, and empty regions are introduced). If we apply a rotation but adjust the bounds within which we calculate the histogram accordingly, then the histogram will be approximately unchanged. The experimental results are shown in Fig. 4. A fixed bounding box was selected, with the image rotates with respect to it. The image content inside the box did change, but since the background (ocean) is mostly homogeneous, the histograms are similar.

|  |  |  |
| --- | --- | --- |
| **Scale** | **Keypoints** | **Keypoints common with 1x** |
| 1x | 16 | 16 |
| 1.25x | 25 | 13 |
| 1.5x | 32 | 16 |
| 2x | 45 | 16 |

|  |  |  |
| --- | --- | --- |
| **Rotation** | **Keypoints** | **Keypoints common with 0°** |
| 0° | 16 | 16 |
| 15° | 16 | 12 |
| 45° | 16 | 11 |
| 75° | 15 | 10 |

The experimental results for the playing card image are similar and may be reproduced using the code in Appendix 1.

## Scale-invariant Feature Transform (SIFT) Keypoints

SIFT [3, 4] keypoints are moderately invariant under small to moderate scaling and rotation. Scale invariance is achieved by examining the image at many scales (via Gaussian filtering and image downsampling), transforming the image into “scale space”. Keypoints are detected within this scale space, yielding not only the keypoint’s location in the image, but also the scale at which the keypoint exists. Further processing of the keypoint is adjusted for this scale, producing keypoints which are largely invariant to scale. The experimental results are summarised in Fig. 5. As the image size increases, the number of keypoints detected increases (presumably since there are simply more pixels in the image), but a significant proportion of them match the keypoints in the original image region. The match proportions upwards of 80% for small scaling factors coincides with the results presented in [3].

Rotational invariance is (theoretically) achieved since keypoint detection only considers the magnitude of the difference of Gaussians (which is isotropic) [3, 4]. The experimental results are summarised in Fig. 6. Surprisingly, only about 65-75% of keypoints from the original image region are present in the rotated versions. My only explanation as to why such a large proportion of keypoints did not match is that it is as a result of the image resampling during rotation, especially given that the image region is small, however, I am unsure. It seems plausible that a more complex explanation exists, due to the many steps in the SIFT algorithm.

The keypoint-annotated images are omitted for brevity but may be reproduced using the code in Appendix 1. The experimental results for the playing card image are similar and may be reproduced using the same code.

# Task 2

## Local Binary Patterns (LBP) Features

LBP [5] generates an 8-bit integer feature for every pixel in a greyscale image (or image region). The feature is determined from the 8 pixels neighbouring the target location. These neighbouring pixels are traversed in a fixed order and compared to the target pixel. If the neighbouring pixel is less than the target pixel, a 1 is recorded; if the neighbouring pixel is greater than the target pixel, a 0 is recorded. The 1s and 0s are concatenated to form an 8-bit value, which is the feature descriptor for the target location. Further processing may involve dividing the image into cells, within which histograms of the feature values are calculated.

The main advantage of LBP is that it is quite effective compared to other texture descriptors, as found by [5] and [6]. Contributing to its effectiveness is its invariance under global changes in illumination. The algorithm may be extended to consider neighbouring pixels at any radius, allowing adjustment of the locality of detail that is extracted [7]. Finally, the algorithm is simple and computationally efficient.

The disadvantages of LBP are that it is not (without extensions) scale nor rotationally invariant and does not consider colour information.

## Scale-invariant Feature Transform (SIFT) Features

SIFT [3, 4] generates feature descriptors from the local neighbourhoods around the detected keypoints. First, the 16x16 region of pixels centred at the keypoint is considered and divided into 16 4x4 pixel blocks. For each block, an 8-bin histogram of gradient orientations is created. The orientation of each pixel is calculated at the scale of the detected keypoint, achieving scale invariance, and is given relative to the orientation of the keypoint, achieving rotational invariance. Each pixel’s histogram vote is weighted by the magnitude of its gradient, and by a Gaussian window centred at the keypoint. The 16 histograms are concatenated into a single 128-element vector to form the feature descriptor. Finally, this vector is thresholded and normalised to achieve some invariance to illumination changes.

The main advantages of SIFT features is, of course, that they are invariant under scaling, rotation, and illumination changes. Another advantage, demonstrated in [4], is that the high distinctiveness of the descriptors enables reliable feature matching even within very large databases.

The disadvantages of SIFT features are that the algorithm is quite complex, and colour information is not considered. [4] also notes that the high dimensionality of the feature descriptors inhibited feature matching efficiency when used with nearest-neighbour classification.

## Histogram of Oriented Gradients (HOG) Features

HOG [8] generates a feature descriptor for an image (or image region), derived from 8x8 pixel cells. A gradient orientation histogram is created for each cell. The histogram uses 9 bins for 0° to 180°, with orientations taken modulo 180° (this was found to be most effective for human detection). Each pixel’s vote is linearly interpolated between the 2 nearest histogram bins and is additionally weighted by the magnitude of the gradient. The cells are then grouped into overlapping 2x2 cell (16x16 pixel) blocks and normalised, to account for local illumination changes. Finally, the blocks are flattened and concatenated into a single vector to form the feature descriptor.

The primary advantage of HOG features is their excellent performance in the detection of humans (significantly better than other techniques), in part due to their illumination invariance [8].

The disadvantages of HOG features are that they are not scale nor rotationally invariant. Another possible disadvantage is the high dimensionality of the descriptor, which may be a challenge for classification.

## Comparison of SIFT and HOG Descriptors

Fig. 7 and 8 compare the similarity of SIFT and HOG feature descriptors taken from a region of the dugong image, across different scales and rotations. The descriptors are taken at SIFT keypoints that match in location and orientation across all versions of the image. For each transformed version of the image, the Euclidean distance is calculated between its matrix of descriptors and the matrix of descriptors from the original, untransformed image.

As expected, the HOG descriptors exhibit less similarity as the degree of transformation increases, since HOG is variant under scaling and rotation. The SIFT descriptors also exhibit some dissimilarity; however, to a lesser amount. Critically, while the HOG descriptor similarity is strictly decreasing, the SIFT descriptor similarity is not particularly correlated to the amount of transformation. This result highlights the scale and rotational invariance of SIFT.

Interestingly, both SIFT and HOG descriptors exhibit a large jump in dissimilarity between the untransformed image versions and the first transformed version. From a theoretical standpoint, I would expect the smallest amount of transformation to produce the smallest descriptor difference. Additionally, the SIFT descriptor difference decreases from scales of 1.6x to 2x – I would expect that it stays the same or increases slightly. I believe these observations are an effect of the image resampling during scaling/rotation. Preliminary test

ing with the application of a transformation followed by the inverse transformation – theoretically producing no change – shows similar patterns. Further investigation may be required to determine the effects of resampling on the results presented here.

The experimental results for the playing card image are similar and may be reproduced using the code in Appendix 2.

# Task 3

## Binarisation

Despite consisting of only 3 colours and simple geometry, it is not entirely trivial to binarise the playing card image. I assumed that both the black and white portions of the image are to be considered as background, meaning that both low and high intensity pixels are background. Therefore, neither a simple threshold nor Otsu’s binarisation are applicable. However, between the white and black on the edge of the card, there is also a small region of medium-intensity pixels (presumably from interpolation or antialiasing), meaning that foreground pixels are not uniquely identified by medium intensity either. I therefore decided to threshold based on pixels’ red values: a pixel is foreground if its red value is significantly larger than its blue and green values; all other pixels are background. This scheme produces a good binarisation, which may be seen in Fig. 9. The only imperfections present are aliasing artifacts on the edges of the diamonds, which I do not think are possible to avoid in a binary image.

As a real-world scene, the dugong image was somewhat more difficult to binarise. Simple thresholding or Otsu’s binarisation were not applicable, as the ocean has a deceptively large intensity gradient between the outer edges and centre (this is not easily perceivable to the human eye). Such methods are likely to fail to classify the entire ocean as background. However, the ocean does have an approximately constant hue, distinct from the foreground objects. Therefore, I decided to use a similar technique to that used for the playing card image. A pixel is classified as background if its blue and green values are significantly larger than its red value; all other pixels are foreground. Some tuning of the threshold ratios was required to obtain the desired binarisation, as the colour variation is greater than that of the playing card image. There are some small speckles of lighter colour, present in the ocean that were similar enough to the dugongs’ colour to be frequently classified as foreground. To remove these outliers, the image is preprocessed with a small median filter. The median filter was chosen as it preserves edges better than a Gaussian or box filter. Its small size avoids the loss of fine detail such as the smaller dugong’s tail. The resulting binarisation is very good and is shown in Fig. 11.

## Extracted Objects

Since clean binarisations were achieved, the extraction of objects was straightforward using a connected components algorithm. In the playing card image, 6 objects were extracted. In the dugong image, 2 objects were extracted. The 2 dugongs could not be separated as distinct objects; their overlap and similar colour makes such a task significantly more difficult. The objects and their areas are shown in Fig. 10 and 12. Note that the areas include only the foreground pixels, not the entire bounding box.

# Task 4

## Playing Card Image

TODO

## Dugong Image

TODO

##### References

1. C. Harris and M. Stephens, “A combined corner and edge detector,” *Alvey Vision Conference*, vol. 15, no. 50, pp. 147-151, 1988.
2. R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Accessed Sep. 14, 2020. [Online]. Available: http://szeliski.org/Book.
3. D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410.
4. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
5. T. Ojala, M. Pietikäinen and D. Harwood, “A comparative study of texture measures with classification based on feature distributions,” *Pattern Recognition*, vol. 29, no. 1, pp 51-59, Jan. 1996, doi: 10.1016/0031-3203(95)00067-4.
6. T. Ahonen, A. Hadid and M. Pietikäinen, “Face description with local binary patterns: Application to face recognition,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.
7. T. Ojala, M. Pietikäinen and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002, doi: 10.1109/TPAMI.2002.1017623.
8. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.