COIFv3: Concentric Oval Intensity Features Version 3

Daniel Puckowski
December 2022

Abstract

In this paper, I present an update to the novel rotation-invariant interest point descriptor, coined COIF (Concentric Oval Intensity Features). The descriptor is straightforward to implement and feature matching is time efficient. COIF may be used to detect rotated images and may be used for image stitching in panorama applications. COIF demonstrates the feasibility of using luminance histograms for feature matching.

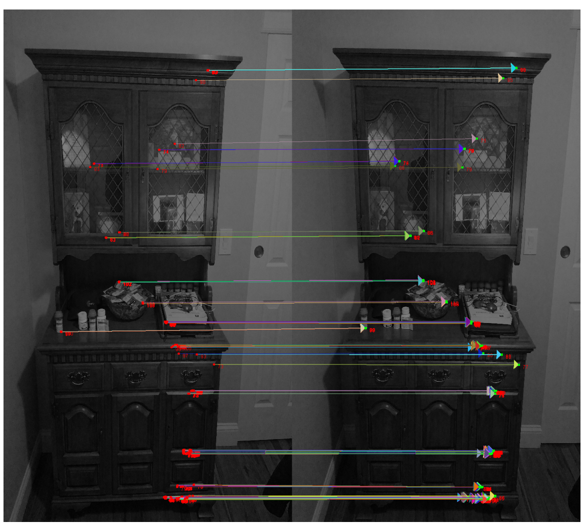

Figure 1. Typical matching result using COIF on real-world images with a minor translation with default settings. Many matches are detected with few incorrect matches.

1. Introduction

Feature matching—the process of finding matching or similar regions between two images of the same scene or object—is a common computer vision task. Feature matching may be a step for object recognition, a step for image stitching, and a step for pattern tracking. Keypoint detectors and descriptors such as SIFT, SURF, and ORB are commonly and successfully used for this task [4] [5] [6]. But the ORB binary descriptor, for example, is not easy to implement. To implement ORB, one needs to implement procedures to produce FAST features filtered by the Harris measure for a scale pyramid of an image, compute the orientation of a FAST feature by determining the intensity centroid, compute BRIEF descriptors for image patches from a set of binary intensity tests, perform a greedy search for a set of uncorrelated tests with means near 0.5 to generate rBRIEF descriptors, then implement Locality Sensitive Hashing (LSH) to perform a nearest neighbor search [4]. By contrast, COIF is meant to be easy to implement and is meant to be easy to optimize so that it is time efficient and so that feature matching may be performed in real time without the need for GPU acceleration on low-power devices.

2. Related work

Keypoints The Moravec corner detection algorithm, introduced by Hans P. Moravec in 1977, is one of the earliest corner detection algorithms [1]. The Moravec algorithm defines a corner as a point with low self-similarity.

Descriptors Traditionally, image retrieval is based on the representation of the image content through features thought to be relevant for the image description. Luminance, color, edge strength, and textural features are commonly used. Vertan and Boujemaa use fuzzy color histograms and their corresponding fuzzy distances for the retrieval of color images within various databases [2]. Vertan and Boujemaa use fuzzy distances due to the imprecision of the pixel color values.

Tola, Lepetit, and Fua developed a local descriptor, DAISY, which depends on histograms of gradients like SIFT and GLOH but uses a Gaussian weighting and circularly symmetrical kernel [3]. Tola, Lepetit, and Fua compute 200-length descriptors for every pixel in an 800x600 image in less than 5 seconds. DAISY consists of a vector made of values from the convolved orientation maps located on concentric circles centered on the location, and where the amount of Gaussian smoothing is proportional to the radii of the circles [3].

Luo, Xue, and Tian proposed a novel method based on making use of both SIFT features and the local intensity histograms on the feature points in order to achieve more robust image matching [7]. Luo, Xue, and Tian demonstrate that many false matches can be rejected by the proposed method.

3. Keypoint extraction

The first step for COIF keypoint extraction is to convert a given image to a grayscale image with intensity values which range from 0 to 255. The grayscale value for a given pixel is determined by adding the sum of (red intensity x 0.299) + (green intensity x 0.114) + (blue intensity x 0.587).

$$\sum_{C \in \{R,G,B\}} w_c \cdot C$$

Figure 2

$$w_R = 0.299,\ w_B = 0.587,\ w_G = 1 - (w_R + w_B)$$

Figure 3

Optionally, the grayscale image may be flattened using a simple approach where luminance values are multiplied by a value *d*, where *d* is typically 0.5, and the smallest integer value that is greater than or equal to the result is assigned as the new luminance value.

$$\begin{array}{l} for\ x = 0\ to\ width \\ \quad for\ y = 0\ to\ height \\ \quad\quad image[x][y]\ =\ ceil(image[x][y]\ \times\ 0.5) \end{array}$$

Figure 4

When set up to perform matching between two images, *d* may be adjusted per image such that the two images will have approximately the same average luminance.

The next step of COIF keypoint extraction is to implement the Moravec corner detection algorithm using an 8 x 8 pixel window shifted in the eight principle directions (horizontally, vertically, and four diagonals). The intensity variation is calculated by taking the sum of squares of intensity difference of corresponding pixels between a window centered on a pixel and the shifted window.

$$S_W(\triangle x,\ \triangle y) = \sum_{x_i \in W} \sum_{y_i \in W} (f(x_i,\ y_i)\ -\ f(x_i\ -\ \triangle x,\ y_i\ -\ \triangle y))^2$$

Figure 5

$$S_W(-1,\ -1),\ S_W(-1,\ 0), \ldots S_W(1,\ 1)$$

Figure 6

The smallest intensity variation from 8 eight windows is selected and then a threshold is applied. If the intensity variation is greater than the threshold a point of interest, either an edge or a corner, at pixel location (x, y) is considered detected. The fact that the Moravec algorithm is not rotation invariant does not matter at this stage. The Moravec algorithm is chosen for simplicity, but another interest point detector which is capable of identifying points of interest centered on a single pixel may be used, like the Harris corner detector.

The next step for COIF keypoint extraction is to create histograms of concentric ovals of pixel intensity for a given radius *r* from the Moravec algorithm identified pixel, typically with a radius of 24 pixels.

$$dx = x - circleX,\ dy = y - circleY$$

$$dist = dx \cdot dx + dy \cdot dy$$

Figure 7

The first histogram consists of 256 bins containing occurrences of intensity values for all pixels within the radius *r* of 24 pixels for the Moravec algorithm identified pixel. The second histogram consists of 256 bins containing occurrences of intensity values for all pixels within *r2* where *r2* is the radius *r* squared divided by 3.

$$r = radius$$

$$if\ (dist \leq r^2)\ histogram[image[x][y]] + 1$$

$$if\ (dist \leq r^2 \div 3)\ innerHistogram[image[x][y]] + 1$$

Figure 8

With the histograms tallied, keypoints are considered extracted and COIF descriptors may be computed next.

4. Descriptor computation

To create the COIF descriptors from the set of concentric oval intensity histograms derived from all Moravec algorithm pixel locations which passed a given threshold, first create a histogram *h* of 256 divided by *k* bins and initialize all values to 0. The initial value of *k* is 2. The value of each bin index of *h* is the absolute value of the difference of the sum *s*, where *s* is the next *k* consecutive outermost intensity histogram indices, with the sum *s2* where *s2* is the same *k* consecutive inner intensity histogram indices. Note that the sums *s* and *s2* are not reset to zero for each collection of four indices.

$$n = 0;\ sum = 0;\ sumInner = 0;\ binIndex = 0;\ k = 2;$$

$$for\ i = 0\ to\ 255$$

$$sum + histogram[i]$$

$$sumInner + innerHistogram[i]$$

$$n + 1$$

$$if\ (n == k)\ distances[binIndex] = |sum - sumInner|;\ n = 0;\ binIndex + 1$$

Figure 9

Once each bin for histogram *h* has been computed the COIF descriptor, which should be rotation invariant, has been created.

In addition to histogram *h*, a distinctiveness measure is also computed to assist with feature matching. The distinctiveness measure simply checks all values of the outermost histogram and increments a measure where the value is less than some value *n*, where *n* is typically 2. To finalize the distinctiveness score, the incremented measure is subtracted from 256 and the result is the distinctiveness score.

$$score = 0$$

$$for\ i = 0\ to\ 255$$

$$if\ (histogram[i] < n)\ score + 1$$

$$distinctiveness = 256 - score$$

Figure 10

5. Matching of features

To begin matching, COIF descriptors are removed as possible matching candidates if its distinctiveness measure matches certain criteria. This step speeds up matching and improves accuracy in cases where there are textures in the source images. The standard deviation of the distinctiveness score of all COIF descriptors must be determined for all source images.

```
sum = 0, standardDeviation = 0
length = coif list size
for i = 0 to length
    sum = sum + coif list[i].distinctiveness
mean = sum ÷ length
for i = 0 to length
    standardDeviation = standardDeviation + pow(coif list[i].distinctiveness − mean, 2)
standardDeviation = sqrt(standardDeviation ÷ length)
```

Figure 11

Next, the highest distinctiveness score occurrence must be found for all source images. For each COIF descriptor for each source image, if the descriptor distinctiveness is less than some threshold *g*, where *g* is typically 40, the COIF descriptor is removed as a possible matching candidate as it is deemed not distinctive enough. Furthermore, if the descriptor distinctiveness falls within one standard deviation above or below the highest occurrence distinctiveness value, it is also removed as a possible matching candidate as the feature is deemed too common.

```
for i = 0 to coif list size
    if (coif list[i].distinctiveness ≤ (commonOccurrence + standardDeviation))
        if (coif list[i].distinctiveness ≥ (commonOccurrence − standardDeviation))
            coif list remove i
    else if (coif list[i].distinctiveness < 40)
        coif list remove i
```

Figure 12

To match our COIF features extracted from two images for applications such as image stitching, we need a bin threshold *t* and a percentage tolerance value *p*. Typically, *t* is 6 where 6 means less than 6 different bins and *p* is typically 0.02 or 2% tolerance of absolute difference when comparing individual bins for matches.

Given two COIF features extracted from two images, the corresponding 256 divided by *k* bins from histogram *h* are compared for absolute distance value where a difference is detected if the absolute difference exceeds the value plus or minus the value times *p*.

```
c1 = COIF feature 1
c2 = COIF feature 2
binDistance = 0
p = 0.02
len = 256 / k
for i = 0 to len
  d1 = c1.distances[i]
  d2 = c2.distances[i]
  if (d2 < (d1·(1.0 − p)) || d2 > (d1·(1.0 + 0)) binDistance + 1
```

Figure 13

To improve the accuracy of matching, if the absolute difference of the value plus or minus the value times *p* exceeds some threshold *i*, where *i* is typically 40, then the bin distance is incremented as the match is considered poor.

Optionally, to increase the number of potential matches, if the absolute value of the difference of two bins is less than a threshold *m*, the bins may be considered identical, where *m* is typically 6.

If all bins from histogram *h* have been compared to be within *p* distance and the total number of different bins is less than the threshold *t*, the two COIF features are considered to be a match. The comparison may be terminated early to improve time efficiency if the threshold *t* is met as the two COIF features will not be a good match.

If less than 5 matches are found for a given *k*, or 80% of the feature matches are within 10% distance of each other, the feature matching process is repeated with a new *k* value that is doubled.

Parameter Name	Symbol	Description
Radius	*r*	The radius from interest point detector pixel to use as luminance values in the outermost COIF histogram. Typically 24, meaning 24 pixels.

Bin threshold	*t*	The threshold of different number of COIF bins to indicate a match. Typically 6, meaning less than 6 different bins.
Optional minimum bin threshold	*m*	An optional minimum bin threshold where if the absolute value of the difference of two bins is less than *m* the bins are a match. Typically 6.
Image flattening value	*d*	The value to multiply image luminance values by to flatten the image. Typically 0.5.
Matching threshold	*p*	The percentage of tolerance to match COIF bin values. Typically 0.02, meaning there is a +/- 2% tolerance to signal bins are identical.
Bin merge count	*k*	The number of bins to merge for the given histogram, which starts at 2 and is doubled for each necessary iteration.

Table 1

6. Results

Matching COIF features yields enough reliable matches to be used for image stitching given a range of affine transformations.

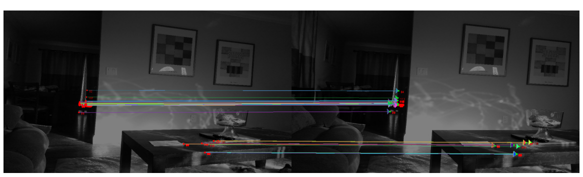

Figure 14

Figure 15

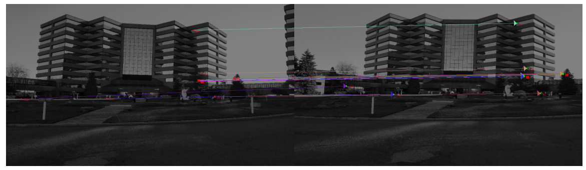

Figure 16

COIF is sensitive to blurs. Increasing the threshold t may yield more matches at the expense of introducing false positives. Further work, such as further computations on a scale pyramid of a given image to introduce scale invariance and refinements of the descriptor computation to be less sensitive to blurs may be researched.

References

[1] Hans P. Moravec. Techniques Towards Automatic Visual Obstacle Avoidance. https://frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1977/aip.txt

[2] Constantin Vertan and Nozha Boujemaa. Using Fuzzy Histograms and Distances for Color Image Retrieval.

[3] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An Efficient Dense Descriptor Applied to Wide Baseline Stereo.

[4] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features.

[6] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints.

[7] Ye Luo, Ping Xue, and Qi Tian. Image Histogram Constrained SIFT Matching.

[8] Martin A. Fischler, and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.