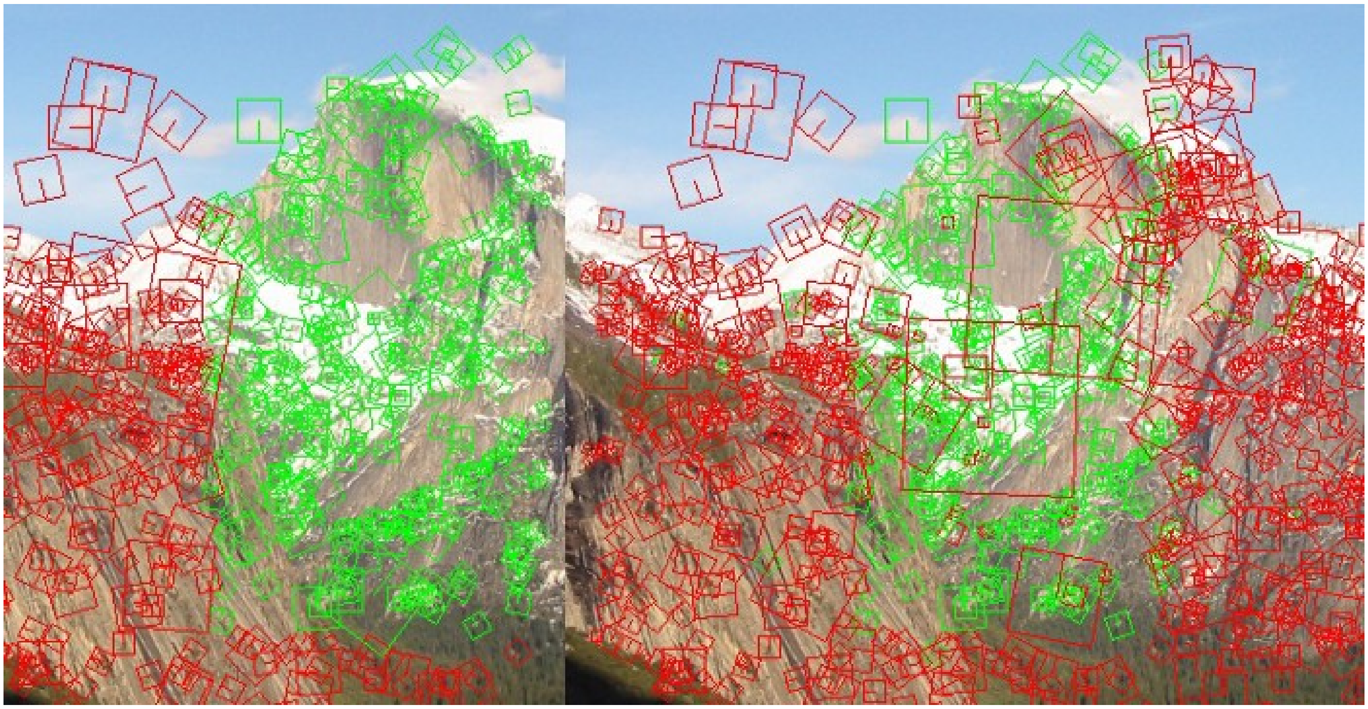
Introduction To Feature Detection And Matching

Deepanshu Tyagi Follow
Jan 3, 2019 · 5 min read

Feature detection and matching is an important task in many computer vision applications, such as structure-from-motion, image retrieval, object detection, and more. In this series, we will be talking about local feature detection and matching.



This is part of a 7-series Feature Detection and Matching. Other articles included

- Introduction to Harris Corner Detector
- Introduction to SIFT (Scale-Invariant Feature Transform)
- Introduction to SURF (Speeded-Up Robust Features)

- Introduction to FAST (Features from Accelerated Segment Test)
- Introduction to BRIEF (Binary Robust Independent Elementary Features)
- Introduction to ORB (Oriented FAST and Rotated BRIEF)

Application Of Feature Detection And Matching

- Automate object tracking
- Point matching for computing disparity
- Stereo calibration(Estimation of the fundamental matrix)
- Motion-based segmentation
- Recognition
- 3D object reconstruction
- Robot navigation
- Image retrieval and indexing

Feature

A feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image. The features can be classified into two main categories:

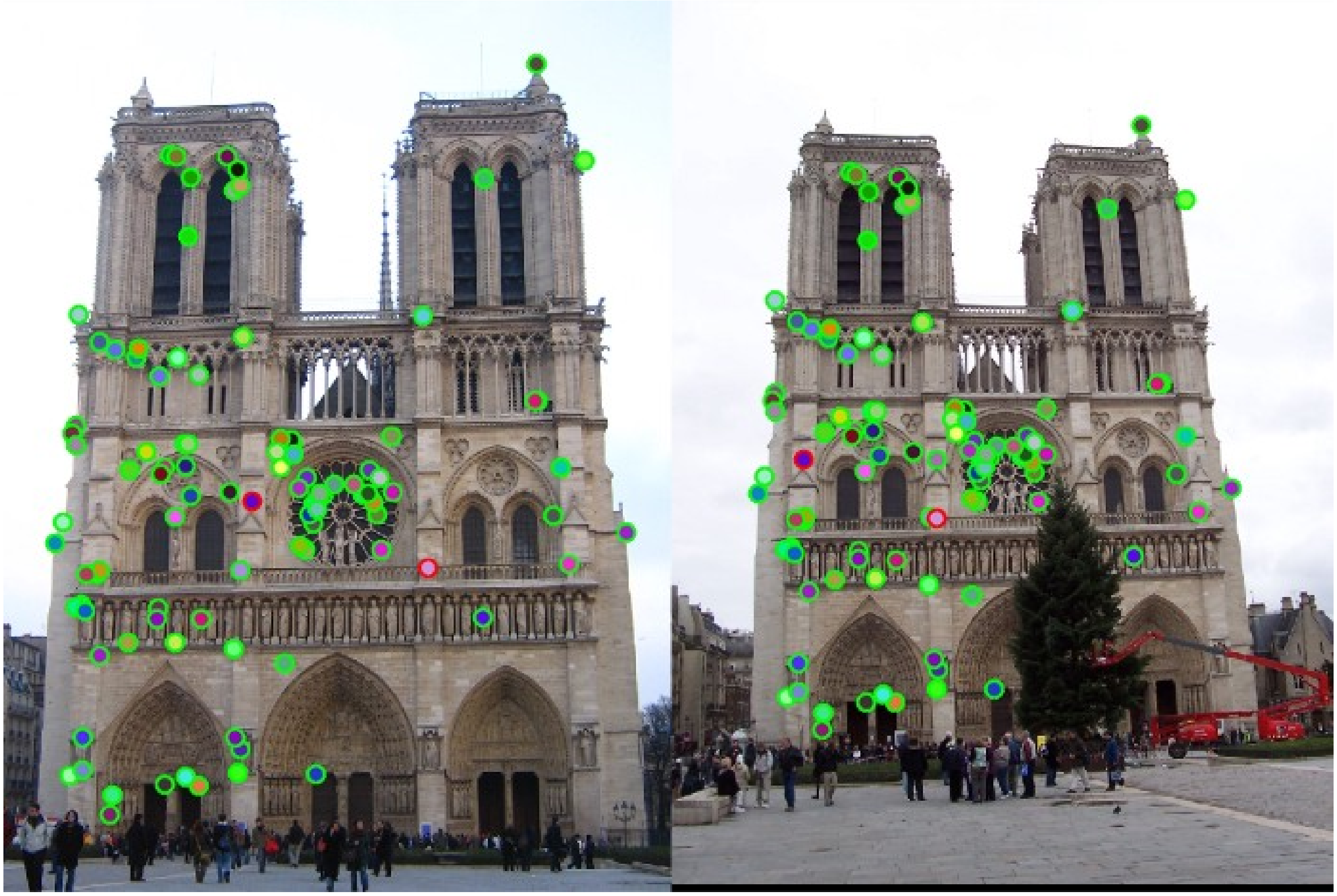
- The features that are in specific locations of the images, such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow. These kinds of localized features are often called **keypoint features** (or even corners) and are often described by the appearance of patches of pixels surrounding the point location.
- The features that can be matched based on their orientation and local appearance (edge profiles) are called **edges** and they can also be good indicators of object boundaries and occlusion events in the image sequence.

Main Component Of Feature Detection And Matching

- **Detection:** Identify the **Interest Point**
- **Description:** The local appearance around each feature point is described in some way that is (ideally) invariant under changes in illumination, translation, scale, and in-plane rotation. We typically end up with a descriptor vector for each feature point.
- **Matching:** Descriptors are compared across the images, to identify similar features. For two images we may get a set of pairs (***Xi, Yi***) ↔ (***Xi`, Yi`***), where (***Xi, Yi***) is a feature in one image and (***Xi`, Yi`***) its matching feature in the other image.

Interest Point

Interest point or Feature Point is the point which is expressive in texture. Interest point is the point at which the direction of the boundary of the object changes abruptly or intersection point between two or more edge segments.



Properties Of Interest Point

- It has a well-defined *position* in image space or well localized.

- It is *stable* under local and global perturbations in the image domain as illumination/brightness variations, such that the interest points can be reliably computed with a high degree of *repeatability*.
- Should provide efficient detection.

Possible Approaches

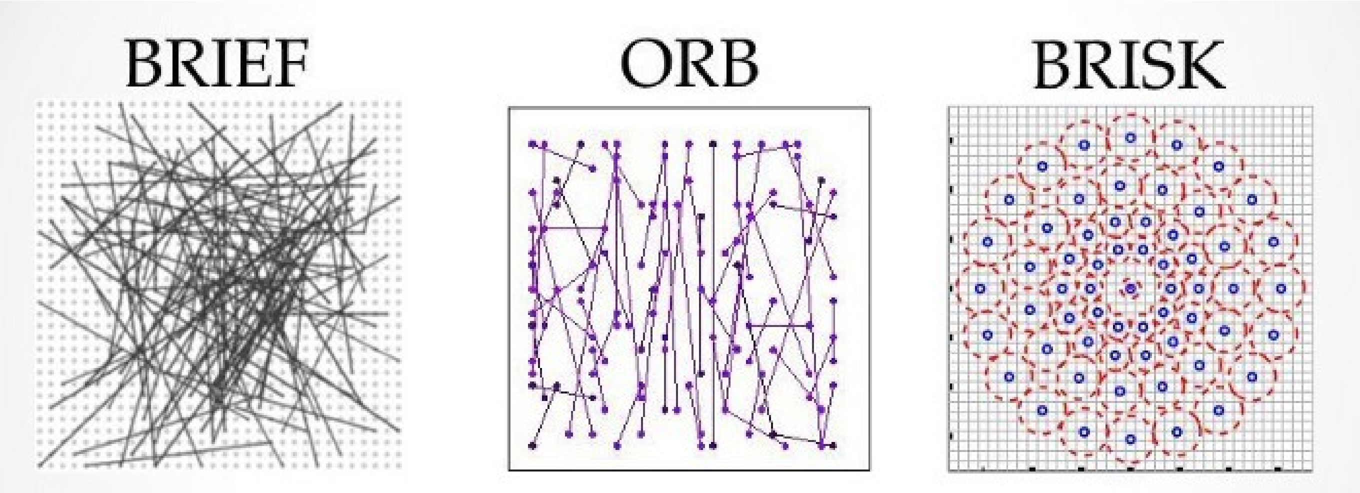
- Based on the brightness of an image(Usually by image derivative).
- Based on Boundary extraction(Usually by Edge detection and Curvature analysis).

Algorithms for Identification

- Harris Corner
- SIFT(Scale Invariant Feature Transform)
- SURF(Speeded Up Robust Feature)
- FAST(Features from Accelerated Segment Test)
- ORB(Oriented FAST and Rotated BRIEF)

Feature Descriptor

A feature descriptor is an algorithm which takes an image and outputs feature descriptors/feature vectors. Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical “fingerprint” that can be used to differentiate one feature from another.



Ideally, this information would be invariant under image transformation, so we can find the feature again even if the image is transformed in some way. After detecting interest point we go on to compute a descriptor for every one of them. Descriptors can be categorized into two classes:

- **Local Descriptor:** It is a compact representation of a point's local neighborhood. Local descriptors try to resemble shape and appearance only in a local neighborhood around a point and thus are very suitable for representing it in terms of matching.

- **Global Descriptor**: A global descriptor describes the whole image. They are generally not very robust as a change in part of the image may cause it to fail as it will affect the resulting descriptor.

Algorithms

- SIFT(Scale Invariant Feature Transform)

- SURF(Speeded Up Robust Feature)

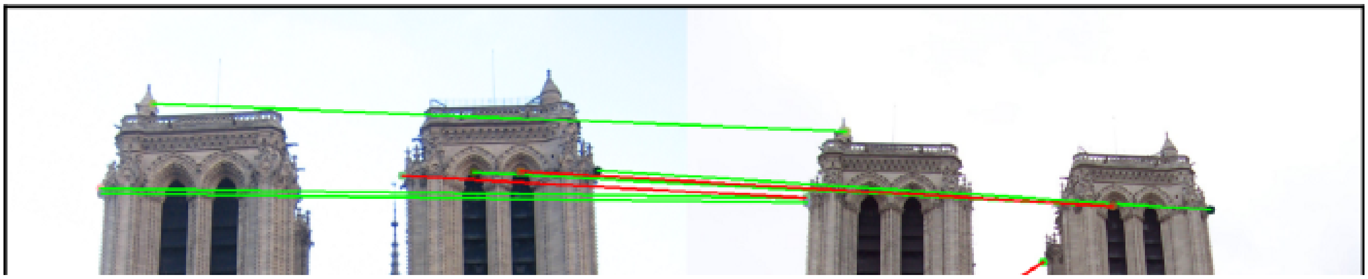
- BRISK (Binary Robust Invariant Scalable Keypoints)

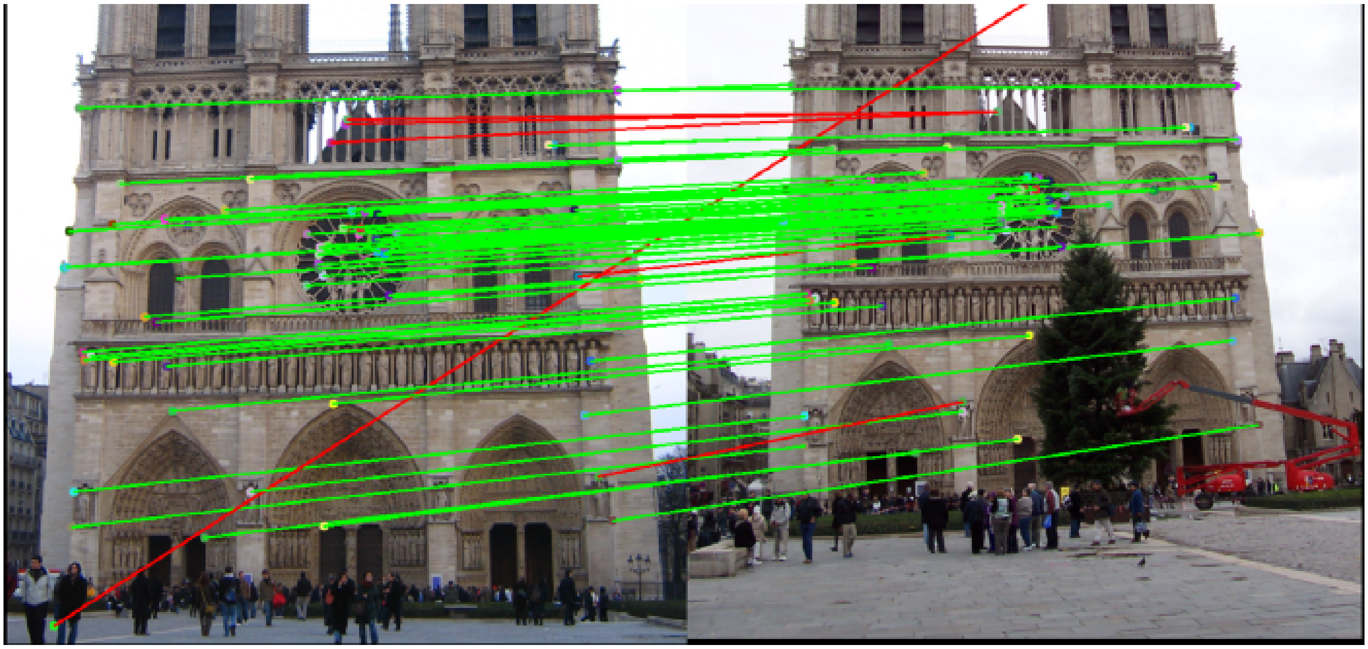
- BRIEF (Binary Robust Independent Elementary Features)

- ORB(Oriented FAST and Rotated BRIEF)

Features Matching

Features matching or generally image matching, a part of many computer vision applications such as image registration, camera calibration and object recognition, is the task of establishing correspondences between two images of the same scene/object. A common approach to image matching consists of detecting a set of interest points each associated with image descriptors from image data. Once the features and their descriptors have been extracted from two or more images, the next step is to establish some preliminary feature matches between these images.





Generally, the performance of matching methods based on interest points depends on both the properties of the underlying interest points and the choice of associated image descriptors. Thus, detectors and descriptors appropriate for images contents shall be used in applications. For instance, if an image contains bacteria cells, the blob detector should be used rather than the corner detector. But, if the image is an aerial view of a city, the corner detector is suitable to find man-made structures. Furthermore, selecting a detector and a descriptor that addresses the image degradation is very important.

Algorithms

- Brute-Force Matcher
- FLANN(Fast Library for Approximate Nearest Neighbors) Matcher

Algorithm For Feature Detection And Matching

- Find a set of distinctive keypoints
- Define a region around each keypoint
- Extract and normalize the region content
- Compute a local descriptor from the normalized region
- Match local descriptors

References

- Bradski and Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, 2008.
- M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In Proc. CVPR pages 510–517, 2005.
- H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In Proc. ECCV, pages 404–417, 2006.
- C. Harris and M. J. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–152, 1988.
- Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In Proc. CVPR, pages 506–513, 2004.
- Edward Rosten and Tom Drummond, "Machine learning for high speed corner detection" in 9th European Conference on Computer Vision, vol. 1, 2006, pp. 430–443.
- Edward Rosten, Reid Porter, and Tom Drummond, "Faster and better: a machine learning approach to corner detection" in IEEE Trans. Pattern Analysis and Machine Intelligence, 2010, vol 32, pp. 105–119.
- R. Szeliski. Computer Vision: Algorithms and Applications. Springer, 2010.
- https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html
- https://udacity.com/course/computer-vision-nanodegree--nd891

Thanks for reading! If you enjoyed it, hit that clap button below and follow Data Breach for more updates

Machine Learning Computer Vision Image Processing Image Recognition