Features Detection, Description and Matching For Image Processing

Ei Shwe Sin
University Of Computer Studies, Taunggyi,Myanmar
Eishwezin31@gmail.com

Abstract

Feature detection, description and matching are one of an essential component of many computer vision applications. In order to detect any object in the given scene it is important to know the key features that describe that object. A number of feature detection algorithm have been developed in recent year. However, the computational complexity and aaccuracy of feature matches limits the applicability of these algorithms. There are four widely used feature detection algorithms, Harris, SURF (Speeded-Up Robust Features), FAST (Features from Accelerated Segment) and FREAK (FAST Retina Key Point) feature detection algorithms. In this paper presented some practical approaches to detecting features, description method and also discuss feature matching.

Keywords: feature detector, corner detection, feature descriptor, feature matching.

1. Introduction

Features detection and description from static and dynamic scenes is an active area of research and one of the most studied topics in computer vision literature. The concept of feature detection and description refers to the process of identifying points in an image (interest points) that can be used to describe the image's contents such as Edges, corners, ridges and blobs. It is primarily aiming towards object detection, analysis and tracking from a video stream to describe the semantics of the its actions and behavior [1]. Some computer vision applications need to identify a set of points to be matched setting correspondences between images. Images share a common image processing pipeline. This pipeline can be split into several processes: The first step is interest point extraction. This process selects a group of pixels (regions) where their surrounding or neighboring pixels retain enough information, that allow the regions to be identified afterwards. Extracted keypoints are subjected to the next process converting the neighborhood of each point into a vector of values, known as descriptor. These descriptors act as identifiers of their corresponding interest points. The simplest descriptor consists in rearranging pixel values of a regular image patch surrounding a keypoint into a one-dimensional vector. Once every interest point is characterized by its corresponding descriptor, a matching process identifies corresponding points between images, looking for the most similar descriptors by using some distance function such as Euclidean, Mahalanobis or Hamming, among others [2]. The filtering process at the end of the pipeline is used for removing wrong keypoint matches. This process is carried out by applying temporal, spatial or geometric restrictions, allowing the identification of outliers with respect to an specific function, or model, such as homography estimation. [3] These processes allow to identify or to separate true correct matches from the set of matches.

It also has a long list of potential applications, which include, but is not limited to access control to sensitive building, crowd and population statistical analysis, human detection and tracking, detecting of suspicious actions, traffic analysis, vehicular tracking, and detection of military targets.

Generally , the performance of matching based on interest points depends on both the properties of the underlying interest points and the choice of associated image descriptors[4].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 corner detector. But, if the image is an aerial view of 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. For example, if there is no scale change present, a corner detector that does not handle scale is highly desirable; while, if image contains a higher level of distortion, such as scale and rotation, the more computationally intensive SURF feature detector and descriptor is a adequate choice in that case. For greater accuracy it is recommended to use several detectors and descriptors at the same time. [5]

2. Related Work

In this section, the feature detection techniques that have been used for image processing are briefly described. The algorithm used in detection technique, descriptor and feature matching has been explained.

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit and Pascal Fua1[10] introduced a novel Deep Network architecture that implements the full feature point handling pipeline, that is, detection, orientation estimation, and feature description.

Golightly and Jones [11] presented an algorithm for both corner detection and matching for visual tracking of power line inspection.

3. Proposed System Design the uses for feature detection, feature descriptor and feature matching

Images are taken from scene. And then preprossing the image used the filters for smoothing, sharpening, removing noise, and edge detection. In the feature detecting, identify points of interest in the image using desired detection method. After identified points of interest, the next step is to come up with a *descriptor* for the feature centered at each interest point. Now detected and described features, find the best matching features in other images.

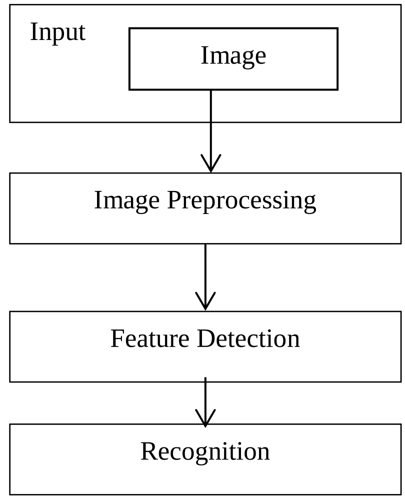

Figure 1. Proposed System Design

4. Feature Detection

Two types of image features can be extracted form image content representation; namely global features and local features. Global features (e.g., color and texture) aim to describe an image as a whole and can be interpreted as a particular property of the image involving all pixels. While, local features aim to detect key points or interest regions in an image and describe them. The feature to be detected is denoted by a set of parameters $P = (p_{1,}p_{2,.} \ldots p_n)$,where each parameter p_i has limited defined by R_i . For example, a line in 2D is denoted by two parameters, (ρ , θ), as $\rho = x \cos\theta + y \, sin\theta$ and the ranges of these parameters are

$$R_\rho \; : -r_{max} < \rho < r_{max} \qquad (1)$$

$$R_\theta \; : \qquad 0 < \theta < \pi \qquad (2)$$

Where $r_{max} = \sqrt{{s_x}^2 + {s_y}^2}$ and s_x and s_y are the dimensions of the image. [6]

The following properties are important for utilizing a feature detector in computer vision applications:

1. **Robustness**, the feature detection algorithm should be able to detect the same feature locations independent of scaling, rotation,

shifting, photometric, deformations, compression artifacts, and noise.
2. **Repeatability**, the feature detection algorithm should be able to detect the same features of the same scene or object repeatedly under variety of viewing conditions.
3. **Accuracy**, the feature detection algorithm should accurately localize the image features (same pixel locations), especially for image matching tasks, where precise correspondences are needed to estimate the epipolar geometry.
4. **Generality**, the feature detection algorithm should be able to detect features that can be used in different applications.
5. **Efficiency**, the feature detection algorithm should be able to detect features in new images quickly to support real-time applications.
6. **Quantity**, the feature detection algorithm should be able to detect all or most of the features in the image. Where, the density of detected features should reflect the information content of the image for providing a compact image representation [1].

4.1. Harris corner detector algorithm

Harris corner detection algorithm detects feature points by designing a local detecting window inside the image. The small amount of shifting of window in different direction can be determined by the average variation in the pixel density. The corner point is the center point of the window. Hence, on shifting the window in any of the direction, a large variation in pixel intensity is seen. When the window is shifted, no change in pixel intensity is seen in any direction if a flat region appears. But, when there is no change in pixel intensity along the edge direction, then an edge region is detected. But, when there is a significant change in pixel intensity in every direction, a corner is detected. A mathematical approach for determining whether the region found is flat, edge or corner is provided by Harris corner detection algorithm. More number of features are detected using this detection algorithm. Though, it is found to be scale variant, but it is invariant to rotation. The resulting detector based on the auto-correlation matrix is the most widely used technique. The 2×2 symmetric auto-correlation matrix used for detecting image features and describing their local structures [8]. Consider the Harris Detector Algorithm shown in Figure 2.

Step 1: Compute Gaussian derivatives at each pixel
Step 2: Compute auto correlation matrix in a Gaussian window around each pixel
Step 3: Compute corner response function *R*
Step 4: Threshold *R*
Step 5: Find local maxima of response function

Figure 2. Harris Detector Algorithm

4.2. Compute auto correlation matrix in a Gaussian window around each pixel

The change in pixel intensity for the shift[u,v] is given as below:

$$\in(\boldsymbol{u},\boldsymbol{v}) = \sum_{x,y} \boldsymbol{w}(\boldsymbol{x},\boldsymbol{y})[\boldsymbol{I}(\boldsymbol{x}+\boldsymbol{u},\boldsymbol{y}+\boldsymbol{v}) - \boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})]^2 \ \mathbf{(3)},$$

Where, $w(x, y)$ is a window function, $I(x, y)$ is the *intensity* of the individual pixel, and $I(x + u, y + v)$ is the pixel intensity after shift.

$$\in(u,v) \approx \sum_{x,y}[I(x,y) + uI_x + uI_y - I(x,y)]^2 \quad (4),$$

using Taylor expansion.

$$\in(u,v) \approx \sum_{x,y} u^2 I_x^2 + 2\,uvI_xI_y + v^2 I_y^2 \qquad (5),$$

by expanding the equation and cancelling properly.

$$\in(u,v) \approx [u \ \ v] \left(\sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right) \begin{bmatrix} u \\ v \end{bmatrix} \ (6)$$

$$\mathrm{M} = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \qquad (7),$$

expressed in a matrix form.

$$\in(u,v) \approx [u \ \ v]\, M \begin{bmatrix} u \\ v \end{bmatrix} \qquad (8)$$

Calculating the corners measure (R) for each pixel(x, y) by 2 eigenvalues λ_1, λ_2

$R = \det(M) - k * trace(M)^2$ (9)
where: $\det(M) = \lambda_1\lambda_2$ and $trace(M) = \lambda_1 + \lambda_2$ k (empirical constant) =0.04-0.06.We choose a local maximum point. The feature points whose pixel values are corresponding with the local maximum interest point are considered in Harris corner detection method. The detection of corner points is done after setting the threshold value T.

5. Image Feature descriptor

After a set of interest points has been detected from an image at a location $p(x, y)$,scale s, and orientation θ, their content or image structure in a neighborhood of p needs to be encoded in a suitable descriptor for discriminative matching and insensitive to local image deformations. The descriptor should be aligned with θ and proportional to the scale s. There are a large number of image feature descriptors in the literature; the most frequently used descriptor is discussed in the following sections.

5.1. Scale Invariant Feature Transform (SIFT)

SIFT(ScaleInvariant Feature Transformation) descriptor is one of the most successful approaches for feature or interest point extractor and description. SIFT algorithm has a local feature detector and local histogram-based descriptor. SIFT features are formed by computing the gradient at each pixel in a 16×16 window around the detected key point, using the appropriate level of the Gaussian pyramid at which the key point was detected. The gradient magnitudes are down weight gradients by a Gaussian fall-off function (blue circle) to reduce the influence of gradients for from the center. In each 4×4 quadrant, compute a gradient orientation histogram using 8 orientation histogram bins. The resulting 128 non-negative values form a raw version of the SIFT descriptor vector. To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.

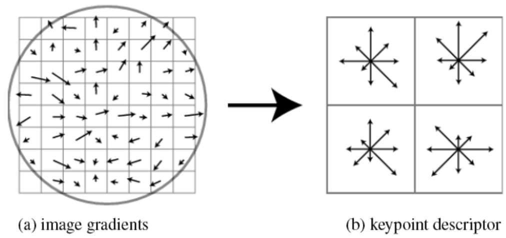

Figure 3. A schematic representation of Lowe's(2004) scale invariant feature transform(SIFT). (a) Gradient orientations and magnitudes are computed at each pixel and weighted by a Gaussian fall-off function (blue circle). (b) A weighted gradient orientation histogramis then computed in each subregion, using trilinear interpolation.

6. Feature Matching

Once we have extracted features and their descriptors, we need to match the features between these images. All the features that have been detected are matched so as to confirm that features are from the corresponding locations from completely different images. [7]

7. Matching Strategy and error rates

The simplest matching strategy is to set a threshold (maximum distance) and to return all matches for other images within this threshold. Setting the threshold too high results in to many false positive, i.e., incorrect matches being returned. Setting the threshold too low results in many false negatives, i.e., too many correct matches being missed.

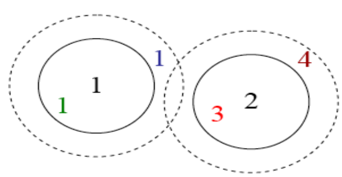

Figure 4. False positives and negatives

The black digits 1 and 2 are features being matched against a database of features in other images. At the current threshold setting (the solid circles), the green 1 is a true positive(good match), the blue 1 is a false negative(failure to match), and the red 3 is a false positive(incorrect match). If we set the threshold higher (the dashed circles), the blue 1 become a true positive but the brown 4 becomes an additional false positive.

The first counting the number of true and false matches and match failures. We can convert these numbers into unit rates by defining quantities. [9]

- True positive rate(TPR),

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{10}$$

- False positive(FPR),

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N} \tag{11}$$

- Positive predictive value(PPV),

$$PPV = \frac{TP}{TP + FP} = \frac{TP}{P} \tag{12}$$

- Accuracy(ACC),

$$ACC = \frac{TP + TN}{P + N} \tag{13}$$

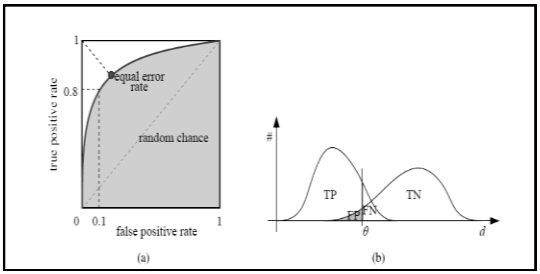

Figure 5.ROC curve and its related rates: (a) The ROC curve plots the true positive rate against the false positive rate for a particular combination of feature extraction and matching algorithms. Ideally, the true positive rate should be close to 1, while the false positive rate is close to 0. The area under the ROC curve (AUC) is often used as a single (scalar) measure of algorithm performance. Alternatively, the equal error rate is sometimes used. (b)The distribution of positives (matches) and negatives (non-matches) as a function of interfeature distance d.[9]

Any particular matching strategy (at a parti_ cular threshold or parameter setting) can be rated by the TPR and FPR numbers; ideally, the true positive rate will be close to 1 and the false positive rate close to 0. As we vary the matching threshold, we obtain a family of such points, which are collectively known as the receiver operating characteristic (ROC curve)(Figure .5(a)). The closer this curve lies to the upper left corner, i.e., the larger the area under the curve (AUC), the better its performance. Figure 5.(b) shows how we can plot the number of matches and non-matches as a function of inter-feature distance d.

8. Conclusion

In this paper the performance of features detection, description and matching are presented. This paper describes various efficient techniques of key point extraction. Key points support the capability to identify image correspondences despite of change in view conditions, occlusion, or the presence of clutter. Approaches to feature matching strategy and error rates will improve the feature detection challenges.

ACKNOLEDGMENTS

I would like to gratefully acknowledge to Reactor, Dr. Thinn Thu Naing for support and encouragement to write this paper. I ‘m also like to thank Dr. Nan Saw Kalayar, for her advice. Finally, I wish to thank my parents for their support and encouragement throughout my study.

References

[1] Awad, A.I; Hassaballah, M,[Eds],” ,Image Feature Detectors and Descriptors Foundations and Applications”,Springer, 2016.

[2] Barandiaran, I.; Cortes, C; Nieto, M.; Graña, M. and Ruiz, O. “A New Image Dataset and Evaluation Framework for Keypoint Extraction and Feature Descriptor Matching”. Proceedings of International Conf., on Computer Vision Theory and Application, (2013). pp. 252-257.

[3] O. Chum, T. Werner, and J. Matas, “Two-view geometry estimation unaffected by a dominant plane,” in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE,2005, pp. 772–779.

[4] C.Harris and M.Stephens."A Combined Corner and Edge Detector.“Proceedings of the 4th Alvey Vision Conf.,UK, 1988, pp.147-151.

[5] Gurbuz, A. C.,Cevher, V., and McClellan, J. H., “A compressive beam-former,” (Las Vegas, Nevada), March 30 –April 4 2008.
[6.] W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,”IEEE Transactions on Systems, Man, and Cybernetics, Part C ,Aug 2004,vol. 34, pp. 334–352.
[7]N. Paragios.Y.Chen and O.D Faugeras,”Handbook of Mathematical Models in Computer Vision” Springer,2005,pp.501-517.
[8] Pooja Ghosh, Achala Pandey* and Umesh C. Pati,”Comparison of Different Feature Detection Tech. for Image Mosaicing”,journal, Vol-1 Nov-2015.
[9] Richard Szeliski,” Computer Vision: Algorithms and Applications”, Springer 2010,pp.230-235.
[10] K. M. Yi, E. Trulls, V. Lepetit and P. Fua,”LIFT: LearnedInvariant Feature Transform” ,paper,European Conference, 2016.
[11] I. Golightly, and D. Jones, “Corner Detection and Matching for Visual Tracking during Power Line Inspection” Image and Vision Computing, Vol 21,pp.827_840.2003.