Explaining feature detection Mechanisms: A Survey

Ahmed A. Elngar[1], Mohamed Arafa[2], Mustafa Marouf[3], Mahmoud Ahmed[4*], Nehal Fawzy[5]

[1]Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef City, 62511, Egypt,

Email: elngar_7@yahoo.co.uk

[2]Teaching assistant, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni-Suef University of Computers and Artificial Intelligence, E-mail: jenuiman4rt@gmail.com

[3]Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni-Suef University of Computers and Artificial Intelligence, E-mail: mustafa.elhersh2@gmail.com

[4]Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni-Suef University of Computers and Artificial Intelligence, E-mail: ma7mouda7med2251999@gmail.com

[5]Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt, Email: Neehal.fawzy91@yahoo.com

(Corresponding Author: Mahmoud Ahmed[4*], ma7mouda7med2251999@gmail.com)

Abstract

Feature detection, description and matching are essential components of various computer vision applications; thus, they have received a considerable attention in the last decades. Several feature detectors and descriptors have been proposed in the literature with a variety of definitions for what kind of points in an image is potentially interesting (i.e., a distinctive attribute). This chapter introduces basic notation and mathematical concepts for detecting and describing image features. Then, it discusses properties of perfect features and gives an overview of various existing detection and description methods. Furthermore, it explains some approaches to feature matching. Finally, the chapter discusses the most used techniques for performance evaluation of detection algorithms.

Keywords: Interest points, Feature detector, Feature descriptor, Feature extraction , Feature matching

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021

DOI: 10.5281/zenodo.4898005

1. Introduction:

There is no universal or exact definition of what constitutes a feature, and the exact definition often depends on the problem or the type of application. Nevertheless, a feature is typically defined as an "interesting" part of an image, and features are used as a starting point for many computer vision algorithms. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is *repeatability*: whether or not the same feature will be detected in two or more different images of the same scene.

Feature detection is a low-level image processing operation. That is, it is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features. As a built-in pre-requisite to feature detection, the input image is usually smoothed by a Gaussian kernel in a scale-space representation and one or several feature images are computed, often expressed in terms of local image derivatives operations.

Occasionally, when feature detection is computationally expensive and there are time constraints, a higher-level algorithm may be used to guide the feature detection stage, so that only certain parts of the image are searched for features.

There are many computer vision algorithms that use feature detection as the initial step, so as a result, a very large number of feature detectors have been developed. These vary widely in the kinds of feature detected, the computational complexity and the repeatability.

In this paper, we will talk about many sections.in section (1), we discuss the definition of feature definition, feature detection. Then in section (2), we will explain the mechanism of feature detection. In section (3), we will talk about to the algorithms used in feature detection. We will talk about the basic idea and characteristics of each algorithm, how it works by steps, the Mathematics of each algorithm, the advantages, and disadvantages of each. And finally, we will give an example for each algorithm and a comparison among them.

1.2 Feature definition:

There is no universal or exact definition of what constitutes a feature, and the exact definition often depends on the problem or the type of application. Nevertheless, a feature is typically defined as an "interesting" part of an image. there are three types of feature: corner, edge, and region.

1.3. Feature detection.

Includes methods for computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions

2. Mechanism of feature detection: There are many ways to detect the first way by edge, the second by corner the last by region.

2.1. Detection by Edge:

Edges are points where there is a boundary (or an edge) between two image regions. In general, an edge can be of almost arbitrary shape, and may include junctions. In practice, edges are usually defined as sets of points in the image which have a strong gradient magnitude. Furthermore, some common algorithms will then chain high gradient points together to form a more complete description of an edge. These algorithms usually place some constraints on the properties of an edge, such as shape, smoothness, and gradient value.
Locally, edges have a one-dimensional structure.

2.2. Detection by Corner:

The terms corners and interest points are used somewhat interchangeably and refer to point-like features in an image, which have a local two-dimensional structure. The name "Corner" arose since early algorithms first performed edge detection, and then analyzed the edges to find rapid changes in direction (corners). These algorithms were then developed so that explicit edge detection was no longer required, for instance by looking for high levels of curvature in the image gradient. It was then noticed that the so-called corners were also being detected on parts of the image which were not

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

corners in the traditional sense (for instance a small bright spot on a dark background may be detected). These points are frequently known as interest points, but the term "corner" is used by tradition

2.3. Detection by region:

Blobs provide a complementary description of image structures in terms of regions, as opposed to corners that are more point-like. Nevertheless, blob descriptors may often contain a preferred point (a local maximum of an operator response or a center of gravity) which means that many blob detectors may also be regarded as interest point operators. Blob detectors can detect areas in an image which are too smooth to be detected by a corner detector.

Consider shrinking an image and then performing corner detection. The detector will respond to points which are sharp in the shrunk image, but may be smooth in the original image. It is at this point that the difference between a corner detector and a blob detector becomes somewhat vague. To a large extent, this distinction can be remedied by including an appropriate notion of scale. Nevertheless, due to their response properties to different types of image structures at different scales, the LoG and DoH blob detectors are also mentioned in the article on corner detection.

3. Feature detection algorithms:

There are many types of algorithms each of them can detect by edge or corner or region.

Feature detectors	**Edge**	**Corner**	**Blob**
Canny	Yes	NO	No
Sobel	Yes	No	No
Kayyali	Yes	No	No
Harris	Yes	Yes	No
Susan	Yes	Yes	No
Shi	No	Yes	No
Level curve curvature	No	Yes	No
Fast	No	Yes	Yes
Laplacian of Gaussian	No	Yes	Yes
Difference of Gaussian	No	Yes	Yes
Determinant of Hessian	No	Yes	Yes
MSER	No	No	Yes
PCBR	No	No	Yes
Gray-level blobs	No	No	Yes

But we will discuss one algorithm for each type of features. We will choose Harris algorithm because it detects by corner, canny algorithm because it detects by edge and finally difference of gaussians because it detects by region or blobs.

3.1. Harris corner detection

It's a mathematical approach to detect a corner in an image

3.1.1. Steps of Harris corner detection.

Harris improved upon Morava's corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches. (This corner score is often referred to as autocorrelation, since the term is used in the paper in which this detector is described. However, the mathematics in the paper clearly indicate that the sum of squared differences is used.) Without loss of generality, we will assume a grayscale 2-dimensional image is used. Let this image be given by I. Consider taking an image patch over the area (u, v) and shifting it by (x, y). The weighted *sum of squared differences* (SSD) between these two patches, denoted S, is given by:

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

$$m(x,y) = \sum_{u,v} w(u,v) * \begin{matrix} Ix^2(x,y) & IxIy(x,y) \\ IxIy(x,y) & Iy^2(x,y) \end{matrix} \qquad (1)$$

If we rewrite this equation into a matrix form, we will get something similar.

For small shifts [u, v] we have a bilinear approximation:

$$E(U,V) \equiv [U,V] \; M \begin{bmatrix} U \\ V \end{bmatrix} \qquad (2)$$

Where M is a 2 * 2 matrix computed from image derivatives:

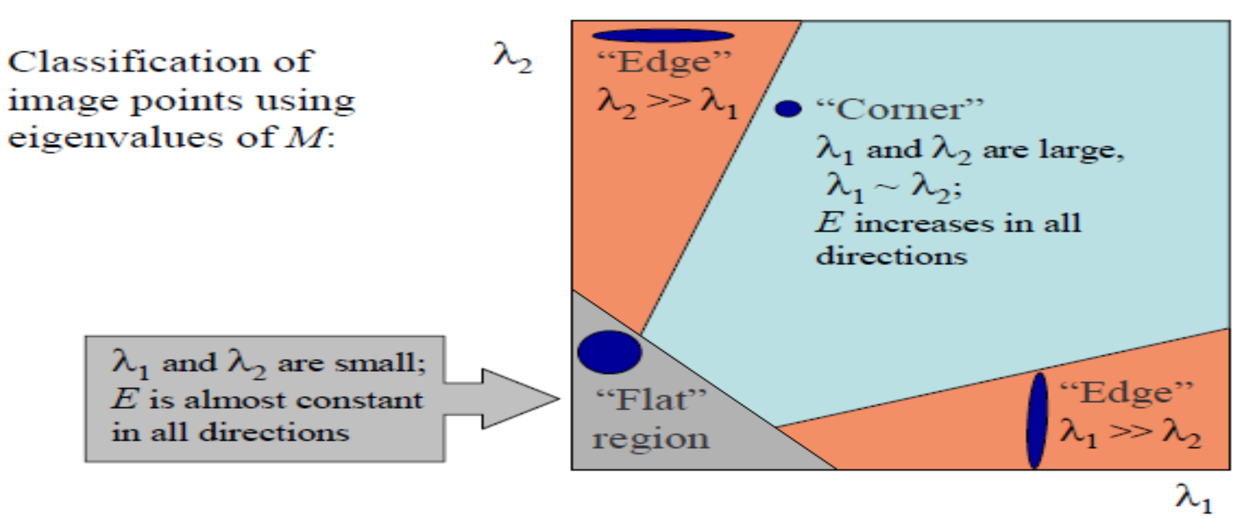
$$M = \sum_{x,y} w(x,y) \begin{matrix} Ix^2 & IxIy \\ IxIy & Iy^2 \end{matrix} \qquad (3)$$

Note that: we now have the equation where W is windows function (usually equal 1).

$$R = detM - k(traceM)^2 \quad (4)$$

$$trace\ M = \lambda 1 + \lambda 2 \quad (5)$$
$$detM = \lambda 1 \lambda 2 \quad (6)$$

- If R is positive and large then there will be a corner, otherwise not
- If one lambda is greater than the other, then R <0 So there will be an edge
- If R is very small, then there will be a flat region



DOI: 10.5281/zenodo.4898005

Fig 1. Harris corner detection

3.1.2. Advantages of Harris corner detection.

- Invariance of rotation
- Invariance of brightness
- Invariance of translation

3.1.3. Disadvantages of Harris corner detection

- Non invariant in scaling.

3.2. Canny edge detection.

Is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

3.2.1. Steps of Harris corner detection.
The canny edge detection algorithm is composed of steps

- Noise reduction.
- Calculating the intensity gradient.
- Non-maximum suppression.
- Hysteresis thresholding.

3.2.1.1. Noise reduction.

Using Gaussian filter to reduce noises, is widely used in the field of image processing. It is used to reduce the noise of an image. In this article we will generate a 2D Gaussian Kernel. The 2D Gaussian Kernel follows the below given Gaussian Distribution.

$$G(X,Y) = \frac{1}{2\pi\partial^2} e^{-\frac{X^2+Y^2}{2\partial^2}} \qquad (7)$$

3.2.1.2. Calculating the intensity gradient.

Using sobel kernel in the vertical and horizontal direction.

$$\begin{matrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{matrix}$$

G(X)

$$\begin{matrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

G(Y)

$$G = \sqrt{G(X)2 + G(Y)2} \qquad (8)$$

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

3.2.1.3. Non-maximum suppression.

- Full scan of image is the first step here : remove unwanted pixels
- Check if point is the local maximum ,yes consider for next stage or suppress it to zero
- Result : binary image with thin edges

3.2.1.4. Hysteresis thresholding.

- It takes to inputs: minVal & maxVal.
- Below this min value is discarded and the ones above are kept.
- If in between, cross checked whether it is connected to a sure edge or not.
- Random small noise pixels are removed.

3.2.2. Advantages of canny edge detection algorithm.

- The presence of Gaussian filter allows removing of any noise in an image.
- Detects the edges in a noisy state by applying the thresholding method.

3.2.3. Disadvantages of canny edge detection.

- The primary disadvantage of using canny edge detector is that it consumes a lot of time due to its complex computation.
- It is difficult to implement to reach the real-time response.

3.3. Difference of Gaussian region/blob detection Algorithm:
Technique is a highly influential approach to detecting Blob features in a scale invariant manner.

3.3.1. Steps of Gaussian region/blob detection Algorithm:

- Convert the image to grayscale to focus on the intensity and not on individual color channels
- The process involves running Gaussian convolutions in various images
- Then subtracting adjacent images from each other.
- The resulting image is one that highlights edges and blob features.
- We then look at all of the pixels across all of the scales to find the local extremes — the pixels which represent the greatest amount of change
- The last step is to filter out any unwanted features by using Threshold

3.3.1.1. Convert the image to grayscale to focus on the intensity and not on individual color channels

- - look N x N regions in an image and calculate average intensities inside / outside radius R
- - light blob: average inside >> average outside
- - Dark blob: average inside << average outside

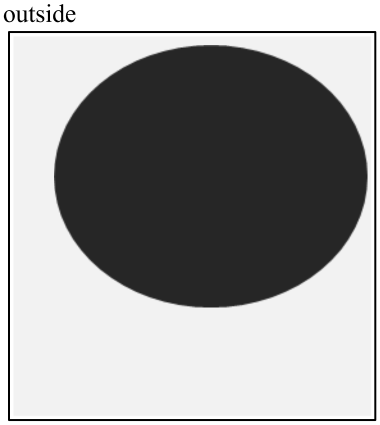


Fig2. Example Convert the image to grayscale

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

3.3.1.2. The process involves running Gaussian convolutions in various images:

$$G(X,Y) = \frac{1}{2\pi\partial^2} e^{-\frac{X^2+Y^2}{2\partial^2}} \quad (9)$$

Fig 3. Example Gaussian Blur

3.3.1.3. Then subtracting adjacent images from each other:

Fig 4. Example Difference of Sigma (small)

3.3.1.4. The resulting image is one that highlights edges and blob features:
We can repeat this process on another two sets of sigmas as seen below.

Fig 5, Example Difference of Sigma

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

3.3.1.5. We then look at all of the pixels across all of the scales to find the local extremes — the pixels which represent the greatest amount of change:

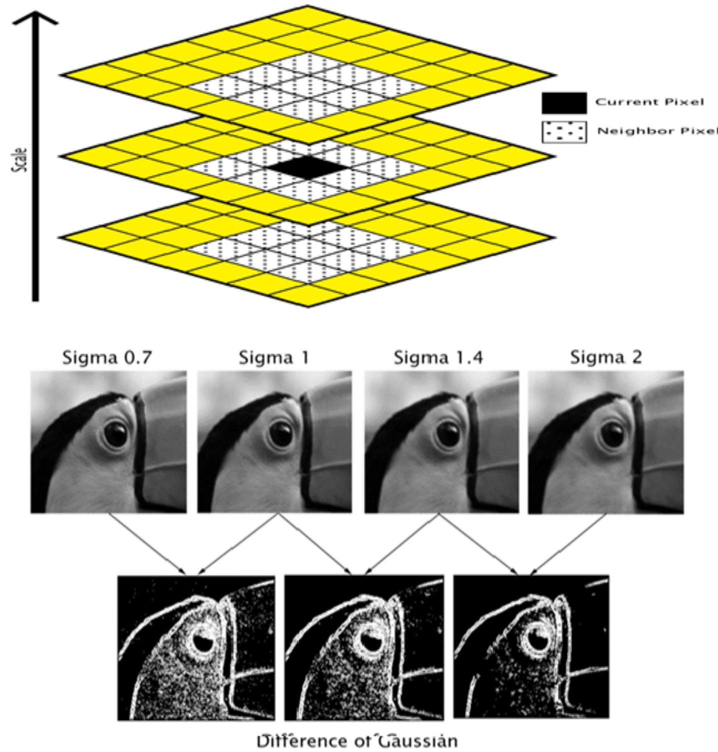


Fig 6. Example Difference of Gaussian

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

3.3.2. Advantages of Difference of Gaussian region/blob detection Algorithm:

- The presence of Gaussian filter allows reducing of any noise in an image.
- The larger circles are pointing out features which are still visible even after the image has been blurred significantly.
- Smaller circles are showing us features which are only prominent in weaker blurs.

3.3.3. Disadvantages of Difference of Gaussian region/blob detection Algorithm:

- remove edge features

3.4. We showed an example for each feature but we will show an algorithm which solved the problems of scaling (Harris algorithm can't solve it).

This algorithm is called scale invariant feature transformation (SIFT).

3.4.1. **Sift**: is an algorithm in computer vision used to detect and describe local features in images.

3.4.2. **Steps of sift algorithm:**

- **Scale space Extrema Detection:** we can't use the same window to detect key points with different scale. To detect larger corners we need larger windows. For this, scale space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values.
- **Key point Localization**: Once potential key point's locations are found, they have to be refined to get more accurate results.
- **Orientation Assignment:** Now an orientation is assigned to each key point to achieve invariance to image rotation.
- Key point Descriptor
- Key point Matching

3.4.2.1. **Scale space:**

SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different \sigma, let it be \sigma and k\sigma. This process is done for different octaves of the image in Gaussian Pyramid. It is represented in below image:

Approximation of LOG by Difference of Gaussian

$$\rightarrow \frac{\partial G}{\partial \sigma} = \sigma \Delta G \quad (10)$$

Heat Equation

$$\rightarrow \sigma \Delta G = \frac{\partial G}{\partial \sigma} = \frac{G((x,y,k\sigma) - G(x,y,\sigma)}{k\sigma - \sigma} \quad (11)$$

$$\rightarrow G(x,y,k\sigma) - G(x,y,\sigma) = (k-1)\sigma^2 \Delta G \quad (12)$$

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

Typical values of σ = 1.6, K=$\sqrt{2}$

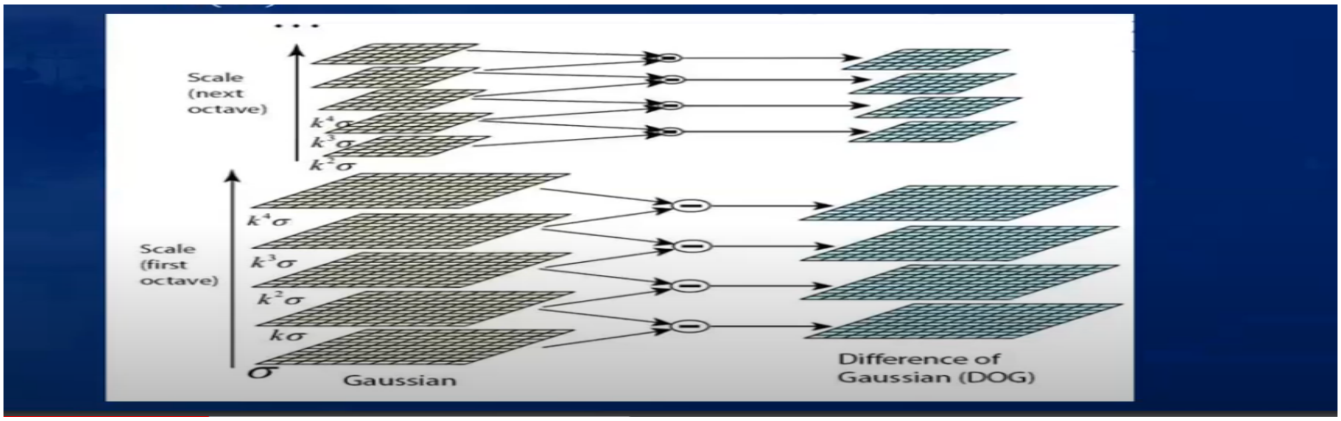
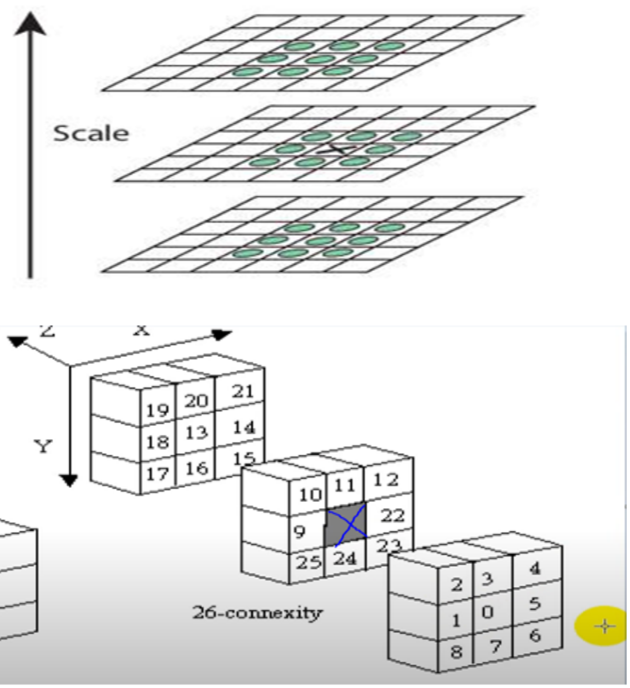


Fig 7: Example Scale space

3.4.2.2. Key point localization:

- Once this Difference of Gaussian (DoG) are found, images are searched for local extrema over scale and space. For one pixel in an image is compared with its 8 neighbors as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in below image:

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005



Note: number of octaves = 4, number of scale levels = 5, initial \sigma=1.6, k=\sqrt {2} etc as Optimal values.

Fig 8, Example Key point localization

3.4.2.3. Orientation assignment:

- Now an orientation is assigned to each keypoint to achieve invariance to image rotation.

3.4.3. Advantages of sift algorithm:

- **Invariant in scaling**
- **Invariant in rotation**
- **Features are robust**
- **More efficient compared to older algorithms**

3.4.4. Disadvantages of sift algorithm:

- **Still quite slow (SURF provides similar performance while running faster)**
- **Generally, doesn't work well with lighting changes and blur**

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

3.5. Comparison between algorithms:

Algorithm	**Advantages**	**Disadvantages**
Harris	1-Invariance of rotation 2-Invariance of brightness 3-Invariance of translation	Non invariant in scaling
Canny	1-The presence of Gaussian filter allows removing of any noise in an image. 2-Detects the edges in a noisy state by applying the thresholding method.	1-The primary disadvantage of using Canny edge detector is that it consumes a lot of time due to its complex computation. 2- It is difficult to implement to reach the real-time response.
Dog	1-The presence of Gaussian filter allows reducing of any noise in an image. 2-The larger circles are pointing out features which are still visible even after the image has been blurred significantly. 3-Smaller circles are showing us features which are only prominent in weaker blurs	1-remove edge features
Sift	1-Invariant in scaling 2-Invariant in rotation 3-Features are robust 4-More efficient compared to older algorithms	1-Still quite slow (SURF provides similar performance while running faster) 2-Generally doesn't work well with lighting changes and blur

4. Conclusion:

We have showed a feature detection in computer vision. Beginning of feature types and mentioned four types of algorithms which detect images by different types. Feature types like (corner , edge , region or blob) and we talked about many typed of algorithms like (Harris , Canny , Dog , sift), Each of them has some advantages & disadvantages I know this has been really long and probably not all that clear in places, finally we introduced a comparison among them . But I advise you to use sift algorithm because it has a lot of features and high efficiency than any algorithm additional to its disadvantages are very little so I hope you learned something.

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021
DOI: 10.5281/zenodo.4898005

References

[1] *T. Lindeberg (2008–2009). "Scale-space". In Benjamin Wah (ed.). Encyclopedia of Computer Science and Engineering.* ***IV****. John Wiley and Sons. pp. 2495–2504. doi:10.1002/9780470050118.ecse609. ISBN 978-0470050118.* (summary and review of a number of feature detectors formulated based on a scale-space operations)

[2] *Canny, J. (1986). "A Computational Approach To Edge Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence.* ***8*** *(6): 679–714. doi:10.1109/TPAMI.1986.4767851*. (Canny edge detection)

[3] *C. Harris; M. Stephens (1988). "A combined corner and edge detector" (PDF). Proceedings of the 4th Alvey Vision Conference. pp. 147–151.*(Harris/Plessey corner detection)

[4] *S. M. Smith; J. M. Brady (May 1997). "SUSAN - a new approach to low level image processing". International Journal of Computer Vision.* ***23*** *(1): 45–78. doi:10.1023/A:1007963824710.*(The SUSAN corner detector)

[5] *J. Shi; C. Tomasi (June 1994). "Good Features to Track,". 9th IEEE Conference on Computer Vision and Pattern Recognition. Springer.*(The Shi and Tomasi corner detector)

[6] *E. Rosten; T. Drummond (2006). "Machine learning for high-speed corner detection". European Conference on Computer Vision. Springer. pp. 430–443. CiteSeerX 10.1.1.60.3991. doi:10.1007/11744023_34.*(The FAST corner detector)

[7] *T. Lindeberg (1998). "Feature detection with automatic scale selection" (abstract). International Journal of Computer Vision.* ***30*** *(2): 77–116. doi:10.1023/A:1008045108935.*(Laplacian and determinant of Hessian blob detection as well as automatic scale selection)

[8] *D. Lowe (2004). "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision.* ***60*** *(2): 91. CiteSeerX 10.1.1.73.2924. doi:10.1023/B:VISI.0000029664.99615.94.*(DOG blob detection with automatic scale selection)

[9] *J. Matas; O. Chum; M. Urban; T. Pajdla (2002). "Robust wide baseline stereo from maximally stable extremum regions" (PDF). British Machine Vision Conference. pp. 384–393.*(The MSER blob detector)

[10] *T. Lindeberg (1993). "Detecting Salient Blob-Like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention" (abstract). International Journal of Computer Vision.* ***11*** *(3): 283–318. doi:10.1007/BF01469346.*(Grey-level blob detection and scale-space blobs)

[11] R. Haralick, "Ridges and Valleys on Digital Images," Computer Vision, Graphics, and Image Processing vol. 22, no. 10, pp. 28–38, Apr. 1983. (Ridge detection using facet model)

[12] J. L. Crowley and A. C. Parker, "A Representation for Shape Based on Peaks and Ridges in the Difference of Low Pass Transform", IEEE Transactions on PAMI, PAMI 6 (2), pp. 156–170, March 1984. (Ridge detection based on DOGs)

[13] D. Eberly, R. Gardner, B. Morse, S. Pizer, C. Scharlach, Ridges for image analysis, Journal of Mathematical Imaging and Vision, v. 4 n. 4, pp. 353–373, Dec. 1994. (Fixed scale ridge detection)

[14] *T. Lindeberg (1998). "Edge detection and ridge detection with automatic scale selection" (abstract). International Journal of Computer Vision.* ***30*** *(2): 117–154. doi:10.1023/A:1008097225773.*(Ridge detection with automatic scale selection)

[15] *T. Lindeberg (2008–2009). "Scale-space". In Benjamin Wah (ed.). Encyclopedia of Computer Science and Engineering.* ***IV****. John Wiley and Sons. pp. 2495–2504. doi:10.1002/9780470050118.ecse609. ISBN 978-0470050118.* (summary and review of a number of feature detectors formulated based on a scale-space operations)

[16] *Canny, J. (1986). "A Computational Approach To Edge Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence.* ***8*** *(6): 679–714. doi:10.1109/TPAMI.1986.4767851.*. (Canny edge detection)

[17] *C. Harris; M. Stephens (1988). "A combined corner and edge detector" (PDF). Proceedings of the 4th Alvey Vision Conference. pp. 147–151.*(Harris/Plessey corner detection)

[18] *S. M. Smith; J. M. Brady (May 1997). "SUSAN - a new approach to low level image processing". International Journal of Computer Vision.* ***23*** *(1): 45–78. doi:10.1023/A:1007963824710.*(The SUSAN corner detector)

[19] *J. Shi; C. Tomasi (June 1994). "Good Features to Track,". 9th IEEE Conference on Computer Vision and Pattern Recognition. Springer.*(The Shi and Tomasi corner detector)

[20] *E. Rosten; T. Drummond (2006). "Machine learning for high-speed corner detection". European Conference on Computer Vision. Springer. pp. 430–443. CiteSeerX 10.1.1.60.3991. doi:10.1007/11744023_34.*(The FAST corner detector)

[21] *T. Lindeberg (1998). "Feature detection with automatic scale selection" (abstract). International Journal of Computer Vision.* ***30*** *(2): 77–116. doi:10.1023/A:1008045108935.*(Laplacian and determinant of Hessian blob detection as well as automatic scale selection)

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021

DOI: 10.5281/zenodo.4898005

[22] *D. Lowe (2004). "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision.* ***60*** *(2): 91. CiteSeerX 10.1.1.73.2924. doi:10.1023/B:VISI.0000029664.99615.94.*(DOG blob detection

Received: November 24, 2020 Revised: February 9, 2021 Accepted: March 15, 2021

DOI: 10.5281/zenodo.4898005