**Scale Invariant Feature Transform**

-A DEEKSHANA

**ABSTRACT:**

“If you fall asleep now, you will dream. If you study now, you will live your dream!”

Before we start to read, we must have a idea on what is transform? In simple way, data transformation makes your data useful. Transform it into data that’s joined together, dimensionally modeled, de-normalized, and ready for analysis.

There is a great deal of system dealing with image processing that are being used and developed on a daily basis. Those systems needed to development of some basic operation such as detecting their properties. The efficiency of this algorithm is its performance in the process of detection and property description, and that is due to the fact that it operates on a big number of key-points, the only drawback it has is that it is rather time consuming.

**CONTENT:**

Introduction

Scale-invariant interest point from scale-space extrema

Application

Robot localization and mapping

3D scene modelling

Analysis the human brain in 3D magnetic resonance images

Feature extracting

Result

Conclusion

Reference link

GitHub link

**INTRODUCTION:**

Object identification is the task that is concerned with finding a specific item in an image or video sequence. It is considered as an important issue facing

systems of computer vision. Generally, the process of object identification has the following phases: property deduction and property matching. In

computer vision, the Scale-Invariant Feature Transform (SIFT) is an algorithm for the identification and characterization of local image properties. It offers the ability of precise object recognition with low possibility of mismatching and is easy to be matched against a large data-base of local properties.

**SCALE-INVARIANT INTEREST POINTS FROM SCALE-SPACE EXTREMA:**

The original SIFT descriptor was computed from the image intensities around interesting locations in the image domain which can be referred to as interest points, alternatively key points. These interest points are obtained from scale-space extrema of differences-of-Gaussians within a difference-of Gaussians pyramid.

The difference-of-Gaussians approach proposed by Lowe constitutes a computationally efficient way to compute approximations of such Laplacian interest points. Another way of detecting scale-space extrema of the Laplacian efficiently for real-time implementation has been presented by Linde-berg and Bret-zner based on a hybrid pyramid. A closely related method for real-time scale selection has been developed by Crowley and Riff.

**APPLICATION:**

The scale invariant feature transform (SIFT) with its related image descriptors in terms of histograms of receptive field-like image operations have opened up an area of research on image-based matching and recognition with numerous application areas. Being based on theoretically well-founded scale-space operations or approximations thereof, these approaches have been demonstrated to allow for robust computation of image features and image descriptors from real-world image data.

MULTI-VIEW MATCHING:

The SIFT descriptor with its associated matching methods can be used for establishing point matches between different views of a 3-D object or a scene. By combining such correspondences with *multi-view geometry* , 3-D models of objects and scenes can be constructed.

Similar methods for establishing multi-view correspondences can also be used for synthesizing novel views of a 3-D object/scene given a set of other views of the same object/scene (view interpolation) or for combining multiple partially overlapping images of the same scene into wider .

OBJECT RECOGNITION:

In his pioneering work on object recognition using the SIFT operator, Lowe demonstrated that robust and efficient recognition of objects in natural scenes can be performed based on collections of local image features. In close relation to this, a growing area of research has been developed concerning so-called bag of words methods and related methods for recognizing objects in real-world scenarios.

Features are matched to the SIFT feature database obtained from the training images. This feature matching is done through a Euclidean-distance based nearest neighbor approach. To increase robustness, matches are rejected for those key points for which the ratio of the nearest neighbor distance to the second-nearest neighbor distance is greater than 0.8. This discards many of the false matches arising from background clutter. Finally, to avoid the expensive search required for finding the Euclidean-distance-based nearest neighbour, an approximate algorithm called the best-bin-first algorithm is used.

Besides the specific area of object recognition, these types of methods can also be used for related tasks such as

OBJECT CATEGORY CLASSIFICATION:

Whereas the task of recognizing a previously seen object in a scene can be effectively addressed using the SIFT descriptor or the other closely related image descriptors described in this survey, the task of classifying previously unseen objects into object categories has turned out to be a harder problem. In the research to develop such methods, object categorization in terms of dense SIFT is as of 2012 still one of the better approaches.

ROBOTICS:

For a robot that moves in a natural environment, image correspondences in terms of SIFT features or related image descriptors can be used for tasks such as localizing the robot with respect to a set of known references, mapping the surrounding from image data that are acquired as the robot moves recognizing and establishing geometric relations to objects in the environment for robot manipulation.

**ROBOT LOCALIZATION AND MAPPING:**

 As the robot moves, it localizes itself using feature matches to the existing 3D map, and then incrementally adds features to the map while updating their 3D positions using a Kalman filter. This provides a robust and accurate solution to the problem of robot localization in unknown environments. Recent 3D solvers leverage the use of keypoint directions to solve trinocular geometry from three keypointsand absolute pose from only two keypoints, an often disregarded but useful measurement available in SIFT.

Modern [self driving cars](https://en.wikipedia.org/wiki/Self_driving_cars" \o "Self driving cars) mostly simplify the mapping problem to almost nothing, by making extensive use of highly detailed map data collected in advance. This can include map annotations to the level of marking locations of individual white line segments and curbs on the road. Location-tagged visual data such as Google's [StreetView](https://en.wikipedia.org/wiki/StreetView" \o "StreetView) may also be used as part of maps. Essentially such systems simplify the SLAM problem to a simpler [localisation](https://en.wikipedia.org/wiki/Robot_localization) only task, perhaps allowing for moving objects such as cars and people only to be updated in the map at runtime.

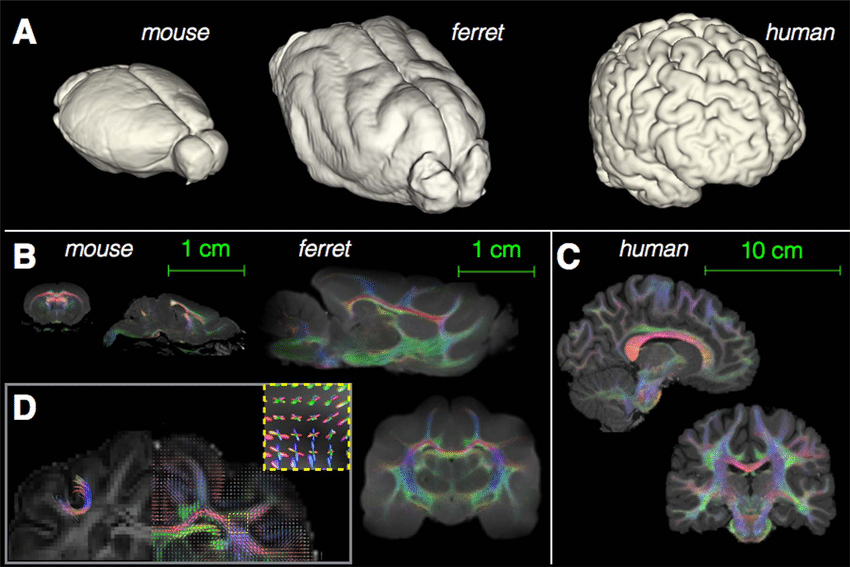
**3D SCENE MODELING:**

SIFT matching is done for a number of 2D images of a scene or object taken from different angles. This is used with [bundle adjustment](https://en.wikipedia.org/wiki/Bundle_adjustment) initialized from an [essential matrix](https://en.wikipedia.org/wiki/Essential_matrix) or [trifocal tensor](https://en.wikipedia.org/wiki/Trifocal_tensor) to build a sparse 3D model of the viewed scene and to simultaneously recover camera poses and calibration parameters. Then the position, orientation and size of the virtual object are defined relative to the coordinate frame of the recovered model. Explore some examples and here and explain how 3D models can be used on markers with AR.js in a video conference. Explain how these 3D models got increased importance during home schooling scenarios.

For online [match moving](https://en.wikipedia.org/wiki/Match_moving), SIFT features again are extracted from the current video frame and matched to the features already computed for the world mode, resulting in a set of 2D-to-3D correspondences. These correspondences are then used to compute the current camera pose for the virtual projection and final rendering. A regularization technique is used to reduce the jitter in the virtual projection.

**Analysing the Human Brain in 3D Magnetic Resonance Images:**

Feature-based Morphometry models the image probabilistically as a collage of independent features, conditional on image geometry and group labels, e.g. healthy subjects and subjects with Alzheimer's disease (AD). Features are first extracted in individual images from a 4D difference of Gaussian scale-space, then modeled in terms of their appearance, geometry and group co-occurrence statistics across a set of images.



The experimented results produced by the proposed technique depicted for the segmented outcome for the three classes of WM, GM, and CSF and for the extracted tumour region. The experimental results also find dice overlap image, indicating the comparison between the algorithm output and ground truth.

**Feature Extraction:**

A set of local properties of an image is extracted from every image. Each of

those properties includes a registry of:

1.Position, or pixel location (x, y), of the image.

2.Scale, characterized by the standard

deviation σ.

3.Orientation, the dominating direction of the image structure in neighbourhood.

4.Detailing of the image’s local structure, characterized according to gradient histograms.

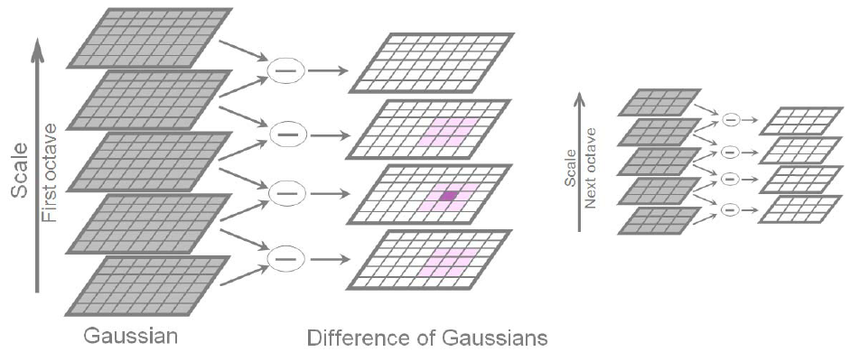
Implemented for generating the group of properties are:

• Detecting the local extrema of the

scale-space- the property positions are set as the local extremal of DOG pyramid. For building the pyramid of DOG the input image undergoes convolution in an iterative manner with a Gaussian kernel of σ = 1.6. The last convolved image undergoes down-sampling in every one of the image directions by factor equal to 2, and the convolving procedure is done again. This process is performed over and over until the down sampling is no longer possible. Every group of

images of identical sizes is known as an octave. All octaves construct together

the Gaussian pyramid that is given by a three dimensional function L (x ,y, σ).



The DOG pyramid D(x,y, σ) is

calculated from the difference of every neighbouring images in Gaussian pyramid. The local extrema (maxima or minima) of DOG function are found via the comparison of every

one of the pixels with its 26 adjacent pixels in the scale-space (8 neighbours in

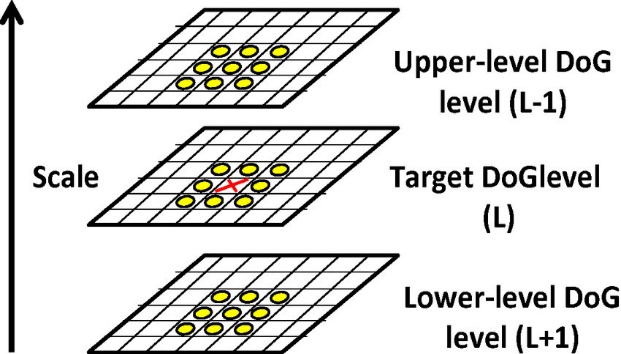
the same scale, 9 corresponding

neighbours in the scale above and 9 neighbours in the scale below). The searching for extrema eliminates the

first and the last image in every one of the octaves due to the fact that they

don’t have a scale above and a scale

below image.



Localizing the key-points- the

detected local extrema are efficient nominees for key-points. On the other hand, they should be precisely localized

via fitting three dimensional quadratic function to the scale-space local sampling point.

**Results:**

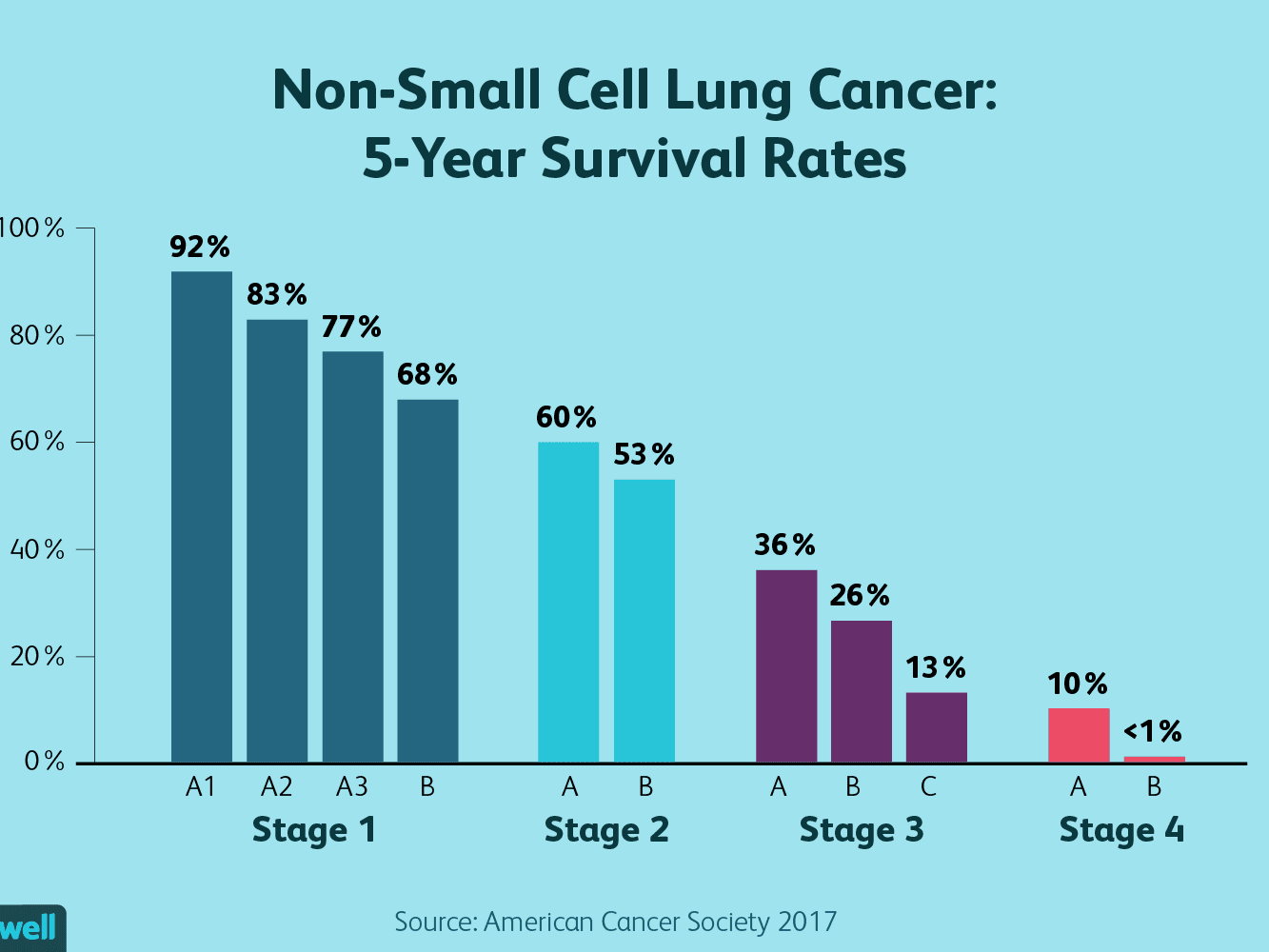
The results were obtained from

testing 8 images taken from a lung cancer dataset.

When applying the original SIFT (224) key points have been found in (6.6) seconds, and all cancer areas were detected and matching.

While implementing the proposed method has resulted in finding (158) key points in (5.619) seconds.

Which means that the proposed method has resulted in a smaller number of key points in less time, while obtaining the accuracy,



This above graph deal with a survival of lung cancer resulting on different stages on SIFT implementation.

**CONCLUSION:**

The main problem of the SIFT algorithm is that it required time to provide the detection, description and matching of the ROI to avoid this problem, fast approximation matching

algorithm is used.

In future work, SURF(Speeded-Up Robust Features algorithm) can be used with fast approximate nearest neighbour in order to reduce the number of features and time .

**REFERENCE LINK:**

<https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>

<https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37>

<https://www.cse.iitb.ac.in/~ajitvr/CS763/SIFT.pdf>

<https://docs.opencv.org/4.5.1/da/df5/tutorial_py_sift_intro.html>

<https://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/>

<https://ieeexplore.ieee.org/abstract/document/6754850>

<https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

<https://www.semanticscholar.org/paper/Scale-Invariant-Feature-Transform-Lindeberg/eab7aa438df70ecefc43c0a6abad1a6592a3b2a1>

**GITHUBLINK:**