**TECHNIQUES OF**

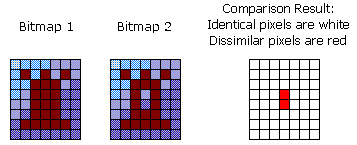
**IMAGE COMPARISON**

1. **PIXEL TOLERANCE**
2. **SSIM**
3. **SIFT**
4. **PIXEL TOLERANCE:**

**Pixel tolerance** is a technique used in computer vision to assess the similarity between two images by analyzing their pixel values. The primary goal of this method is to determine whether two images can be considered identical, despite the presence of minor differences that may arise due to various factors such as noise, lighting variations, or compression artifacts.

**How Pixel Tolerance Works?**

1. **Identifying Dissimilar Pixels**:
   * The first step in the pixel tolerance technique is to compare the pixel values of the two images on a pixel-by-pixel basis.
   * For each pixel in the first image, the corresponding pixel in the second image is examined. If the difference in pixel intensity values exceeds a predetermined or tolerance, that pixel is classified as dissimilar.
2. **Counting Dissimilar Pixels**:
   * After comparing all pixels, the total number of dissimilar pixels is counted.
   * This count provides a quantitative measure of how similar or different the two images are.
3. **Applying Pixel Tolerance**:
   * A predefined **pixel tolerance** value is set, which represents the maximum acceptable number of dissimilar pixels between the two images.
4. **Making a Decision**:
   * If the number of dissimilar pixels is less than or equal to the pixel tolerance, the two images are considered **identical**. This means that the differences are negligible and do not significantly affect the overall perception of similarity.
   * Conversely, if the number of dissimilar pixels exceeds the pixel tolerance, the images are deemed **different**, indicating that they are not sufficiently similar for the purposes of the application.



1. **SSIM: Structural Similarity Index**

In image processing, an image similarity index, also known as a similarity measure or similarity metric, is a numerical value that quantifies the degree of similarity or dissimilarity between two or more images. It's a fundamental concept used in various image analysis tasks, such as image retrieval, image matching, and image classification. Image similarity indices are crucial for comparing and finding similarities between images in a wide range of applications. SSIM assesses the structural similarity between two images by considering luminance, contrast, and structure. It's often used to evaluate the perceived quality of an image compared to a reference image.

**How SSIM Works?**

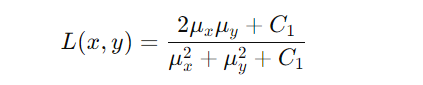
1. **Image Structure:**

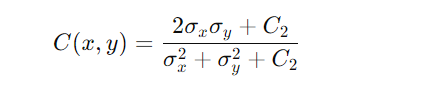
SSIM is based on the idea that the structure of an image (the arrangement of pixels) carries significant information about its visual content. Therefore, it compares local patterns of pixel intensities in the images.

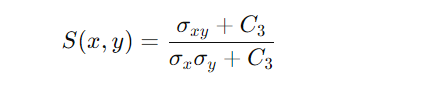
It uses a sliding window approach, dividing the images into smaller patches (usually 8x8 or 11x11 pixels) and computing SSIM for each patch.

1. **Calculating Luminance, Contrast, and Structure:**

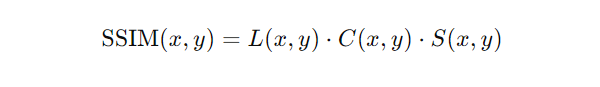
SSIM is computed using three main components:

1. **Luminance**: Measures the brightness of the images. 
2. **Contrast**: Measures the contrast of variance of the images.

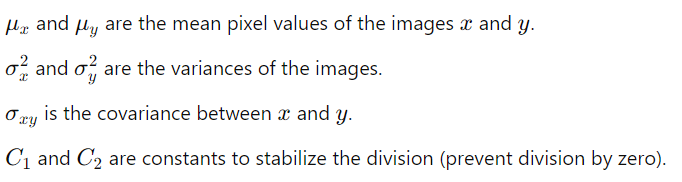


1. **Structure**: Measure the correlation between pixels in the images. 

Overall, it is calculated as:



Where;



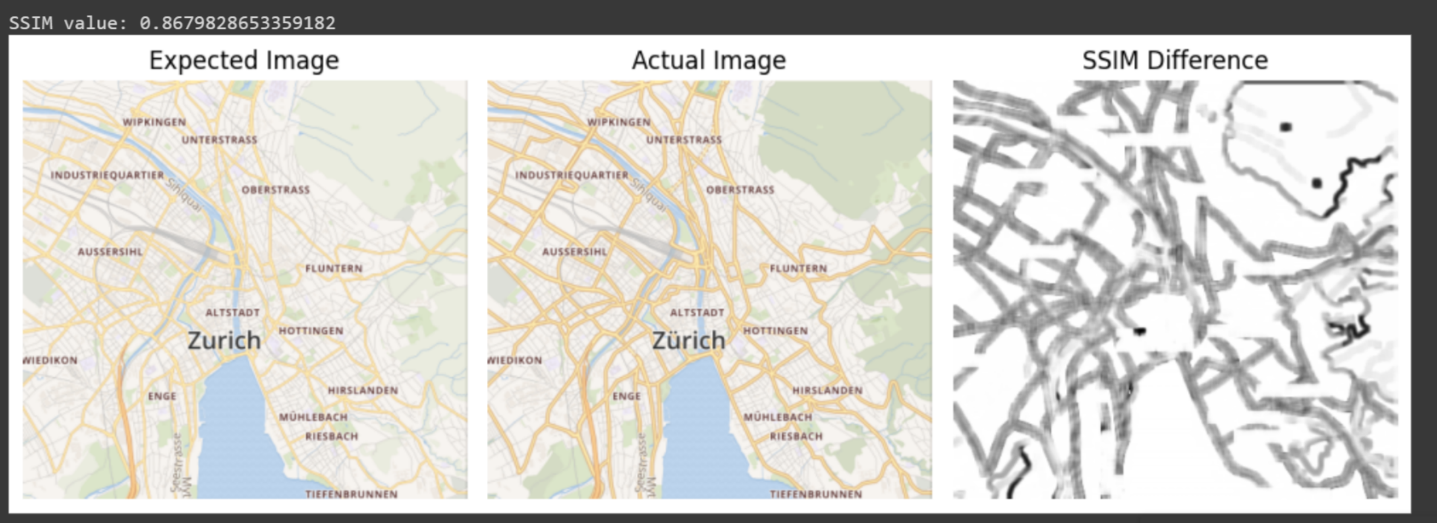
**Overall Similarity Score**:

The SSIM value ranges from -1 to 1, where:

* + **1** indicates perfect similarity (the images are identical).
  + **0** indicates no similarity.
  + Negative values indicate that the images are not similar in terms of structural content.

**Benefits of SSIM**

* **Perceptual Relevance**: SSIM is designed to align with human visual perception, providing a more meaningful assessment of image quality than traditional metrics.
* **Sensitivity to Structural Changes**: SSIM focuses on structural information, making it effective in detecting perceptually important changes in images, such as edges and textures.
* **Robustness to Noise**: SSIM can handle noise and minor variations better than pixel-wise metrics, making it useful in real-world scenarios.



**3.SIFT: Scale-Invariant Feature Transform**

SIFT (Scale-Invariant Feature Transform) is a computer vision algorithm designed to detect and describe local features in images. Developed by David Lowe, it has become one of the most widely used algorithms for feature detection, object recognition, and image matching due to its robustness in handling scaling, rotation, and minor changes in illumination or viewpoint.

**How The SIFT Algorithm Works?**

The SIFT algorithm is designed to detect distinctive features, called key points, in images. These key points are unique areas that can be used to recognize objects in different images, even if the image is scaled, rotated, or slightly changed. Following are the main steps in the SIFT algorithm:

* Scale-space Extrema Detection
* Key point Localization
* Orientation Assignment
* Key point Descriptor

1. **Scale-space Extrema Detection:**

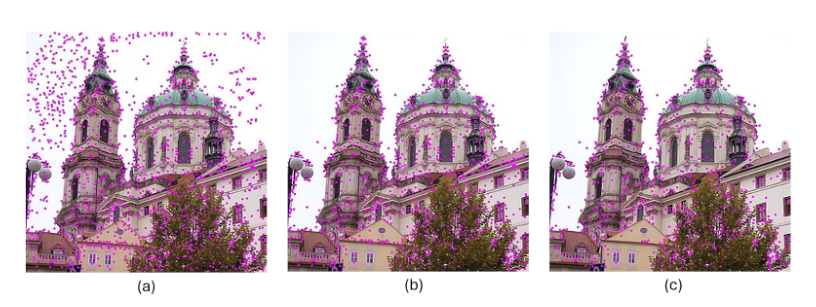
In this step keypoints (features) across different scales of the image are detected. The scale space is the process of creating a set of progressively blurred images at multiple resolutions to detect keypoints that are scale-invariant (they remain the same even if the image size changes). It helps to detect features that are stable and can be recognized even when the image is scaled or resized.  This is done in two steps;

* First, algorithm progressively applies a gaussian blur to the image to blur it at different scales (levels), which smooths it by different amounts. This means we see the image from clear to blurry.
* After blurring images, the algorithm subtracts one blurred image from another, producing the Difference of Gaussians (DoG) images to identify keypoints. These highlight regions where pixel intensity changes significantly. These changes are potential keypoints. DoG is computed by subtracting two gaussian-blurred images at different scales
* Lastly, the algorithm identifies keypoints by finding maxima and minima also known as local extrema (either very bright or very dark spots) over scale and space in the DoG images.

1. **Key Point Localization:**

After building the scale-space and finding potential keypoints (local maxima or minima in the DoG images), the locations of detected keypoints need to be refined to make sure they are accurate. To get a more precise location for each keypoint, a mathematical method called a Taylor series expansion is used.

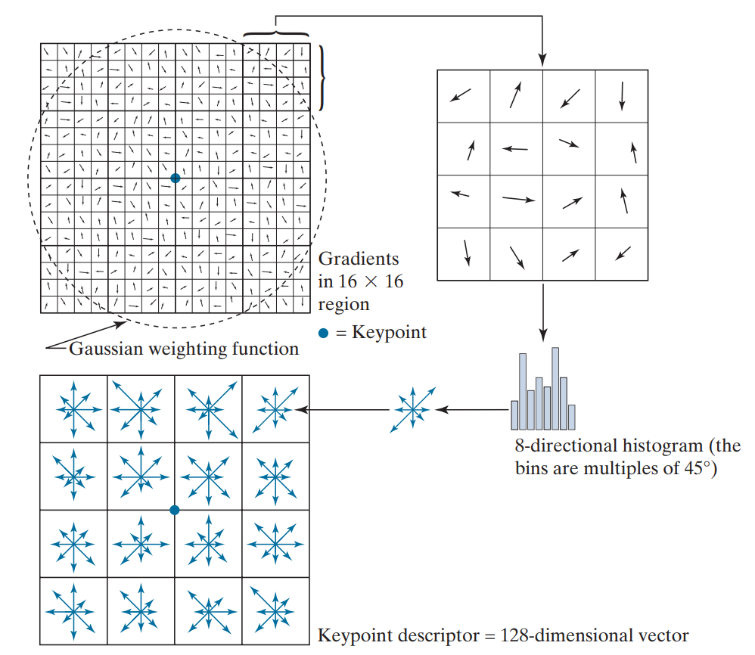
After filtering out low-contrast and edge-response keypoints, the remaining keypoints are considered as well-localized and stable keypoints, making them ideal for use in further processing.



1. **Orientation Assignment:**

The identified keypoints are considered stable (they won’t change much if the image is modified slightly). To make these keypoints invariant to rotation, a direction or orientation is given. This helps the algorithm understand that if the image is rotated, it can still find the keypoint in the same spot because it knows which way it’s pointing. This is important because, without it, the keypoint wouldn’t be able to handle rotated images well.

Each keypoint is given a direction to make the algorithm resistant to image rotation. A small region around the keypoint is analyzed based on its scale, and the magnitude and gradients in the image are calculated. An orientation histogram with 36 bins is created, covering all possible directions (360 degrees). The highest peak in the histogram represents the main direction, but any other peak that is 80% or more of the highest peak is also used to assign an additional direction. This means that keypoints at the same location and scale can have different directions, improving the reliability of matching across images.

1. **Key Points Descriptor:**

A keypoint descriptor is a way to describe keypoints in an image. To create it, we look at a 16x16 area around each keypoint and divide this area into 16 smaller blocks, each 4x4 in size. For each small block, we create an 8-bin histogram to capture the directions of features. This results in a total of 128 values that make up the keypoint descriptor. The descriptor is scale-invariant, rotation-invariant, and robust to changes in lighting and viewpoint.

**Summary of SIFT Steps:**

In summary, the SIFT algorithm processes an image through these four main steps to extract distinctive keypoints that can be used for various computer vision tasks, such as object recognition, image stitching, and 3D reconstruction. By leveraging scale-space representation, robust keypoint localization, orientation assignment, and detailed keypoint descriptors, SIFT effectively identifies features that remain consistent across different images, enabling powerful matching and recognition capabilities.

**Applications of SIFT:**

* Object Recognition: SIFT is commonly used for recognizing objects in images, even when they are viewed from different angles or distances.
* Image Stitching: It is widely applied in panorama stitching, where multiple images are combined to create a single wide-angle view.
* 3D Reconstruction: SIFT features can be used to identify common points in multiple images, aiding in the creation of 3D models.
* Robot Vision: In robotics, SIFT helps in navigating environments by recognizing landmarks.

Ressource : <https://blog.roboflow.com/sift/>