**EEE 508 | Project 01 | Spring 2017**

**Stereo Correspondence Algorithms using SIFT and SURF**

**Bijan Fakhri, Anirudh Som**

1. **Introduction**

Image matching and finding point correspondences between two images are two of the fundamental topics for many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, Image registration, camera calibration and motion tracking. The history of research on visual recognition contains work on a diverse set of other image properties that can be used as feature measurements.

Feature: In computer vision and image processing, 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.

Feature Detection: In computer vision and image processing the concept of feature detection refers to methods that aim at 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.

Feature Description: Given a feature point at location x, scale σ, and orientation θ, we describe the image structure in a neighborhood of x, aligned with θ, and proportional to σ. To facilitate matching, the descriptor should be distinctive and insensitive to local image deformations.

The search for discrete image point correspondences can be divided into three main steps. First, ‘interest points’ are selected at distinctive locations in the image, such as corners, blobs, and T-junctions. The most valuable property of an interest point detector is its repeatability. The repeatability expresses the reliability of a detector for finding the same physical interest points under different viewing conditions. Next, the neighborhood of every interest point is represented by a feature vector. This descriptor has to be distinctive and at the same time robust to noise, detection displacements and geometric and photometric deformations. Finally, the descriptor vectors are matched between different images. The matching is based on t distance between the vectors, e.g. the Mahalanobis or Euclidean distance. The dimension of the descriptor has a direct impact on the time this takes, and less dimensions are desirable for fast interest point matching. However, lower dimensional feature vectors are in general less distinctive than their high-dimensional counterparts.

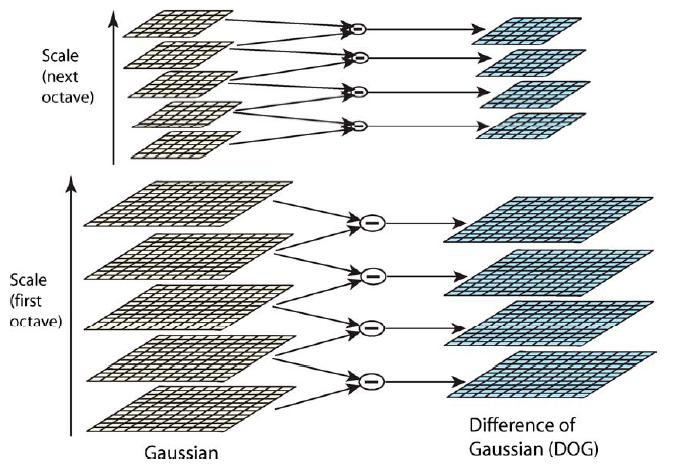
1. **Feature Detector and Descriptors**
2. **Scale Invariant Feature Transform (SIFT)**

SIFT is a very efficient feature detection method based on image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from typical images with these efficient algorithms. The quantity of features is particularly important for object recognition, where the ability to detect small objects in cluttered backgrounds requires that at least 3 features be correctly matched from each object for reliable identification. For image matching and recognition, SIFT features are first extracted from a set of reference images and stored in a database. A new image is matched by individually comparing each feature from the new image to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors.

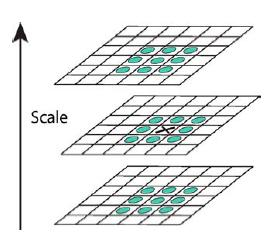
Following are the major stages of computation used to generate the set of image features:

1. **Scale-space Extrema Detection**

The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.



For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.



Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3×3 regions at the current and adjacent scales (marked with circles).

1. **Keypoint Localization**

At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability. Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

1. **Orientation Assignment**

One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

1. **Keypoint Descriptor**

The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. In this way a keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location. These samples are weighted by a Gaussian window and then accumulated into orientation histograms summarizing the contents over 4x4 sub-regions, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region.

The keypoint descriptors are highly distinctive, which allows a single feature to find its correct match with good probability in a large database of features. A different approach is used in which the local descriptor allows relative feature positions to shift significantly with only small changes in the descriptor which not only allows the descriptors to be reliably matched across a considerable range of affine distortion, but it also makes the features more robust against changes in 3D viewpoint for non-planar surfaces.

An important aspect of this approach is that it generates large numbers of features that densely cover the image over the full range of scales and locations. Atypical image of size 500×500 pixels will give rise to about 2000 stable features (although this number depends on both image content and choices for various parameters).

The SIFT keypoints are particularly useful due to their distinctiveness, which enables the correct match for a keypoint to be selected from a large database of other keypoints. This distinctiveness is achieved by assembling a high-dimensional vector representing the image gradients within a local region of the image. The keypoints have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. Large numbers of keypoints can be extracted from typical images, which leads to robustness in extracting small objects among clutter. The fact that keypoints are detected over a complete range of scales means that small local features are available for matching small and highly occluded objects, while large keypoints perform well for images subject to noise and blur. Their computation is efficient, so that several thousand keypoints can be extracted from a typical image with near real-time performance on standard PC hardware. On the other hand, affine invariance is a valuable property for matching planar surfaces under very large view changes, and further research should be performed on the best ways to combine this with non-planar 3D viewpoint invariance in an efficient and stable manner.

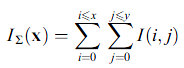
1. **Speeded-Up Robust Features (SURF)**

SURF is a novel scale and rotation invariant detector and descriptor. The SURF detector relies on integral images for image convolutions; by building on the strengths of the leading existing detectors and descriptors specifically, using a Hessian matrix-based measure for the detector, and a distribution-based descriptor; and by simplifying these methods optimally. Blob-like structures are detected at locations where the determinant is maximum. In image I, x = (x, y) is the given point, the Hessian matrix H(x, σ) in x at scale σ, it can be define as



This method pushes the approximation for the Hessian matrix even further with box filters. These approximate second order Gaussian derivatives and can be evaluated at a very low computational cost using integral images. The calculation time therefore is independent of the filter size.

Concept of integral images - The entry of an integral image *I*∑(x) at a location x = (*x,y*)T represents the sum of all pixels in the input image *I* within a rectangular region formed by the origin and x.



Once the integral image has been computed, it takes three additions to calculate the sum of the intensities over any upright, rectangular area. Hence, the calculation time is independent of its size. This is important in our approach, as we use big filter sizes. They allow for fast computation of box type convolution filters.

SURF descriptor describes the distribution of the intensity content within the interest point neighborhood, similar to the gradient information extracted by SIFT and its variants. It builds on the distribution of first order Haar-wavelet responses in x and y direction rather than the gradient, exploit integral images for speed, and use only 64D. This reduces the time for feature computation and matching, and simultaneously increases the robustness. Furthermore, it presents a new indexing step based on the sign of the Laplacian, which increases not only the robustness of the descriptor, but also the matching speed.

Following are the major stages to generate the set of image features: The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. Then, in second step it constructs a square region aligned to the selected orientation and extracts the SURF descriptor from it. Finally, features are matched between two images.

These three steps are listed and explained below -

1. **Orientation Assignment**

In order to be invariant to image rotation, a reproducible orientation for the interest points is dentified. For that purpose, the Haar wavelet responses in x and y direction within a circular neighborhood of radius 6s around the interest point are calculated, with *s* the scale at which the interest point was detected. The sampling step is scale dependent and chosen to be *s*. In keeping with the rest, also the size of the wavelets is scale dependent and set to a side length of 4s. Therefore, integral images can be used for fast filtering. Only six operations are needed to compute the response in x or y direction at any scale.

1. **Descriptor Based on Sum of Haar Wavelet Responses**

For the extraction of the descriptor, the first step consists of constructing a square region centered around the interest point and oriented along the orientation selected in previous section. For each sub-region, Haar wavelet responses should be computed at 5 x 5 regularly spaced sample points. Once the wavelet responses are calculated and weighted with a Gaussian (σ =2s) centered at the interest point, the responses are represented as points in a space with the horizontal response strength along the abscissa and the vertical response strength along the ordinate. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of size. The horizontal and vertical responses within the window are summed. The two summed responses then yield a local orientation vector. The longest such vector over all windows defines the orientation of the interest point. Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

1. **Fast Indexing for Matching**

For fast indexing during the matching stage, the sign of the Laplacian (i.e. the trace of the Hessian matrix) for the underlying interest point is included. The interest points are found at blob-type structures. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature requires no extra computational cost as it was already computed during the detection phase. In the matching stage, only features are compared if they have the same type of contrast. Hence, this minimal information allows for faster matching, without reducing the descriptor’s performance.

Experiments for camera calibration and object recognition highlighted SURF‟s potential in a wide range of computer vision applications. In the former, the accuracy of the interest points and the distinctiveness of the descriptor showed to be major factors for obtaining a more accurate 3D reconstruction, or even getting any 3D reconstruction at all in difficult cases. In the latter, the descriptor generalizes well enough to outperform its competitors in a simple object classification task as well.

**Comparison**

SIFT provides a robust enhancement, however, it is relatively slow. SURF is a variant of SIFT that shares the same robustness and distinctiveness but with a much faster computing speed.

SURF algorithm is employed in slightly different way for detecting image features than SIFT which builds an image pyramids by filtering each layer with Gaussians of increasing sigma values and taking the difference. On the other hand, SURF creates a “stack” without 2:1 down sampling for higher levels in the pyramid which results in having images of same resolution. Due to the use of integral images, SURF filters the stack using a box filter approximation of second order Gaussian partial derivatives. This is because the integral images allow the computation of rectangular box filters in a near constant time.

SURF is, up to some point, similar in concept as SIFT, in that they both focus on the spatial distribution of gradient information. Nevertheless, SURF outperforms SIFT in practically all cases due to the integration the gradient information within a sub-patch, whereas SIFT depends on the orientations of the individual gradients. This makes SURF less sensitive to noise.

1. **Results**
2. **Conclusion**
3. **References**