Scale-invariant feature transform

Scale-invariant feature transform (or **SIFT**) is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by David Lowe in 1999.[1]

Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, and match moving.

The algorithm is patented in the US; the owner is the University of British Columbia.[2]

Overview

For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image be detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges.

Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

Lowe's patented method [3] can robustly identify objects even among clutter and under partial occlusion, because his SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes.[4] This section summarizes Lowe's object recognition method and mentions a few competing techniques available for object recognition under clutter and partial occlusion.

David Lowe's method

SIFT keypoints of objects are first extracted from a set of reference images[4] and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.[5]

Problem	Technique	Advantage
key localization / scale / rotation	DoG / scale-space pyramid / orientation assignment	accuracy, stability, scale & rotational invariance
geometric distortion	blurring / resampling of local image orientation planes	affine invariance
indexing and matching	nearest neighbor / Best Bin First search	Efficiency / speed
Cluster identification	Hough Transform voting	reliable pose models
Model verification / outlier detection	Linear least squares	better error tolerance with fewer matches
Hypothesis acceptance	Bayesian Probability analysis	reliability

Key stages

Scale-invariant feature detection

Lowe's method for image feature generation transforms an image into a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. These features share similar properties with neurons in inferior temporal cortex that are used for object recognition in primate vision.[6] Key locations are defined as maxima and minima of the result of difference of Gaussians function applied in scale-space to a series of smoothed and resampled images. Low contrast candidate points and edge response points along an edge are discarded. Dominant orientations are assigned to localized keypoints. These steps ensure that the keypoints are more stable for matching and recognition. SIFT descriptors robust to local affine distortion are then obtained by considering pixels around a radius of the key location, blurring and resampling of local image orientation planes.

Feature matching and indexing

Indexing consists of storing SIFT keys and identifying matching keys from the new image. Lowe used a modification of the k-d tree algorithm called the **Best-bin-first search** method [7] that can identify the nearest neighbors with high probability using only a limited amount of computation. The BBF algorithm uses a modified search ordering for the k-d tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This search order requires the use of a heap-based priority queue for efficient determination of the search order. The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbors are defined as the keypoints with minimum Euclidean distance from the given descriptor vector. The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest.

Lowe[5] rejected all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches. To further improve the efficiency of the best-bin-first algorithm search was cut off after checking the first 200 nearest neighbor candidates. For a database of 100,000 keypoints, this provides a speedup over exact nearest neighbor search by about 2 orders of magnitude, yet results in less than a 5% loss in the number of correct matches.

Cluster identification by Hough transform voting

Hough Transform is used to cluster reliable model hypotheses to search for keys that agree upon a particular model pose. Hough transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with the feature. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. An entry in a hash table is created predicting the model location, orientation, and scale from the match hypothesis.The hash table is searched to identify all clusters of at least 3 entries in a bin, and the bins are sorted into decreasing order of size.

Each of the SIFT keypoints specifies 2D location, scale, and orientation, and each matched keypoint in the database has a record of its parameters relative to the training image in which it was found. The similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object and also does not account for any non-rigid deformations. Therefore, Lowe[5] used broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times the maximum projected training image dimension (using the predicted scale) for location. The SIFT key samples generated at the larger scale are given twice the weight of those at the smaller scale. This means that the larger scale is in effect able to filter the most likely neighbours for checking at the smaller scale. This also improves recognition performance by giving more weight to the least-noisy scale. To avoid the problem of boundary effects in bin assignment, each keypoint match votes for the 2 closest bins in each dimension, giving a total of 16 entries for each hypothesis and further broadening the pose range.

Model verification by linear least squares

Each identified cluster is then subject to a verification procedure in which a linear least squares solution is performed for the parameters of the affine transformation relating the model to the image. The affine transformation of a model point $[x\ y]^T$ to an image point $[u\ v]^T$ can be written as below

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m1 & m2 \\ m3 & m4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}$$

where the model translation is $[tx\ ty]^T$ and the affine rotation, scale, and stretch are represented by the parameters m1, m2, m3 and m4. To solve for the transformation parameters the equation above can be rewritten to gather the unknowns into a column vector.

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ \ldots\ldots \\ \ldots\ldots \end{bmatrix} \begin{bmatrix} m1 \\ m2 \\ m3 \\ m4 \\ tx \\ ty \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix}$$

This equation shows a single match, but any number of further matches can be added, with each match contributing two more rows to the first and last matrix. At least 3 matches are needed to provide a solution. We can write this linear system as

$$A\hat{\mathbf{x}} \approx \mathbf{b},$$

where A is a known m-by-n matrix (usually with $m > n$), $\mathbf{x}$ is an unknown n-dimensional parameter vector, and $\mathbf{b}$ is a known m-dimensional measurement vector.

Therefore the minimizing vector $\hat{\mathbf{x}}$ is a solution of the **normal equation**

$$A^T A\hat{\mathbf{x}} = A^T\mathbf{b}.$$

The solution of the system of linear equations is given in terms of the matrix $(A^T A)^{-1}A^T$, called the pseudoinverse of A, by

$$\hat{\mathbf{x}} = (A^T A)^{-1}A^T\mathbf{b}.$$

which minimizes the sum of the squares of the distances from the projected model locations to the corresponding image locations.

Outlier detection

Outliers can now be removed by checking for agreement between each image feature and the model, given the parameter solution. Given the linear least squares solution, each match is required to agree within half the error range that was used for the parameters in the Hough transform bins. As outliers are discarded, the linear least squares solution is re-solved with the remaining points, and the process iterated. If fewer than 3 points remain after discarding outliers, then the match is rejected. In addition, a top-down matching phase is used to add any further matches that agree with the projected model position, which may have been missed from the Hough transform bin due to the similarity transform approximation or other errors.

The final decision to accept or reject a model hypothesis is based on a detailed probabilistic model.[8] This method first computes the expected number of false matches to the model pose, given the projected size of the model, the number of features within the region, and the accuracy of the fit. A Bayesian probability analysis then gives the probability that the object is present based on the actual number of matching features found. A model is accepted if the final probability for a correct interpretation is greater than 0.98. Lowe's SIFT based object recognition gives excellent results except under wide illumination variations and under non-rigid transformations.

Competing methods for scale invariant object recognition under clutter / partial occlusion

RIFT [9] is a rotation-invariant generalization of SIFT. The RIFT descriptor is constructed using circular normalized patches divided into concentric rings of equal width and within each ring a gradient orientation histogram is computed. To maintain rotation invariance, the orientation is measured at each point relative to the direction pointing outward from the center.

G-RIF[10] : Generalized Robust Invariant Feature is a general context descriptor which encodes edge orientation, edge density and hue information in a unified form combining perceptual information with spatial encoding. The object recognition scheme uses neighbouring context based voting to estimate object models.

"SURF[11] : Speeded Up Robust Features" is a high-performance scale and rotation-invariant interest point detector / descriptor claimed to approximate or even outperform previously proposed schemes with respect to repeatability, distinctiveness, and robustness. SURF relies on integral images for image convolutions to reduce computation time, builds on the strengths of the leading existing detectors and descriptors (using a fast Hessian matrix-based measure for the detector and a distribution-based descriptor). It describes a distribution of Haar wavelet responses within the interest point neighbourhood. Integral images are used for speed and only 64 dimensions are used reducing the time for feature computation and matching. The indexing step is based on the sign of the Laplacian, which increases the matching speed and the robustness of the descriptor.

PCA-SIFT [12] and GLOH [13] are variants of SIFT. PCA-SIFT descriptor is a vector of image gradients in x and y direction computed within the support region. The gradient region is sampled at 39x39 locations, therefore the vector is of dimension 3042. The dimension is reduced to 36 with PCA. Gradient location-orientation histogram (GLOH) is an extension of the SIFT descriptor designed to increase its robustness and distinctiveness. The SIFT descriptor is computed for a log-polar location grid with three bins in radial direction (the radius set to 6, 11, and 15) and 8 in angular direction, which results in 17 location bins. The central bin is not divided in angular directions. The gradient orientations are quantized in 16 bins resulting in 272 bin histogram. The size of this descriptor is reduced with PCA. The covariance matrix for PCA is estimated on image patches collected from various images. The 128 largest eigenvectors are used for description.

Wagner et al. developed two object recognition algorithms especially designed with the limitations of current mobile phones in mind.[14] In contrast to the classic SIFT approach Wagner et al. use the FAST corner detector for feature detection. The algorithm also distinguishes between the off-line preparation phase where features are created at different scale levels and the on-line phase where features are only created at the current fixed scale level of the phone's camera image. In addition, features are created from a fixed patch size of 15x15 pixels and form a SIFT descriptor with only 36 dimensions. The approach has been further extended by integrating a Scalable Vocabulary

Tree in the recognition pipeline.[15] This allows the efficient recognition of a larger number of objects on mobile phones. The approach is mainly restricted by the amount of available RAM.

Features

The detection and description of local image features can help in object recognition. The SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In addition to these properties, they are highly distinctive, relatively easy to extract, allow for correct object identification with low probability of mismatch and are easy to match against a (large) database of local features. Object description by set of SIFT features is also robust to partial occlusion; as few as 3 SIFT features from an object are enough to compute its location and pose. Recognition can be performed in close-to-real time, at least for small databases and on modern computer hardware.

Algorithm

Scale-space extrema detection

This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images are taken. Keypoints are then taken as maxima/minima of the Difference of Gaussians (DoG) that occur at multiple scales. Specifically, a DoG image $D\left(x,y,\sigma\right)$ is given by

$$D\left(x,y,\sigma\right)=L\left(x,y,k_i\sigma\right)-L\left(x,y,k_j\sigma\right),$$

where $L\left(x,y,k\sigma\right)$ is the convolution of the original image $I\left(x,y\right)$ with the Gaussian blur $G\left(x,y,k\sigma\right)$ at scale $k\sigma$, i.e.,

$$L\left(x,y,k\sigma\right)=G\left(x,y,k\sigma\right)*I\left(x,y\right)$$

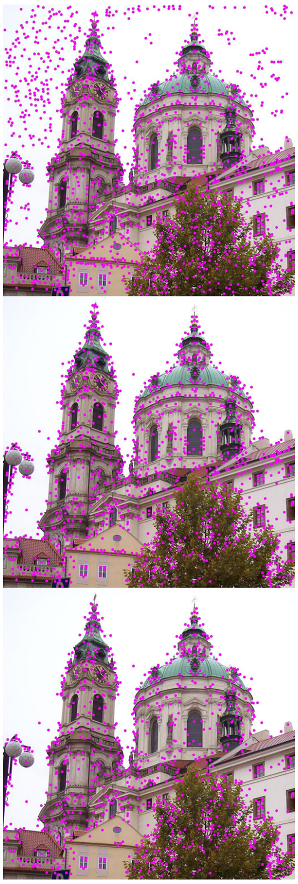
Hence a DoG image between scales $k_i\sigma$ and $k_j\sigma$ is just the difference of the Gaussian-blurred images at scales $k_i\sigma$ and $k_j\sigma$. For scale-space extrema detection in the SIFT algorithm, the image is first convolved with Gaussian-blurs at different scales. The convolved images are grouped by octave (an octave corresponds to doubling the value of σ), and the value of k_i is selected so that we obtain a fixed number of convolved images per octave. Then the Difference-of-Gaussian images are taken from adjacent Gaussian-blurred images per octave.

Once DoG images have been obtained, keypoints are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint.

This keypoint detection step is a variation of one of the blob detection methods developed by Lindeberg by detecting scale-space extrema of the scale normalized Laplacian,[16] that is detecting points that are local extrema with respect to both space and scale, in the discrete case by comparisons with the nearest 26 neighbours in a discretized scale-space volume. The difference of Gaussians operator can be seen as an approximation to the Laplacian, here expressed in a pyramid setting.

Keypoint localization

Scale-space extrema detection produces too many keypoint candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate 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.



After scale space extrema are detected (their location being shown in the uppermost image) the SIFT algorithm discards low contrast keypoints (remaining points are shown in the middle image) and then filters out those located on edges. Resulting set of keypoints is shown on last image.

Interpolation of nearby data for accurate position

First, for each candidate keypoint, interpolation of nearby data is used to accurately determine its position. The initial approach was to just locate each keypoint at the location and scale of the candidate keypoint.[1] The new approach calculates the interpolated location of the extremum, which substantially improves matching and stability.[5] The interpolation is done using the quadratic Taylor expansion of the Difference-of-Gaussian scale-space function, $D\left(x,y,\sigma\right)$ with the candidate keypoint as the origin. This Taylor expansion is given by:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

where D and its derivatives are evaluated at the candidate keypoint and $\mathbf{x} = \left(x,y,\sigma\right)$ is the offset from this point. The location of the extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{x}$ and setting it to zero. If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, then that's an indication that the extremum lies closer to another candidate keypoint. In this case, the candidate keypoint is changed and the interpolation performed instead about that point. Otherwise the offset is added to its candidate keypoint to get the interpolated estimate for the location of the extremum. A similar subpixel determination of the locations of scale-space extrema is performed in the real-time implementation based on hybrid pyramids developed by Lindeberg and his co-workers[17]

Discarding low-contrast keypoints

To discard the keypoints with low contrast, the value of the second-order Taylor expansion $D(\mathbf{x})$ is computed at the offset $\hat{\mathbf{x}}$. If this value is less than 0.03, the candidate keypoint is discarded. Otherwise it is kept, with final location $\mathbf{y} + \hat{\mathbf{x}}$ and scale σ, where $\mathbf{y}$ is the original location of the keypoint at scale σ.

Eliminating edge responses

The DoG function will have strong responses along edges, even if the candidate keypoint is not robust to small amounts of noise. Therefore, in order to increase stability, we need to eliminate the keypoints that have poorly determined locations but have high edge responses.

For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. Finding these principal curvatures amounts to solving for the eigenvalues of the second-order Hessian matrix, **H**:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The eigenvalues of **H** are proportional to the principal curvatures of D. It turns out that the ratio of the two eigenvalues, say α is the larger one, and β the smaller one, with ratio $r = \alpha/\beta$, is sufficient for SIFT's purposes. The trace of **H**, i.e., $D_{xx}+D_{yy}$, gives us the sum of the two eigenvalues, while its determinant, i.e., $D_{xx}D_{yy}-$ product. The ratio $\mathrm{R}=\mathrm{Tr}\left(\mathbf{H}\right)^2/\mathrm{Det}\left(\mathbf{H}\right)$ can be shown to be equal to $\left(r+1\right)^2/r$, which depends only on the ratio of the ei individual values. R is minimum when the eigenvalues are equal to each other. Therefore the higher the absolute difference between the two eigenvalues, which is equivalent to a higher absolute difference between the two principal curvatures of D, the higher the value of R. It follows that, for some threshold eigenvalue ratio r_{th}, if R for a candidate keypoint is larger than $\left(r_{\mathrm{th}}+1\right)^2/r_{\mathrm{th}}$, that keypoint is poorly localized and hence rejected. The new approach uses r_{th} This processing step for suppressing responses at edges is a transfer of a corresponding approach in the Harris operator for corner detection. The difference is that the measure for thresholding is computed from the Hessian matrix instead of a second-moment matrix (see structure tensor).

Orientation assignment

In this step, each keypoint is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

First, the Gaussian-smoothed image $L\left(x,y,\sigma\right)$ at the keypoint's scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L\left(x,y\right)$ at scale σ, the gradient magnitude, $m\left(x,y\right)$, and orientation, $\theta\left(x,y\right)$, are precomputed using pixel differences:

$$m\left(x,y\right)=\sqrt{\left(L\left(x+1,y\right)-L\left(x-1,y\right)\right)^2+\left(L\left(x,y+1\right)-L\left(x,y-1\right)\right)^2}$$

$$\theta\left(x,y\right)=\tan^{-1}\left(\frac{L\left(x,y+1\right)-L\left(x,y-1\right)}{L\left(x+1,y\right)-L\left(x-1,y\right)}\right)$$

The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the keypoint in the Gaussian-blurred image L. An orientation histogram with 36 bins is formed, with each bin covering 10 degrees. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations. Once the histogram is filled, the orientations corresponding to the highest peak and local peaks that are within 80% of the highest peaks are assigned to the keypoint. In the case of multiple orientations being assigned, an additional keypoint is created having the same location and scale as the original keypoint for each additional orientation.

Keypoint descriptor

Previous steps found keypoint locations at particular scales and assigned orientations to them. This ensured invariance to image location, scale and rotation. Now we want to compute a descriptor vector for each keypoint such that the descriptor is highly distinctive and partially invariant to the remaining variations such as illumination, 3D viewpoint, etc. This step is performed on the image closest in scale to the keypoint's scale.

First a set of orientation histograms are created on 4x4 pixel neighborhoods with 8 bins each. These histograms are computed from magnitude and orientation values of samples in a 16 x 16 region around the keypoint such that each histogram contains samples from a 4 x 4 subregion of the original neighborhood region. The magnitudes are further weighted by a Gaussian function with σ equal to one half the width of the descriptor window. The descriptor then becomes a vector of all the values of these histograms. Since there are 4 x 4 = 16 histograms each with 8 bins the vector has 128 elements. This vector is then normalized to unit length in order to enhance invariance to affine changes in illumination. To reduce the effects of non-linear illumination a threshold of 0.2 is applied and the vector is again normalized.

Although the dimension of the descriptor, i.e. 128, seems high, descriptors with lower dimension than this don't perform as well across the range of matching tasks[5] and the computational cost remains low due to the approximate BBF (see below) method used for finding the nearest-neighbor. Longer descriptors continue to do better but not by much and there is an additional danger of increased sensitivity to distortion and occlusion. It is also shown that feature matching accuracy is above 50% for viewpoint changes of up to 50 degrees. Therefore SIFT descriptors are invariant to minor affine changes. To test the distinctiveness of the SIFT descriptors, matching accuracy is also measured against varying number of keypoints in the testing database, and it is shown that matching accuracy decreases only very slightly for very large database sizes, thus indicating that SIFT features are highly distinctive.

Comparison of SIFT features with other local features

There has been an extensive study done on the performance evaluation of different local descriptors, including SIFT, using a range of detectors.[18] The main results are summarized below:

- SIFT and SIFT-like GLOH features exhibit the highest matching accuracies (recall rates) for an affine transformation of 50 degrees. After this transformation limit, results start to become unreliable.
- Distinctiveness of descriptors is measured by summing the eigenvalues of the descriptors, obtained by the Principal components analysis of the descriptors normalized by their variance. This corresponds to the amount of variance captured by different descriptors, therefore, to their distinctiveness. PCA-SIFT (Principal Components Analysis applied to SIFT descriptors), GLOH and SIFT features give the highest values.
- SIFT-based descriptors outperform other local descriptors on both textured and structured scenes, with the difference in performance larger on the textured scene.
- For scale changes in the range 2-2.5 and image rotations in the range 30 to 45 degrees, SIFT and SIFT-based descriptors again outperform other local descriptors with both textured and structured scene content.
- Introduction of blur affects all local descriptors, especially those based on edges, like shape context, because edges disappear in the case of a strong blur. But GLOH, PCA-SIFT and SIFT still performed better than the others. This is also true for evaluation in the case of illumination changes.

The evaluations carried out suggests strongly that SIFT-based descriptors, which are region-based, are the most robust and distinctive, and are therefore best suited for feature matching. However, most recent feature descriptors such as SURF have not been evaluated in this study.

SURF has later been shown to have similar performance to SIFT, while at the same time being much faster.[19]

Recently, a slight variation of the descriptor employing an irregular histogram grid has been proposed that significantly improves its performance.[20] Instead of using a 4x4 grid of histogram bins, all bins extend to the center of the feature. This improves the descriptor's robustness to scale changes.

The Rank SIFT technique was shown to improve the performance of the standard SIFT descriptor for affine feature matching.[21] A SIFT-Rank descriptor is generated from a standard SIFT descriptor, by setting each histogram bin to its rank in a sorted array of bins. The Euclidean distance between SIFT-Rank descriptors is invariant to arbitrary monotonic changes in histogram bin values, and is related to Spearman's rank correlation coefficient.

Applications

Object recognition using SIFT features

Given SIFT's ability to find distinctive keypoints that are invariant to location, scale and rotation, and robust to affine transformations (changes in scale, rotation, shear, and position) and changes in illumination, they are usable for object recognition. The steps are given below.

- First, SIFT features are obtained from the input image using the algorithm described above.
- These 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 keypoints 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 neighbor, an approximate algorithm called the best-bin-first algorithm is used.[22] This is a fast method for returning the nearest neighbor with high probability, and can give speedup by factor of 1000 while finding nearest neighbor (of interest) 95% of the time.
- Although the distance ratio test described above discards many of the false matches arising from background clutter, we still have matches that belong to different objects. Therefore to increase robustness to object identification, we want to cluster those features that belong to the same object and reject the matches that are left out in the clustering process. This is done using the Hough transform. This will identify clusters of features that vote for the same object pose. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. Each keypoint votes for the set of object poses that are consistent with the keypoint's location, scale, and orientation. *Bins* that accumulate at least 3 votes are identified as candidate object/pose matches.
- For each candidate cluster, a least-squares solution for the best estimated affine projection parameters relating the training image to the input image is obtained. If the projection of a keypoint through these parameters lies within half the error range that was used for the parameters in the Hough transform bins, the keypoint match is kept. If fewer than 3 points remain after discarding outliers for a bin, then the object match is rejected. The least-squares fitting is repeated until no more rejections take place. This works better for planar surface recognition than 3D object recognition since the affine model is no longer accurate for 3D objects.

SIFT features can essentially be applied to any task that requires identification of matching locations between images. Work has been done on applications such as recognition of particular object categories in 2D images, 3D reconstruction, motion tracking and segmentation, robot localization, image panorama stitching and epipolar calibration. Some of these are discussed in more detail below.

Robot localization and mapping

In this application,[23] a trinocular stereo system is used to determine 3D estimates for keypoint locations. Keypoints are used only when they appear in all 3 images with consistent disparities, resulting in very few outliers. 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.

Panorama stitching

SIFT feature matching can be used in image stitching for fully automated panorama reconstruction from non-panoramic images. The SIFT features extracted from the input images are matched against each other to find *k* nearest-neighbors for each feature. These correspondences are then used to find *m* candidate matching images for each image. Homographies between pairs of images are then computed using RANSAC and a probabilistic model is used for verification. Because there is no restriction on the input images, graph search is applied to find connected components of image matches such that each connected component will correspond to a panorama. Finally for each connected component Bundle adjustment is performed to solve for joint camera parameters, and the panorama is rendered using multi-band blending. Because of the SIFT-inspired object recognition approach to panorama stitching, the resulting system is insensitive to the ordering, orientation, scale and illumination of the images. The input images can contain multiple panoramas and noise images (some of which may not even be part of the composite image), and panoramic sequences are recognized and rendered as output.[24]

3D scene modeling, recognition and tracking

This application uses SIFT features for 3D object recognition and 3D modeling in context of augmented reality, in which synthetic objects with accurate pose are superimposed on real images. 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 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. For online 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.[25]

3D SIFT-like descriptors for human action recognition

Extensions of the SIFT descriptor to 2+1-dimensional spatio-temporal data in context of human action recognition in video sequences have been studied. [26][27][28][29] The computation of local position-dependent histograms in the 2D SIFT algorithm are extended from two to three dimensions to describe SIFT features in a spatio-temporal domain. For application to human action recognition in a video sequence, sampling of the training videos is carried out either at spatio-temporal interest points or at randomly determined locations, times and scales. The spatio-temporal regions around these interest points are then described using the 3D SIFT descriptor. These descriptors are then clustered to form a spatio-temporal Bag of words model. 3D SIFT descriptors extracted from the test videos are then matched against these *words* for human action classification.

The authors report much better results with their 3D SIFT descriptor approach than with other approaches like simple 2D SIFT descriptors and Gradient Magnitude.[30]

Analyzing the Human Brain in 3D Magnetic Resonance Images

The Feature-based Morphometry (FBM) technique[31] uses extrema in a difference of Gaussian scale-space to analyze and classify 3D magnetic resonance images (MRIs) of the human brain. FBM 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. FBM was validated in the analysis of AD using a set of ~200 volumetric MRIs of the human brain, automatically identifying established indicators of AD in the brain and classifying mild AD in new images with a rate of 80%.[31]

References

[1] Lowe, David G. (1999). "Object recognition from local scale-invariant features" (http://doi.ieeecs.org/10.1109/ICCV.1999.790410). *Proceedings of the International Conference on Computer Vision*. **2**. pp. 1150–1157. doi:10.1109/ICCV.1999.790410. .

[2] US 6711293 (http://worldwide.espacenet.com/textdoc?DB=EPODOC&IDX=US6711293)

[3] U.S. Patent 6,711,293 (http://www.google.com/patents?vid=6,711,293), "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image", David Lowe's patent for the SIFT algorithm, March 23, 2004

[4] Lowe, D. G., “Object recognition from local scale-invariant features”, International Conference on Computer Vision, Corfu, Greece, September 1999.

[5] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[6] Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T., “ A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex (http://cbcl.mit.edu/projects/cbcl/publications/ai-publications/2005/AIM-2005-036.pdf)”, Computer Science and Artificial Intelligence Laboratory Technical Report, December 19, 2005 MIT-CSAIL-TR-2005-082.

[7] Beis, J., and Lowe, D.G “Shape indexing using approximate nearest-neighbour search in high-dimensional spaces”, Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997, pp. 1000–1006.

[8] Lowe, D.G., Local feature view clustering for 3D object recognition. IEEE Conference on Computer Vision and Pattern Recognition,Kauai, Hawaii, 2001, pp. 682-688.

[9] Lazebnik, S., Schmid, C., and Ponce, J., " Semi-Local Affine Parts for Object Recognition (http://hal.archives-ouvertes.fr/docs/00/54/85/42/PDF/bmvc04.pdf)", Proceedings of the British Machine Vision Conference, 2004.

[10] Sungho Kim, Kuk-Jin Yoon, In So Kweon, "Object Recognition Using a Generalized Robust Invariant Feature and Gestalt’s Law of Proximity and Similarity", Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006

[11] Bay, H., Tuytelaars, T., Gool, L.V., " SURF: Speeded Up Robust Features (http://www.vision.ee.ethz.ch/~surf/eccv06.pdf)", Proceedings of the ninth European Conference on Computer Vision, May 2006.

[12] Ke, Y., and Sukthankar, R., " PCA-SIFT: A More Distinctive Representation for Local Image Descriptors (http://www.cs.cmu.edu/~rahuls/pub/cvpr2004-keypoint-rahuls.pdf)", Computer Vision and Pattern Recognition, 2004.

[13] Mikolajczyk, K., and Schmid, C., "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp 1615--1630, 2005.

[14] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, " Pose tracking from natural features on mobile phones (http://mi.eng.cam.ac.uk/~gr281/docs/WagnerIsmar08NFT.pdf)" Proceedings of the International Symposium on Mixed and Augmented Reality, 2008.

[15] N. Henze, T. Schinke, and S. Boll, "What is That? Object Recognition from Natural Features on a Mobile Phone" Proceedings of the Workshop on Mobile Interaction with the Real World, 2009.

[16] Lindeberg, Tony (1998). "Feature detection with automatic scale selection" (http://www.nada.kth.se/cvap/abstracts/cvap198.html). *International Journal of Computer Vision* **30** (2): 79–116. doi:10.1023/A:1008045108935. .

[17] Lindeberg, Tony and Bretzner, Lars (2003). "Real-time scale selection in hybrid multi-scale representations" (http://www.nada.kth.se/cvap/abstracts/cvap279.html). *Proc. Scale-Space'03, Springer Lecture Notes in Computer Science* **2695**: 148–163. doi:10.1007/3-540-44935-3_11. ISBN 978-3-540-40368-5. .

[18] Mikolajczyk, K.; Schmid, C. (2005). "A performance evaluation of local descriptors" (http://research.microsoft.com/users/manik/projects/trade-off/papers/MikolajczykPAMI05.pdf). *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (10): 1615–1630. doi:10.1109/TPAMI.2005.188. PMID 16237996. .

[19] TU-chemnitz.de (http://www.tu-chemnitz.de/etit/proaut/rsrc/iav07-surf.pdf)

[20] Cui, Y.; Hasler, N.; Thormaehlen, T.; Seidel, H.-P. (July 2009). "Scale Invariant Feature Transform with Irregular Orientation Histogram Binning" (http://www.mpi-inf.mpg.de/~hasler/download/CuiHasThoSei09igSIFT.pdf). *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR 2009)*. Halifax, Canada: Springer. .

[21] Matthew Toews, William M. Wells III (2009). "SIFT-Rank: Ordinal Descriptors for Invariant Feature Correspondence" (http://www.cim.mcgill.ca/~mtoews/papers/cvpr09-matt.final.pdf). *IEEE International Conference on Computer Vision and Pattern Recognition*. pp. 172–177. doi:10.1109/CVPR.2009.5206849. .

[22] Beis, J.; Lowe, David G. (1997). "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces" (http://www.cs.ubc.ca/~lowe/papers/cvpr97.pdf). *Conference on Computer Vision and Pattern Recognition, Puerto Rico: sn*. pp. 1000–1006. doi:10.1109/CVPR.1997.609451. .

[23] Se, S.; Lowe, David G.; Little, J. (2001). "Vision-based mobile robot localization and mapping using scale-invariant features" (http://citeseer.ist.psu.edu/425735.html). *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2**. pp. 2051. doi:10.1109/ROBOT.2001.932909. .

[24] Brown, M.; Lowe, David G. (2003). "Recognising Panoramas" (http://graphics.cs.cmu.edu/courses/15-463/2005_fall/www/Papers/BrownLowe.pdf). *Proceedings of the ninth IEEE International Conference on Computer Vision*. **2**. pp. 1218–1225. doi:10.1109/ICCV.2003.1238630. .

[25] Iryna Gordon and David G. Lowe, " What and where: 3D object recognition with accurate pose (http://www.cs.ubc.ca/labs/lci/papers/docs2006/lowe_gordon.pdf)," in Toward Category-Level Object Recognition, (Springer-Verlag, 2006), pp. 67-82

[26] Laptev, Ivan and Lindeberg, Tony (2004). "Local descriptors for spatio-temporal recognition" (ftp://ftp.nada.kth.se/CVAP/reports/LapLin04-SCVMA.pdf). *ECCV'04 Workshop on Spatial Coherence for Visual Motion Analysis, Springer Lecture Notes in Computer Science, Volume 3667*. pp. 91–103. doi:10.1007/11676959_8. .

[27] Ivan Laptev, Barbara Caputo, Christian Schuldt and Tony Lindeberg (2007). "Local velocity-adapted motion events for spatio-temporal recognition" (http://www.csc.kth.se/cvap/abstracts/LapCapSchLin07-CVIU.html). *Computer Vision and Image Understanding* **108** (3): 207–229. doi:10.1016/j.cviu.2006.11.023. .

[28] Scovanner, Paul; Ali, S; Shah, M (2007). "A 3-dimensional sift descriptor and its application to action recognition". *Proceedings of the 15th International Conference on Multimedia*. pp. 357–360. doi:10.1145/1291233.1291311.

[29] Flitton, G.; Breckon, T. (2010). "Object Recognition using 3D SIFT in Complex CT Volumes" (http://www.cranfield.ac.uk/~toby.breckon/publications/papers/flitton10baggage.pdf). *Proceedings of the British Machine Vision Conference*. pp. 11.1-12. doi:10.5244/C.24.11. .

[30] Niebles, J. C. Wang, H. and Li, Fei-Fei (2006). "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words" (http://vision.cs.princeton.edu/niebles/humanactions.htm). *Proceedings of the British Machine Vision Conference (BMVC)*. Edinburgh. . Retrieved 2008-08-20.

[31] Matthew Toews, William M. Wells III, D. Louis Collins, Tal Arbel (2010). "Feature-based Morphometry: Discovering Group-related Anatomical Patterns" (http://www.cim.mcgill.ca/~mtoews/papers/matt_neuroimage09.pdf). *NeuroImage* **49** (3): 2318–2327. doi:10.1016/j.neuroimage.2009.10.032. PMID 19853047. .

External links

- Rob Hess's implementation of SIFT (http://web.engr.oregonstate.edu/~hess/#[[SIFT Feature Detector]]) accessed 20 Mar 2010
- The Invariant Relations of 3D to 2D Projection of Point Sets, Journal of Pattern Recognition Research (http://www.jprr.org/index.php/jprr/article/view/26) (JPRR) (http://www.jprr.org), Vol. 3, No 1, 2008.
- Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004. (http://citeseer.ist.psu.edu/lowe04distinctive.html)
- Mikolajczyk, K., and Schmid, C., "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp 1615--1630, 2005. (http://lear.inrialpes.fr/pubs/2005/MS05/)
- PCA-SIFT: A More Distinctive Representation for Local Image Descriptors (http://www.cs.cmu.edu/~yke/pcasift/)
- Lazebnik, S., Schmid, C., and Ponce, J., Semi-Local Affine Parts for Object Recognition, BMVC, 2004. (http://www-cvr.ai.uiuc.edu/ponce_grp/publication/paper/bmvc04.pdf)
- ASIFT (Affine SIFT) (http://www.ipol.im/pub/algo/my_affine_sift/): large viewpoint matching with SIFT, with source code and online demonstration
- VLFeat (http://www.vlfeat.org/), an open source computer vision library in C (with a MEX interface to MATLAB), including an implementation of SIFT
- LIP-VIREO (http://www.cs.cityu.edu.hk/~wzhao2/lip-vireo.htm), A toolkit for keypoint feature extraction (binaries for Windows, Linux and SunOS), including an implementation of SIFT
- (Parallel) SIFT in C# (https://sites.google.com/site/btabibian/projects/3d-reconstruction/code),SIFT algorithm in C# using Emgu CV and also a modified parallel version of the algorithm.
- Affine + LoG/DoH + SIFT descriptor (http://feature-detection.kilu.de)

Article Sources and Contributors

Scale-invariant feature transform *Source*: http://en.wikipedia.org/w/index.php?oldid=478081938 *Contributors*: Adoniscik, Akinoame, Andreas Kaufmann, BD2412, Bbk8T, Beland, BenFrantzDale, Blackshadowshade, Braddodson, CV SUN, Calliopejen1, Charles Matthews, Chia-Kai Liang, Connelly, Crystallina, Dcoetzee, Dicklyon, Dinesh.srikantha, Dpv, EEMIV, Gemini1980, GxVoyager, Halloleo, HooHooHoo, IByte, Iridescent, Jiuguang Wang, Juan.manuel.gonzalez.gonzalez, KYN, Liface, Lixx0607, Lotje, Lukas Mach, Male1979, Michael Hardy, Mukadderat, NeuronExMachina, Nilx, Paranoid, Peni, Petter Strandmark, Piken, Proberts2003, Qutezuce, RJFJR, Rainald62, Ramunasg, Redgecko, Rich Farmbrough, Richie, Rl, Rwalker, Sanchom, Sauerland, Sbstn, SeanAhern, Shadowjams, Sharat sc, Simplicius, Skapur, Supostat, Surturpain, Sweot.ttam, Tedder, Tobias Bergemann, TomasReiff, Tpl, Tybruce, Waldir, Wallace.cx, Yoderj, 127 anonymous edits

Image Sources, Licenses and Contributors

Image:Sift keypoints filtering.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Sift_keypoints_filtering.jpg *License*: Creative Commons Attribution 3.0 *Contributors*: Original uploader was Lukas Mach at en.wikipedia

License