

1 State of the art

Since few years color image processing is a major problem, indeed lot of the colour texture discrimination has been explored in a marginal colour way. The problematic is that today we can do color image recognition on numerical images but we don't have good results on nature images.

The CLEF contest is an answer to that problematic, the aim of the challenge is to put in competition some university laboratories and company laboratories. In this contest each laboratories could compare its colour texture feature against all the other challengers.

In this way

1.1 Key-points

1.1.1 Classical

1.1.2 Dense Grid

1.2 Descriptors

1.2.1 SIFT

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

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

SIFT keypoints of objects are first extracted from a set of reference images 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.

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 $D(x,y,\sigma)$ is given by:

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma) \quad (1)$$

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

- 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.

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. The new approach calculates the interpolated location of the extremum, which substantially improves matching and stability. The interpolation is done using the quadratic Taylor expansion of the Difference-of-Gaussian scale-space function, $D(x,y,k\sigma)$ with the candidate keypoint as the origin. This Taylor expansion is given by:

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} \quad (2)$$

where D and its derivatives are evaluated at the candidate keypoint and $X=(x,y,k\sigma)$ is the offset from this point. The location of the extremum, \hat{X} , is determined by taking the derivative of this function with respect to X and setting it to zero. If the offset \hat{X} is larger than 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.

Discarding low-contrast keypoints

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

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 :

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

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

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (3)$$

$$\theta(x,y) = \text{atan2}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y)) \quad (4)$$

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 is created on 4x4 pixel neighborhoods with 8 bins each. These histograms are computed from magnitude and orientation values

of samples in a 16×16 region around the keypoint such that each histogram contains samples from a 4×4 subregion of the original neighborhood region. The magnitudes are further weighted by a Gaussian function with 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 $4 \times 4 = 16$ 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 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.

1.2.2 SURF

1.2.3 Opponent SIFT

1.2.4 C₂O