SIFT

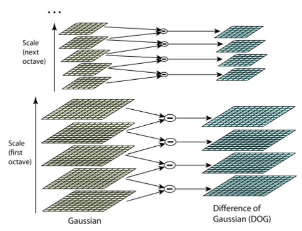
A Preprint

Creation of panoramas, creation of a stereo pair and reconstruction of a three-dimensional model of an object based on its two-dimensional projections in principle, recognition of objects and search by sample from some database, tracking the movement of an object based on several images, reconstruction of affine transformations of images. All of these tasks may be solved when the task of comparing images emerges. We call a detector a method of extracting key points from an image. The detector must ensure invariance of finding the same special points with respect to image transformations. Descriptor itself is a point identificator as point should be identified from the point of clouds. Overlall process of comparting images looks like that:

- Identify key points and their descriptors
- By matching descriptors find key points
- Compare based on the sequence of these key points.

It s also important to note that key points should be invariant to all linear transformations(affine etc) and also invariant to changes in camera direction.The key point in detecting singular points is to construct a pyramid of Gaussians and differences of Gaussians (DoG). A Gaussian (or an image blurred by a Gaussian filter) is an image $L(x, y, z) = G(x, y, \sigma) \cdot I(x, y)$, where L is value of Gaussian in point (x, y), σ - radius of blur, G - gaussian kernel, I - values of original image, * is a conv operation. DoG is image which was achieved by subtracting one Gaussian of the original image pixel by pixel from a Gaussian with a different blur radius.

It is proved that the Gaussian scalable space is linear, invariant with respect to shifts, rotations, scale, does not shift local extrema, and has the property of semigroups. It is important for us that different degrees of blurring of an image by a Gaussian filter can be taken as the original image taken at a certain scale. Overall, invariance to scale is achieved by pyramid of gaussians and the space is split into octaves, and the part of the scaled space occupied by the next octave is twice as large as the part occupied by the previous one. In addition, when moving from one octave to another, the image is resampled, its dimensions are halved. Naturally, each octave covers an infinite number of Gaussians of the image, so only a certain number of them N are built, with a certain step along the blur radius. With the same step, two additional Gaussians are built (in total, N + 2), going beyond the octave. It will be seen below why this is necessary. The scale of the first image of the next octave is equal to the scale of the image from the previous octave with the number N. At the same time sa building PoG the DoG pyramid is build.



The point is counted as special point if the point is local extremum. Local extremum is defined by comparing neigbours of point to itself(8 neighbours). To define wether point is a key point or not next algorithm is used:

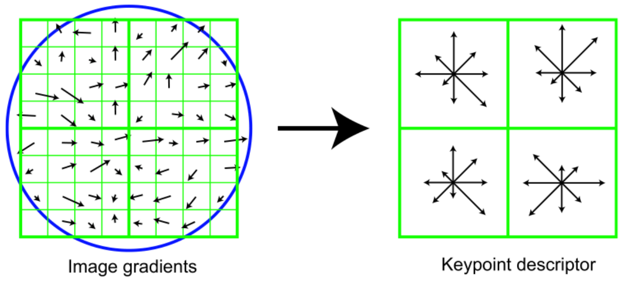
- Coordinates of special point are defined by approximating the DoG function with a second-order Taylor polynomial taken at the point of the calculated extremum
- The derivative is taken and equated to zero
- After that we obtain system of linear equations 3x3 dim
- If one of the components of the vector X is greater than 0.5*grid step in this direction, then this means that the extremum point was actually calculated incorrectly and we need to move to the neighboring point in the direction of the specified components. For the neighboring point, everything is repeated again. If we have thus gone beyond the octave, then this point should be excluded from consideration.
- If a special point lies on the border of some object or is poorly lit, then such a point can be excluded from consideration. This can be checked by Hessian matrix
- Direction is defined by

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

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

- To assign an orientation to a keypoint, we define a circular window centered at the keypoint. The radius of this window is determined by the rule of three sigmas: it is set to $[3 \cdot \sigma]$, where $\sigma = 1.5 \cdot$scale of the keypoint. Within this window, a histogram O of gradient orientations is constructed. The histogram consists of 36 bins covering the full 360° range. Each pixel (x, y) in the window contributes a weighted vote to the histogram based on the gradient magnitude $m(x, y)$ and a Gaussian weighting function $G(x, y, \sigma)$. The vote is added to the bin corresponding to the gradient orientation $\theta(x, y)$.
  The dominant orientation is determined by finding the maximum value in the histogram. To increase accuracy, the peak and its two neighboring bins are interpolated using a parabola, and the maximum of this parabola is taken as the keypoint orientation. If other local maxima in the histogram are at least 80% of the global maximum, they are also assigned as additional orientations for the keypoint.

Now let s come to building descriptors. Basically, descriptor is some information about key point surroundings but actually it can be anything.



The descriptor is computed over a square window centered near the keypoint, subdivided into 4×4 regions. Each region contains several pixels, and each pixel contributes a gradient (represented by a small arrow) with a direction and magnitude. The center of the window is aligned with subpixel accuracy, as close as possible to the precise location of the keypoint. A circular Gaussian weighting kernel is applied to the entire descriptor window, with standard deviation $\sigma = \frac{1}{2} \times$ descriptor window width, to assign weights to gradient contributions.

Each of the 4×4 regions accumulates an 8-bin orientation histogram, resulting in a descriptor of dimensionality $4 \times 4 \times 8 = 128$ (although the schematic shows a simplified $2 \times 2 \times 8 = 32$ version).

Gradient votes are distributed into histograms using:

- The gradient magnitude, weighted by the Gaussian kernel.
- Tri-linear interpolation in 3D: across spatial dimensions (x, y) and orientation (angle).

Each gradient is associated with fractional coordinates (x, y, n), where:

- x and y represent spatial offsets relative to the lower-left corner of the descriptor window, normalized by region width/height.
- n is the fractional bin index of the orientation histogram (based on 360° divided into 8 bins).

Tri-linear interpolation distributes each gradient's contribution across neighboring bins, weighted by $1 - d$, where d is the offset from the center of the target bin along each dimension.

After building the descriptor:

1. It is normalized to unit length.
2. Components above 0.2 are clipped to 0.2.
3. It is renormalized to unit length again.

This ensures robustness to illumination changes and outliers.