Distinctive Image Features fom Scale-Invariant Keypoints

Mohammad-Amin Ahantab

Technische Universität München

Abstract. This work presents the Scale Invariant Feature Transform. This algorithm detects stable and distinctive image features which can be matched with high probabilty against other features of diffrent images. These features are invariant to scale , rotation and partly invariant to illumination and affine transformation.The nearest neighbour matching method provides an approach to match image features from diffrent images.

1 Motivation

Distinctive Image features can be used in many computer vision application such as object recognition, motion tracking , 3d scene recogntion , stereo correspondence and Panaroma stitching.

2 Image features

Before we can create stable image features one have to figure out the requirements these image features should meet. If you want to match an image of an object in a scene in a real life scenario the object has been trasformed in various ways in this scene. Therefore it would be beneficial if the image features are invariant to transformations such as scale, rotation, affine transformation (view point) and illumination. This algorithm provides a true invariance to scale and rotation and it is very robust to illumination, however there is no real invariance to affine transformation but it is still robust up to 60 degrees of affine transformation. Also the image feature should be highly distinctive as we would not like to have wrong matches between features.

3 Scale Invariant Feature Transform (Sift)

The Scale Invariant Feature Transform algorithm searches for interest points or "keypoints" in the image and describes local image features. The algorithm produces a large number of features, a image with a resolution of 500x500 pixel results in roughly 2000 image features (although it depends on the image itself). Not only the quality of image features is important but also the quantity as it would be required if you want to detect a small object in a scene. The Sift

algorithm is structured into 4 major steps which will be explained in detail in this work :

1.Scale-space extrema detection
2.Keypoint localization
3.Orientation assignment
4.Keypoint descriptor

3.1 Scale-space extrema detection

Scale-space In order to create scale-invariant image data we need to contrust a scale space so that we can search for interests point over all scales. Therefore we take the original sized image and smoothen it progessively so that we don't add noise to our further calculations .In the next step we take the original image and reduce the resolution by half and smoothen it again progressively. The scales are called octaves, it is suggested to create 4 octaves and for each octave it is suggested to blurr the image 5 times (fig.1). I is the image which takes the

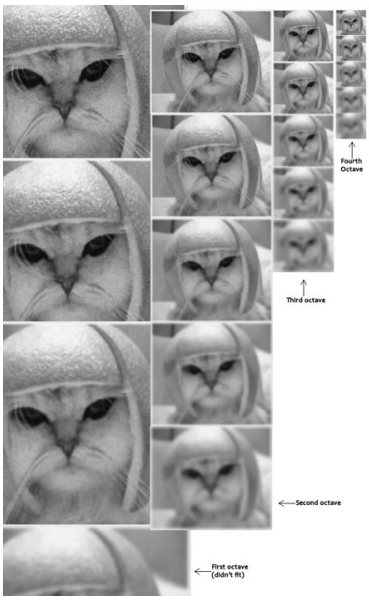

Fig. 1. Scale-Space

The image is smoothened by the Gaussian function :

$$G(x, y, \sigma) = \frac{1}{(2 \cdot pi \cdot \sigma^2)} \cdot e^{-} \frac{(x^2 + y^2)}{2 \cdot \sigma^2} \tag{1}$$

Coordinates x and y. The Gaussian function with the scale σ applied to I results in the smoothened image L (2)

$$L(x, y, \sigma) = G(x, y, \sigma) \cdot I(x, y) \tag{2}$$

Difference the Difference of Gaussian images In this step we calculate the Difference of Gaussian(DoG) Images. Therefore we take two consecutive images for each octave (Fig.2) and subtract them so that we create 4 images for each octave .

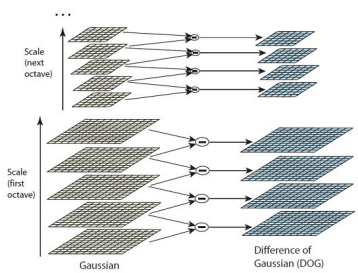

Fig. 2. Difference of Gaussian

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3}$$

The Motivation behind this is based on the approximation of the scale-normalized Laplacian of Gaussian $\sigma^2 \bigtriangledown^2 G$ with the Difference-of Gaussan function $D(x, y, \sigma)$ The importance of the scale-normalized Laplacian of Gaussian is based on two assumptions:
1.Lindeberg (1994) : normalization of the laplacian with σ^2 causes true scale invariance
2.Mikolajczyk (2002) : extrema of $\sigma^2 \bigtriangledown^2 G$ produce the most stable image features
The relation between the DoG function and $\sigma^2 \bigtriangledown^2 G$ can be explained by the heat diffusion equation:

$$\frac{\partial G}{\partial \sigma} = \sigma^2 \bigtriangledown^2 G \tag{4}$$

By applying finite difference approximation (5) we can see that the DoG function already includes σ^2 which is required for scale invariance. The approximation differs by the constant (k-1) , however this does not affect the function significantly. This approximation is much faster and efficient then calculating the second order derivative of the Gaussian function!

$$\sigma \bigtriangledown^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, \sigma) - G(x, y, k\sigma)}{k\sigma - \sigma} \tag{5}$$

$$G(x, y, \sigma) - G(x, y, k\sigma) \approx (k-1)\sigma^2 \bigtriangledown^2 G \tag{6}$$

Local Maximas/Minimas in Dog images As mentioned in the previous section extrema of the DoG images produce the most stable image features. Therefore we take a pixel and compare it to all its neighbour pixel in its current scale, the scale above and below it. if the pixel is smaller or bigger than all of its 26 neighbours it will be selected as an extreme sample point(Fig. 3).

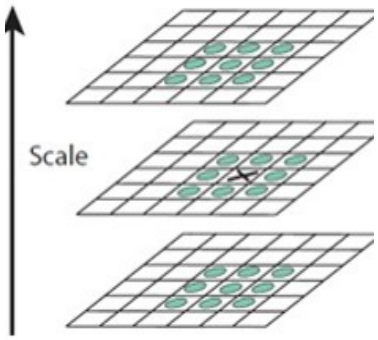

Fig. 3. Maximas/Minimas in DoG

3.2 Keypoint localization

In this step we eliminate unstable interest points such as edge responses and low contrast points. Most importantly we try to locate the keypoints as the previous calculated extreme points are in fact not accurate enough.

Taylor expansion The extrema (samples) which are found in the DoG images by comparing pixels to their neighbours are not accurate as they often times are located "between" the pixels (subpixels). In this approach we calculate the keypoints with the taylor expansion(7). Therefore the origin of the Taylor expansion is set at the sample point $D(x, y, \sigma)$. We then set the derivative of $D(\hat{x})$ to zero and calculate the extremum $\hat{x}$ (8) . If $\hat{x}$ is smaller than 0.5 the offset is added to the location of the local sample point otherwise we proceed wth a diffrent sample point as$D(\hat{x})$ is obviously closer to a diffrent sample point.

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 l D^T}{\partial x^2}x \tag{7}$$

$$\hat{x} = \frac{\partial^2 D^-1}{\partial x^2}x + \frac{\partial D}{\partial x^2}x \tag{8}$$

Eliminating low contrast points Keypoints with a low contrast can be very unstable and therefore have to be eliminated ! That is simply done by eliminating all extrema with the value $D(\hat{x})$ less than 0.03.

Eliminating edge responses While cornes are very stable image features edges on the other side should be eliminated. edges are found by taking into consideration that across an edge the principal curvature is large whereas in perpendicular direction it is small. The principal curvature can be computed with the Hessian matrix of the DoG function because the egenvalues of the Hessian Matrix are propotional to the curvatures! The eigenvalues dont have to be calculated as the ratio of the trace to the power of 2 of H and the Determinant of H is proportional to the ratio of the cuveratures as shown in the calculations below. If the ratio r of the Eigenvalues is bigger than 10 we eliminate the key point.

$$H = \begin{pmatrix} D_{xx} & D_{x2} \\ D_{xy} & D_{y,y} \end{pmatrix} \tag{9}$$

$$Tr(H) = D_{xy} + + D_{xy} = \alpha + \beta \tag{10}$$

$$Det(H)) = D_{xx} - D_{yy}^2 = \alpha\beta \tag{11}$$

$$\frac{Tr(H)^2}{Det(H)} = frac(\alpha + \beta^2\alpha\beta = \frac{r\beta + \beta)^2}{r\beta^2} = frac(r+1)^2 r \tag{12}$$

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{13}$$

3.3 Orientation assignment

In this step an orientation is assigned to the keypoint to ensure rotation invariance. Therefore we compute the magnitude and orientation of the gradients around the key point of the smoothed images L(x,y) (Fig.4). Then a histogram (Fig.5) with 36 bins (each 10 degrees) is created and each bin is quantized with respect to the magnitude of the gradients. The orientation with the highest peak is assigned to the keypoint. If there is a peak with at least 80 % of the highest peak a new keypoint is created with the respective orientation. So it is possible to have multiple keypoints at the same scale and location with diffrent rotation.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{14}$$

$$\theta(x,y) = tan^{-}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \tag{15}$$

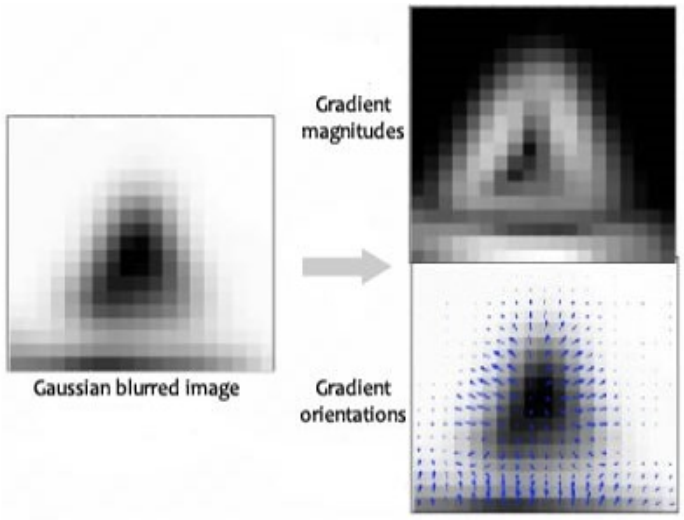

Fig. 4.

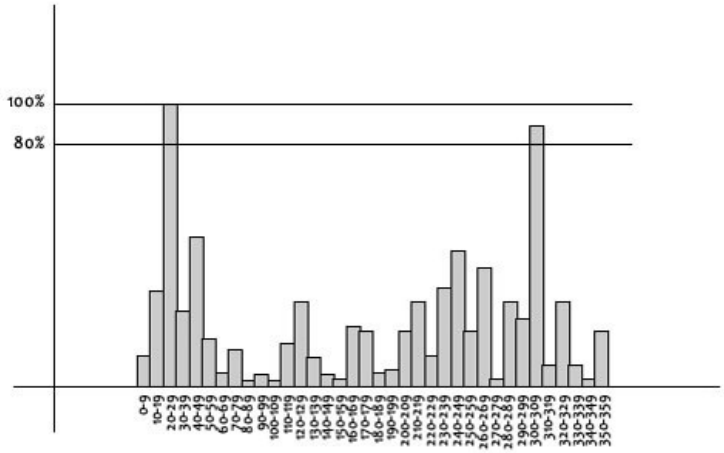

Fig. 5. Histogram

3.4 Keypoint descriptor

Now that we have found interest points which are invariant to scale and rotation we need to describe image features.We use a 16x16 sample array around the keypoint and tranform it into a 4 x 4 descriptor. Therefore the position and orientation of each sample is rotated relatively to the keypoint orientation to ensure rotation invariance. In each field of the 4 x4 descriptor a histogram with 8 bins is created which results in a 128 dimensional vector which constitutes the image feature. The length of each arrow is computed with respect to the magnitutde of the samples. The Gaussian weight function which is used to do so ensures that gradients which are far from the keypoint will not have a greater impact than gradients closer to the keypoint.

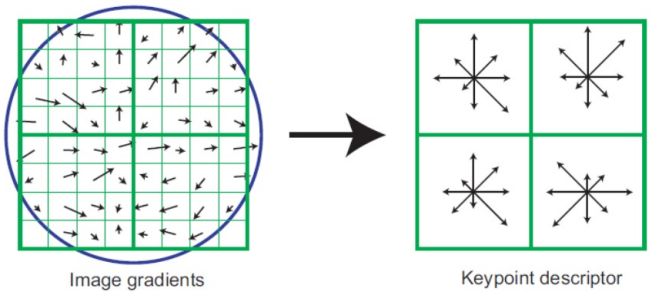

Fig. 6. An example of a 8x8 image gradients window the resulting 2x2 descriptor

3.5 Illumination Invariance

In real life scenarious lightning conditions are most of the time are never the same . Therefore it is important to cope with diffrent illumination scenarios such as dark or bright scenes or changing light sources. Illumination changes have impact on contrast and brightness. If the contrast is changed the pixel values are multiplied by a constant factor therfore it is sufficient to normalize the descriptor vector. If the brightness is changed a constant factor is added to the image values and therefore it does not affect the gradients since they are the result of the diffrence of these values. Non-linear illumination however can change orientations by diffrent amounts. However most likely it has only an effect on large magnitudes of the gradients and therefore we limit the values in the vector to not larger than 0.2 and renormalize the descriptor to unit length.

4 Keypoint matching

If we want to match two images we need to match the image features of those images. Therefore we find the nearest neighbour of an image feature in a database of keypoints. The nearest neighbour is defined as the keypoint with the minimum Euclidean distance. With this approach it can lead to many false matches because the other neighbours can be also close to that keypoint. We can cope with that problem by calculating the ration between the second best neighbour and the best neighbour. If the ratio is bigger than 0.8 the keypoints are not matched.

5 Conclusion

Sift provides image features which are invariant to scale and rotation, partly invariant to illumination and robust to limited affine transformation. The algorithm is very fast as it uses the effcient Gaussian function and the DoG approximation so that it almost can be run in real time. It also ensures high distinctiveness and produces a large number of features.

References

1. Lowe, D. Distinctive image features from scale-invariant keypoints
2. http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/