SUMMARY: DISTINCTIVE IMAGE FEATURES FROM SCALE-INVARIANT KEYPOINTS

Cognitive Robotics
Original: David G. Lowe, 2004
Summary: Coen van Leeuwen, s1460919

Abstract: This article presents a method to extract distinctive keypoints from images that can be used to match images of the same objects, but with different viewpoints. The method is showed to be invariant to scaling and rotation, and very robust for matching over affine changes, change in 3D viewpoint, noise and illumination. The features are highly distinctive for an object, so a single feature can be reliably matched against a large database. Also, a method will be proposed to use these keypoints for object recognition. The proposed object recognition system shows to be capable of robustly identifying objects among clutter and occlusion in near real-time.

1. Introduction

Image matching is one of the most researched aspect in computer vision. It can be used for object recognition, scene recognition, solving 3D structure for multiple images, stereo correspondence and motion tracking. A good way to match objects in images is by finding features in the images that are the same for the objects despite differences in location, rotation, scale, 3D viewpoint, illumination or noise. In order to obtain these features, the method in this paper uses the following steps:

- **Scale Space extrema detection**: By using difference-of-gaussians, the pixel value extrema in both scale and space are found.
- **Keypoint localization**: The most stable extrema are converted to keypoints.
- **Orientation assignment**: The keypoints are transformed so their orientations are normalized.
- **Keypoint descriptor**: The local image gradients are stored to represent the keypoint.

This method is called the Scale Invariant Feature Transform (SIFT) as it finds the features invariant of the scale of the object.

In each image a lot of stable features will be found, and these features can be matched against known features to identify an object in the image. Clusters of 3 or more matching features reliably indicates the presence of an object as the keypoint descriptors are highly distinctive.

In the next chapter some of the related research will be discussed. After that we will look at the method by which the features are extracted.

2. Related research

Image matching using a set of local interest points was first used by Moravec (1981) for stereo image matching. His model showed to be very useful and was improved and extended to be used for motion tracking and 3D structure recovery by Harris (1992) from motion. Since then the Harris corner detector has been widely used for image matching tasks.

Zhang *et al.* (1995) showed that it was possible to match Harris corners on a large image range using a correlation windows around the corners to select matches. After that, Schmid and Mohr (1997) showed that it was possible to use invariant local features to match images against a large database of images. In this way the first image recognition system was created. Schmid and Mohr used a rotationally invariant descriptor of the local image region.

Crowley and Parker (1984) were some of the first people to identify a representation that is stable under scale change. They describe peaks

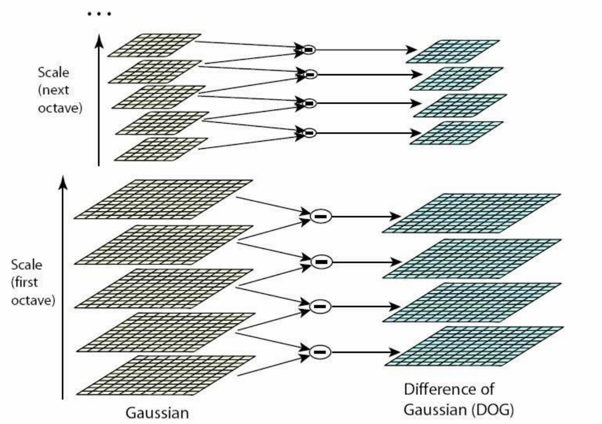

Figure 1: The pyramid like structure of the difference-of-Gaussian function

and ridges in scale space linked in a tree structure. These tree structures could be matched between image with arbitrary scale change.

Since then, much research is done to try to make affine invariant detectors. This means that detectors do not vary under changing 3D viewpoint.

3. Detection of scale-space extrema

The first step in finding the image features is to identify locations in the image that are invariant to scale change. This is achieved by searching to stable features across all possible scales, using a continuous function of scale called the scale space.

We define the scale space *L* by convolving the original image *I* with a set of variable-scaled Gaussians *G*:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$
$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \qquad (1)$$

Where σ denotes the scale of the Gaussian. To efficiently find stable keypoints, two of these Gaussian images are subtracted from each other, with only the factor σ changing. The results is a difference-of-Gaussian function:

$$D(x,y,\sigma) = L(x,y,\sigma) - L(x,y,k\sigma) \qquad (2)$$

Apart from the fact that calculating this is very fast, because the scales are needed anyway for the descriptor, this function closely resembles the scale-normalized Laplacian of Gaussians, which is showed to be required for true scale-invariance. The minima and maxima of these Laplacians are found to give the most stable image features by Mikolajczyk (2002) compared to gradients, Hessian or Harris corner functions. By using a constant factor *k* in equation 2, the difference of Gaussian function incorporates the scale normalization like the Laplacian does.

To construct the difference-of-Gaussians, the original image is incrementally convolved with Gaussians to produce images which are separated by a constant *k* in scale space. The scale space is divided in octaves in which σ is doubled. Each octave in scale space is chosen to be divided into *s* intervals, so $k = 2^{1/s}$. All adjacent images are subtracted from each other to obtain the difference-of-Gaussians. Once a complete octave has been processed, the image resolution is reduced by half. This can be done without any loss of accuracy, while computation is greatly reduced. See figure 1 for a graphical presentation of the difference-of-Gaussian function.

3.1 Local extrema detection

In order to detect local minima and maxima, each pixel is compared to its 26 neighbors in the surrounding scale-space (see figure 2). A pixel is only used if it is larger or smaller than all of the other pixels.

An important issue is to determine the frequency of sampling in the image and scale domains to reliably find the extrema. If we consider an ellipse, we would have to find two maxima near the ends of the ellipse. If the ellipse would slowly be transformed into a circle, at which point should only one location be found? In the following two mini-experiments the optimal values for scale and space frequency are determined for the system.

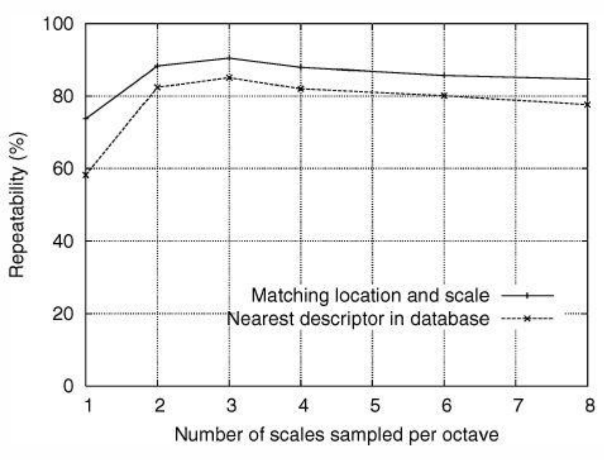

Figure 2: The stability of the keypoints with varying amount of scales per octave

3.2 Frequency of sampling in scale

To compare various settings, a collection of 32 images was transformed by a various set of operations including rotation, scaling, affine stretch, addition of noise an change in brightness and contrast or any combination of these operations. Because the changes are synthetic, it was possible to precisely predict where extrema should be found in the transformed image compared to extrema in the original. This way the stability of the system against any type of transformation could be determined.

Figure 3 shows the results of the experiment in which the amount of scales per octave is varied. For now only the top line is relevant, which shows the percent of keypoints that are detected at a matching location and scale. A matching scale is defined to be within a factor of $\sqrt{2}$ of the correct scale, and a matching location is defined as being within σ pixels from the correct location, where σ is the standard deviation of the smallest Gaussian of the scale it is found in (as defined in equation 2). The lower line shows the percentage of locations that give a correctly matching descriptor, which we will tell more about in chapter 6. We can see from figure 3 that the optimal number of scales per octave is 3.

3.3 Frequency of spatial sampling

Just as the optimal frequency of sampling per octave of scale space was determined, we must determine the frequency of sampling in the image domain relative to the scale of smoothing. Another small experiment was done to determine the amount of prior smoothing before building the scale-space representation for an octave. It's result showed that higher amounts of smoothing will give better results. Because this is a fairly costly computation, we will use the value $\sigma = 1.6$ throughout this paper.

To compensate for the pre-smoothing, the image resolution is doubles before building the first layer of the pyramid. This is done using linear pixel interpolation. This results in much more keypoints created per image. No significant further improvements were found by expansions with even higher factors.

4. Accurate Keypoint localization

Once the difference-of-Gaussian scale space extrema are found, the nearby data for is used to get more stable keypoints. Points with low contrast are for instance rejected because they are sensitive to noise, or localized along an edge. Also, in an attempt to get a more exact location of the keypoint location, a 3D quadratic function is fitted to the local sample points, to obtain a more exact location of the extremum.

Figure 4 shows the effects of keypoint selection in a natural image. 4(a) shows the original image and 4(b) is the image with all the detected extrema of the difference-of-Gaussian function. Then in 4(c) you see the keypoints that remain following the removal of keypoints with low contrast. Also the exact location of the extrema are interpolated, but this is not visible. 4(d) will be explained in the following section.

4.1 Eliminating edge responses

To select the most stable keypoints, it is not sufficient to only reject keypoints with low stability. The difference-of-Gaussian function will have a strong response along edges, even if it is poorly determined. Therefore it is very unstable to noise.

A poorly defined peak in the difference-of-Gaussian function will have a large principal curvature along the edge, but a small one in the perpendicular direction. The two principal

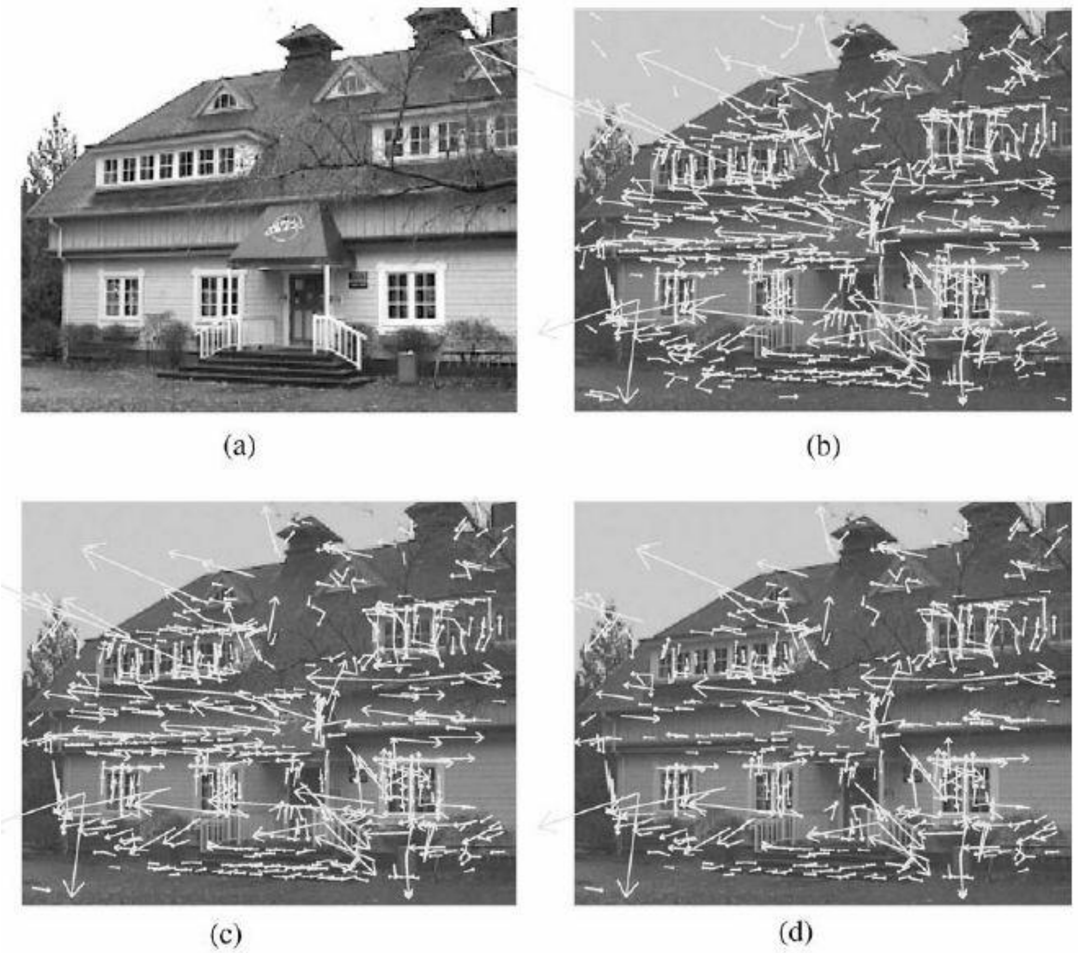

Figure 3: The keypoints as found in the image. (a) The original image. (b) All the extrema in the difference-of-Gaussian. (c) Without thee keypoints with low contrast. (d) Without the keypoints on the edges

curvatures can be calculated with a 2x2 Hessian matrix computed at the scale and location of the keypoint. When the ratio between the two principal curvatures is larger then 10, it is considered to be on an edge in this paper. These keypoints are rejected to obtain more stable keypoints. The results of this operation is presented in figure 4(d).

5 Orientation assignment

The locations that have passed all of the previous test are now transformed so that they become rotationally invariant. The nearest Gaussian smoothed image L is used to compute the gradient magnitude and orientation of the keypoint.

$$m(x,y) = \sqrt{\begin{array}{l}(L(x+1,y)-L(x-1,y))^2 + \\ (L(x,y+1)-L(x,y-1))^2\end{array}}$$
$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \qquad (3)$$

An orientation histogram is formed from the gradient orientations θ of sample points in the region around the keypoint. The orientation histogram has 36 bins covering 360 degree range of orientations. Each sample is weighted according to its magnitude m and the Gaussian weighted circular window around the keypoint

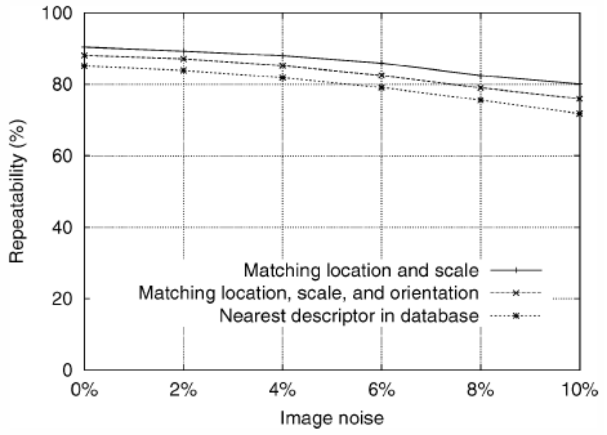

Figure 4: The stability of the keypoints with the orientation assignment.

so that sample points further away from the keypoint are of less influence.

The most dominant direction of the histogram is detected and used as the orientation of the keypoint. If another peak is within 80% of the highest peak, another keypoint is created with that orientation. These "double" keypoint contribute significantly to the stability of the matching. The orientations are interpolated using a parabola function to fit 3 histogram values. Figure 4 shows the repeatability of the orientation assignment.

6. The local image descriptor

All of the previous operations have selected a set of keypoints with a location, scale and orientation. The next step is to compute a descriptor that is highly descriptive as possible while remaining invariant to possible remaining variations such as illumination changes or changes in 3D viewpoint.

One obvious method to do this would be to sample the local image intensities around the keypoint. But Edelman, Intrator and Poggio (1997) proposed a better approach based on biological vision where complex neurons respond to gradients with particular orientations. Their method samples the local gradients, and allows for small positional shifts. This shows to be a very efficient way to describe the local image regions, and is fairly robust to 3D rotations.

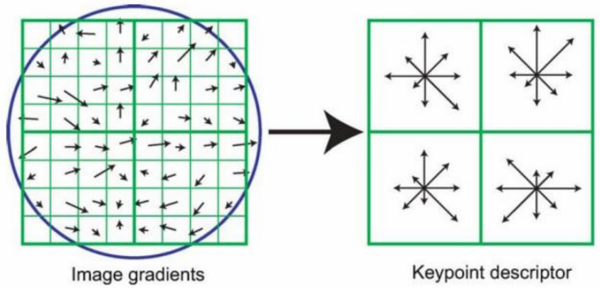

Figure 5: The SIFT keypoint descriptor is calculated from the image gradients in the keypoint region

6.1 Descriptor representation

Figure 5 illustrates the computation of the keypoint descriptor. At the left side is the local image. All of the local image gradient magnitudes and orientations are sampled around the keypoints, which are available from the orientation assignment step. A Gaussian weighting function is again applied to reduce the effect of pixels further away from the keypoint. At the right side is the actual descriptor, it exists of 4x4 histograms (in figure 5 only 2x2 are shown) in which all of the orientations are summarized. Each histogram exists of 8 bins which represent the gradients in that directions.

To allow for small positional shifts, gradient samples contribute not only to the histogram they belong to, but also to the neighboring histograms weighted according to the distance to the other histograms. Also the gradients contribute not only to a single bin within a histogram, but its magnitude is divided amongst a set of bins. The 4x4x8 histogram bins make a feature vector with 128 values for each keypoint.

Finally the descriptor is normalized to unit length. This makes the descriptor invariant for illumination changes and brightness changes. A change in image contrast means that each value is multiplied by a constant, therefore it has no influence on the normalized gradients in the image. A brightness change means that a constant value is added to each pixel. This has no effect at the gradients at all. Non-linear changes in the illumination in 3D rotation for example, do affect the gradient magnitudes however, but not the orientations. Therefore each magnitude is clipped to a value of 0.2 (all magnitudes higher

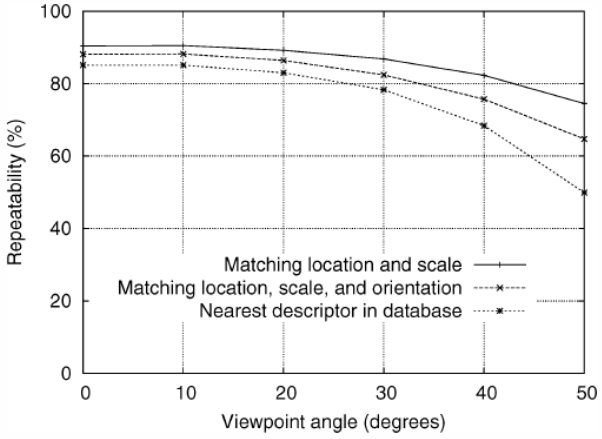

Figure 6: The stability of the keypoints with changing 3D viewpoint

than 0.2 are set to 0.2), and are then renormalized to unit length. Matching the magnitudes is therefore no longer the most important, matching the orientations has greater emphasis.

6.2 Descriptor testing

The descriptor has 2 parameters: *r* the number of orientations in each histogram, and *n* the width and height of the amount of histograms. In a small experiment it was found that a single histogram (*n = 1*) performs very poorly, and the performance improve up to a 4x4 array of histograms (*n = 4)* which is therefore used in this paper. Eight orientations per histograms seems to be a good trade-off between computation time and performance.

In figure 6, the stability of the keypoints is shown for changes in 3D rotation. Even though the image plane is rotated up to 50 degrees, the stability remains above 50%. Mikolajczyk (2002) shows that is it possible to make better affine invariant methods, but even then the performance drops to 40% after 70 degrees of rotation. This means that for reliable object recognition, any object should be trained on from multiple points of view.

When the descriptor matching was tested against large image databases, it showed that on a database with 112 images, with each up to 40,000 keypoints, the percentage of matching location, scale and orientation remained up to 80%. The nearest descriptor in the database matched about 78% of the times.

7. Applications

A very interesting application of the SIFT method is object recognition. When building an object recognition system images are matched against previously seen test images. Because all keypoints have some nearest neighbor in the keypoint database, a keypoint match is considered positive when the ratio between its Euclidian distance and the distance of the second next nearest neighbor is greater than 0.8. When 3 keypoints match according to this test an object is considered recognized. The most computation is done in the nearest-neighbor search. This can be greatly reduced however by using a best-bin-first (BBF) algorithm as described by Beis and Lowe (1997) with losing only up to 5% correct matches.

To even further improve our confidence in the match, a set of matches are clustered. If the clusters match in scale and location compared to the training images, it is so unlikely that it is coincidence, that we can reliably decide the object is in the image.

8. Conclusions

The SIFT keypoints described in this paper are very useful due to their distinctiveness, while being invariant to rotation, scale changes, and being very robust to a large range of other distortions. It can be used to reliably match keypoints against large databases, creating a good basis for object recognition. We showed that an object recognition system works well, even for small objects or occluded objects with a lot of background clutter. The computation is also very efficient: thousands of keypoints can be extracted in near real-time.

A lot of research can still be done to improve the system so it would be more stable against 3D rotations and textures. Also, the method in this paper only considers grayscale images, perhaps a multicolored variant performs better.