Beeldverwerken opdracht 3

Harm Manders (10677186)
Lucas de Vries (10650881)

May 2, 2016

Contents

1 Report

1.1 Description of SIFT

1.1.1 Introduction

SIFT (Scale Invariant Feature Transform) consists of two parts: a detector and a descriptor and was developed by Lowe. His goal was to develop a scale- and rotational invariant operator that can be used for object recognition and matching of images. The basic idea is to transform image data to local feature coordinates, which are invariant to rotation, translation and scale, and (almost) invariant to illumination and viewpoint.
The *detectors* job is to 1) search over all scales and image locations using a difference in Gaussians to find potential (scale/rotalion invariant) points of interest (**Scale-space extrema detection**) and 2) to fit a model to these potential points, to determine theit scale and localtion and eventually decide if they are a keypoint based on their stabilty (**Keypoint localization**).
The *descriptors* job is 3) to compute the best orientations for each keypoint region based on lacal image gradient directions (**Orientation assignment**) and 4) to describe each keypoint region using the image gradients at a certain scale and rotation (**Keypoint descripton**).

1.1.2 Scale-space extrema detection

We want to identify locations and scales in an image that can be assigned from different views of the same scene. To detect locations which are invariant to scale change, we will search for stable features across all posible scales with some continious function of scale (scalespace): the Gaussian. The scalespace of an image $L(x, y, \sigma)$ follows from the convolution of a Gaussian kernel (over multiple scales) with the image.
Scale space is a theory for analysing image structure at different scales through the space scale representation. For a given image $f(x, y)$, its linear Gaussian Scale Space representation is the family of derived signals $L(x, y; \sigma)$ defined by the convolution with the two-dimensional Gaussian kernel

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

such that

$$L(x, y; \sigma) = G(x, y; \sigma) * f(x, y).$$

σ is the scale parameter: image structures of spatial size smaller than σ have mostly been smoothed away at scale σ^2.
Scale space extrema are locations and scales that are identifiable from different views of the same object. There are different techniques to detect these stable keypoints in the scale space, one is Difference of Gaussians, which locates scale-space extrema by computing the difference of two images, one with scale σ, the other with scale $k\sigma$: $D(x, y, \sigma) = L(x, y; k\sigma) - L(x, y; \sigma)$. To detect local maxima and minima, we compare the $DoG(x, y; \sigma)$ to its eigth neighbours with the

same scale, and nine neighbours one scale up, and nine one scale down. When the point is the maxmun or minimum of all these values, it is an extrema. [1]

1.1.3 Keypoint localization

When we have the candidate keypoints, we have to fit the keypoints to nearby data for location and scale. Points with low contrast or poorly localized edges are rejected because they are sensitive to noise.
To do so, we use a quadratic function (a taylor epansion) of the scale-space function, $D(x, y, \sigma)$ with the origin at the sample point:

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

and take te derivative (and set is equal to zero) to obtain:

$$\vec{x} = -\frac{\partial^2 D}{\partial \vec{x}^2}^{-1} \frac{\partial D}{\partial \vec{x}},$$

substituting the latter into the former gives:

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

All extrema with $|D(\vec{x})|$ smaller than 0.03 are thrown away.
Getting rid of the points with low contrast is not enough, as the diffrences in gausians will have strong response along the edges. The Hessian matrix will be used to eliminate edge response.

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

where D_{ij} are (estimated) derivatives from neighbouring points. The pricipal curvatures are proportional to the eigenvalues of the Hessian.Calling α the largest eigenvalue of the Hessian and β the smaller one, and r the ratio $r = \frac{\alpha}{\beta}$, this leads to:

$$\frac{Trace(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(r+1)^2}{r}.$$

Here we've calculated the ratio, avoiding explicitly computing the eigenvalues (we don't care about those). $\frac{(r+1)^2}{r}$ is at its minimum wen $\alpha = \beta$ and increases as r increases. So, to check that the ratio of principal curvatures is below a certain threshold we just have to check whether $\frac{(r+1)^2}{r} > \frac{Trace(\mathbf{H})^2}{Det(\mathbf{H})}$.

[1] Question 3.3, see Appendix

1.1.4 Orientation assignment

The first part of the descriptor deals with the orientation of each keypoint. We assign a consistant orientation to each keypoint, depending on local structure. The discriptor can be represented relative to this orientation and the descriptor consequently is rotational invariant.
The scale of the keypoint is already know, and this scale is used for all following computations, this way the descriptor is also scale-invariant. For every sample the scale space, gradient magnitude and orientation are computed using pixel differences. $L(x, y, \sigma)$ (at the known scale), $m(x, y)$(magnitude gradient) and $\theta(x, y)$ (orientation):

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

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

Next we create a histogram (36 bins, one per 10 degrees) of all local gradient directions at that scale. Every sample added is weighted by m(x,y), the gradient magnitude, and by a Gaussian weighted circular window with 1.5σ.
The highest histogram peak correlates to the most leading directions of local gradients. If there are more peaks of approximately the same magnitude (within 80% of the highest peak), a new keypoint will be made for that peak. So if there are two peaks, two keypoints will be created with the same scale and location, but differing orientations. To get the most accurate peak position, we fit a parabola to the histogram to find its peak orientation.

1.1.5 Keypoint description

Now each keypoint has a location, scale an orientation. Next we want to compute a descriptor, that is distinctive and invariant (as much as possible) to changes in viewpoint and illumination. First, we calculate the gradient magnitude and orientation at each image sample point (16x16 sample grid) in the area around the keypoint and we weigh these by a gaussian to make sure the central point have greater influence. Histograms are made and the orientation samples are accumulated to a 4x4 descriptor (using 16 histograms) with 8 orientations. Therefore there are now 4x4x8 = 128 features for each keypoint.
Normalizing the feature vector to unit length will make the descriptor invariant to changes in illumination (contrast/brightness), because normalization gets rid of the constant (when increasing/decreasing contrast) with which all pixel values are mulitplicated. And because we use pixel differences for the gradient, the constant added to all pixel values (when performing a brightness change) will be lost anyways.
Non-linear illumination (due to changes in viewpoint) changes are tackled by using a theshold for the values in the unit feature vector to reduce the influence of large gradients, and thus highlighting the importance of the distribution of orientations.

1.1.6 Keypoint matching

Now all is left is deciding what the best match for each keypoint is in the database of keypoints from training data, using the nearest neighbour (keypoint with minimum euclidian distance for the descriptor vector). Each keypoint is matched with the database of keypoints (from training images), but many will fail due to noise or background clutter. We want to eliminate features that do not have a good match due to background noise. A way to do that is by looking at the second closest neighbour to the closest neighbour.
There are still a lot of featues from valid object, after discarding the background noise. We still need to find the small percentage of good matched. Consequently, clusters of (at least) three features are tested first, because their chance to be a correct match is higher than for single keypoints. We can filter this using RANSAC or the Hough Transform.

1.2 SIFT scale and rotation invariant

Because we are searching for stable features through all possible scales, and use the same scale, once the good scale is found, for all following computations, SIFT is scale invariant. And because the discriptor can be represented relative to the orientation, rotational invariance is achieved.

2 Theory Questions

3.3 Scale space is a theory for analysing image structure at different scales throught the space scale representation. For a given image $f(x, y)$, its linear Gaussian Scale Space representation is the family of derived signals $L(x, y; \sigma)$ defined by the convolution with the two-dimensional Gaussian kernel

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}},$$

such that

$$L(x, y; \sigma) = g(x, y; \sigma) * f(x, y).$$

σ is the scale parameter: image structures of spatial size smaller than σ have mostly been smoothed away at scale σ^2.
Scale space extrema are locations and scales that are identifiable from different views of the same object. There are different techniques to detect these stable keypoints in the scale space, one is Difference of Gaussians, which locates scale-space extrema by computing the difference of two images, one with scale σ, the other with scale $k\sigma$: $D(x, y, \sigma) = L(x, y; k\sigma) - L(x, y; \sigma)$. To detect local maxima and minima, we compare the $DoG(x, y; \sigma)$ to its eigth neighbours with the same scale, and nine neighbours one scale up, and nine one scale down. When the point is the maxmun or minimum of all these values, it is an extrema.

3.4 Now, from Lowe we know that from the heat equation and the finite difference approximation of the nearby scales σ and $k\sigma$ follows:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} = \frac{G(x,y;k\sigma) - G(x,y;\sigma)}{k\sigma - \sigma}$$

and therefore

$$G(x,y;k\sigma) - G(x,y;\sigma) = (k-1)\sigma^2\nabla^2 G.$$

Now:

$$\begin{aligned} D(x,y,\sigma) &= L(x,y;k\sigma) - L(x,y;\sigma) \\ &= [G(x,y;k\sigma) - G(x,y;\sigma)] * f(x,y) \\ &= (k-1)\sigma^2\nabla^2 G * f(x,y). \end{aligned}$$

Which completes the loop.

3.5 To get the eigenvalues we calculate $\mathbf{H} - \lambda\mathbf{I} = 0$.

$$\det\begin{bmatrix} f_{xx}-\lambda & f_{xy} \\ f_{xy} & f_{yy}-\lambda \end{bmatrix} = 0$$

So,

$$\begin{aligned} (f_{xx}-\lambda)(f_{yy}-\lambda) - f_{xy}^2 &= 0, \\ f_{xx}f_{yy} - \lambda f_{xx} - \lambda f_{yy} + \lambda^2 - f_{xy}^2 &= 0, \\ f_{xx}f_{yy} - \lambda(f_{xx}+f_{yy}) + \lambda^2 - f_{xy}^2 &= 0, \\ \lambda^2 - \lambda(f_{xx}+f_{yy}) + (f_{xx}f_{yy} - f_{xy}^2) &= 0. \end{aligned}$$

The eigenvalue $\lambda = 0$ counts if $f_{xx}f_{yy} = f_{xy}^2$. The other two eigenvalues are:

$$\lambda = \frac{(f_{xx}+f_{yy}) \pm \sqrt{(f_{xx}+f_{yy})^2 - 4(f_{xx}f_{yy} - f_{xy}^2)}}{2}$$

And so, $\lambda^+ + \lambda^- = \frac{f_{xx}+f_{yy}}{2} + \frac{f_{xx}+f_{yy}}{2} = f_{xx} + f_{yy}$, which is the trace of the Hessian.
Diagonalization is always possible for real symmetric matrices. The Hessian is always real and symmetric, so diagonalization is always possible.
3.8 The $\frac{1}{2}$ is not a typo, if you substitute formula (3) into (2) you get

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