Beeldverwerken opdracht 3

Paul Schmidt (11850590)
Elise Sterk (10790330)
TA: Thomas Bellucci

6 Mei 2019

Introduction

This paper is about the SIFT implementation described by Lowe (2004)in his article. SIFT is a method for detecting stable feature points within an image and has been very instrumental for the field of computer vision. In this paper SIFT will be explained in detail and used to make a mosaic out of two Nachtwacht images and combine the two images into one image.
Furthermore, a detailed description and an implementation of the RANSAC method will also be included in this paper. The RANSAC algorithm is an algorithm that that can fit a model to a data set while excluding the outliers in its fit. This algorithm is then used to decide which matching points to use in the projective transformation for the Nachtwacht images.

Theory

SIFT explanation Scale Invariant Feature Transform (SIFT) is an algorithm that detects and describes image features in a way that is invariant of scale, rotation, illumination and viewpoint.
The first step is to create the scale space of the image. This is done by repeatedly blurring the image using the Gaussian convolution kernel. Next, the image is resized to half its original size, creating a new octave, at which point the blurring is once again applied. The articles suggests 4 octaves, each containing 5 scales, for optimal results.
Next, the Laplacian of the Gaussian (LoG) is approximated in order to find key points in the image. The LoG is effective in finding points of interest, but inefficient to calculate. Thus, a close approximation is

used, the Difference of Gaussians (DoG). The DoG is defined in the following formula:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

where k is a small factor, effectively approximating the derivative over the scale space.
This DoG is used to find keypoints in a simple manner. Each sample point is compared to points in its 3x3x3 neighbourhood (so a 3x3 neighbourhood in the current and directly adjacent scales). If it is strictly higher or lower than all 26 other values, it is classified as a keypoint. Finally, quadratic Taylor expansion is used to find the interpolated location of the extrema.
This yields a high number of key points, some of which are of low quality due to either the nature or contrast intensity of the keypoint. Two measures are taken to increase the quality of the keypoints. Firstly, if the contrast intensity of an extremum is below a certain threshold, the keypoint is dismissed. Secondly, edge keypoints are removed, since they can be poorly localized and unstable when noise is involved. This is done by taking two perpendicular principal curvatures at a keypoint. An edge point will have a high ratio r between the magnitude of the two principal curvatures, so any keypoints with a ratio above a certain threshold are dismissed. Mathematically, the ratio between the principal curvatures are retrieved using the 2x2 Hessian matrix.
To provide rotation invariance, the keypoint's orientation is determined. This is done by collecting gradient directions and magnitudes around the keypoint and forming an orientation histogram based on this. 36 bins are formed to represent the full 360 degrees range, and the peak of this histogram, along with any peaks above 80% of the peak, form the orientation of a keypoint.
Finally, a unique description for each keypoint is made, in much the same way as how the orientation is calculated. For each keypoint, a 16x16 window around the keypoint is split up into 16 4x4 windows. An 8 bin histogram is created for each window, resulting in 8x16 = 128 orientations, forming the feature vector. Lastly, the keypoint orientation is substracted from this vector, thus achieving rotation invariance.

SIFT invariance explanation The SIFT is scale invariant because of the DoG function. The DoG function is an approximation of a normalized Laplacian Gaussian ($\sigma^2\Delta^2 G$) and by calculating the DoG for different

scales the DoG finds features for the different scales and thereby ensures scale invariance.
It is rotation invariant because of the keypoint orientation. In the keypoint orientation step we already have keypoints, for every keypoint an orientation histogram is created based on a window of pixels around the keypoint. From this histogram the most prominent orientation is calculated and assigned to the keypoint. The feature vector is based on the prominent orientation and the promiment orientation is assigned for every keypoint. Therefore rotations have no influence and is SIFT rotation invariant.

Match projection The projection matrix was estimated using 4 coordinates from 'nachtwacht1.jpg'. In a qualitative comparison of the projected coordinates and the true locations of the matches in 'nachtwacht2.jpg', the majority of the results are close to the true location. There is always some deviation, but the projection is approximately correct. Incorrect matches stand out, since their projected values are nowhere near the true locations in the original image. When the projection is based on one of these matches, the projection will only succeed for the coordinates that are used for projection. Since the projection is not correct, the other matches will not be projected successfully.

RANSAC explanation Random Sample Consensus (RANSAC) is a method to find the best model for a data set. The advantage of RANSAC is that it can handle outliers. If we were to select a random sample of data and use the least square method to find a model that fits this sampled data, then if the sample data included outliers, the least square method would not be able to find a good fit for this data, since the least square method includes the outliers in its calculations. The RANSAC method has a way to deal with these outliers and to ensure that outliers have no influence on the model fit for the sampled data. The RANSAC algorithm achieves this by finding a model that fits as many inliners as possible and excludes the outliers. This algorithm and its parameters are explained in the next few paragraphs and the explanation is based on the lecture by professor Dorst.
To find the optimal model fit, the RANSAC algorithm first randomly selects a subset (with at least size s data samples, which depends on the operation) of the data. Then a model is fitted for this sample data. This model is used to test all the data points that were not

included in the sample. For all these data points the distance from the data point to the fitted model is calculated. If this distance is less than a specified threshold (d), the data point is labeled as 'close'. The amount of 'close' data points is compared to another threshold (T). If the amount is higher than this threshold, then the model is a good fit and is optimized by fitting the model again for all the data points that were labeled 'close'. For the first loop of the RANSAC algorithm, this model is then saved as the 'best' model. If the amount is lower than T, then the model is not a good fit and gets discarded.
This process is repeated a specific number of times (N), of which each loop will result in either a model that fits the data sufficiently well, or a model that does not. If it results in a good model fit, than that model is compared to the 'best' model. The model that fits the most data points is then saved as the 'best' model. After repeating this process a specific amount of times, the model that fits the data best will be returned, resulting in a model that fits the data and excludes the outliers.
For the projection in this paper RANSAC was applied by taking four matching points from both Nachtwacht images. A projection matrix was created for these matching points and then RANSAC was used to determine which projection matrix results in the most inliers.

Number of iterations : The number of iterations can be calculated with the equation from the lecture, namely: $N = \frac{log(1-p)}{log(1-(1-e)^s)}$. Here p is the probability that we get a sample with only inlier data points, which is 0.99. e is the probability that a data point is an outlier. This depends on the amount of outliers in the data.

Algorithm

SIFT

All SIFT functionality is well documented in MATLAB. sift.m functions as a main file for the SIFT implementation, performing SIFT feature detection, description and matching. There are further calls to plotMatches.m and projectKeypoints.m, which perform self-explanatory tasks. These function are also commented for further clarification.

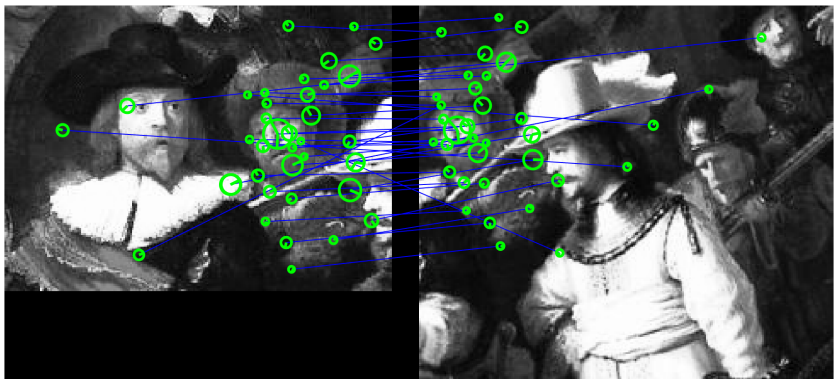
RANSAC

Pseudo code RANSAC:
iterations = 0
bestmatrix = [] (Projection matrix which results in the most inliers)
    while iterations is smaller than N
        data1 = four random matching points from image 1
        data2 = four random matching points from image 2
        matrix = createProjectionmatrix
        For all points of data1:
            apply matrix
            calculate distance points to matrix
            if distance is smaller than d:
                point is labeled as close and added to close list
        if size(close) is bigger than T:
            bestmatrix = data1
            iterations += 1
    return bestmatrix

Experiments

The result of demo_mosaic with our own implementation of the projection matrix:

The matches found by vlfeat visualized:



Conclusion

In general, the SIFT algorithm in combination with the RANSAC algorithm is a very good algorithm for feature matching in images. The advantages of SIFT is that it is rotation and scale invariant. The main advantage of RANSAC is that the algorithm can exclude outliers while model fitting. The combination of the two algorithms can succesfully result in the merging of the two Nachtwacht images into one image.

Appendix

2 Projectivity

3 Sift theory questions

3.3 A scale space is used to focus on certain aspects of an image based on scale. This is done by blurring the image repeatedly for various scales. To explain scale space extrema:
We take a pixel in the image after appliance of the DoG function and compare its value to all values in a 3x3x3 neighbourhood (so a 3x3 neighbourhood in the current and directly adjacent scales). If the pixel's value is the lowest or highest of its neighbors, then it is a local minimum or maximum. This is repeated for all the pixels in an image. These local extremas are the scale space extremas of the image. Intu-

itively this means that the scale space extremas are an image's highest and lowest pixel values.

3.4 D is defined as:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

$G(x, y, k\sigma) - G(x, y, \sigma)$ is thus the convolution kernel applied to obtain $D(x, y, \sigma)$ from I(x,y).
In the following calculations Lowe shows that $G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2$.
It is thus shown that the convolution kernel used to obtain D only differs from the scale-normalized LoG by a factor of k-1.

3.5 The Hessian is a square n x n matrix and the trace of a matrix is equal to the sum of its eigenvalues. What we need to proof is thus that trace(H) = $\sum_{i=1}^{n} \lambda_i$:

For the hessian matrix (H) there exists an invertible matrix X so that we can use the formula XHX^{-1} to calculate the normal form (N).
The trace of a matrix is the sum of its diagonal entries. The normal form matrix (N) has the eigenvalues as its diagonal entries, thus we can write:
trace(N) = $\sum_{i=1}^{n} \lambda_i$ = trace(XHX^{-1}).
As stated in the lecture, trace(ABC) = trace(BCA). This means that we can state that trace(XHX^{-1}) = trace($HX^{-1}X$) = trace(H).
Thus proving that trace(H) = $\sum_{i=1}^{n} \lambda_i$.
It is always possible to diagonalize a Hessian matrix, because a Hessian matrix is always symmetric.

3.6 the parameters for SIFT implementation with magic numbers are the following:

octave = magic number 4.

s = number of scales per octave, with magic number 5.

threshold for contrast intensity, where every extrema with a contrast intensity below this threshold is discarded, with magic number 0.03.

r ratio between eigenvalues of the Hessian matrix of D, with magic number 10.

Scaling by a random amount between 0.2 and 0.9.

3.8 To make it easier to write and understand, in the equations, the gradient of D and the Hessian are replaced with G and H.
So the equations are then:
$D(x) = D + G^T x + \frac{1}{2} x^T H x$
$\hat{x} = -H^{-1} \cdot G$
$D(\hat{x}) = D + \frac{1}{2} G^T \hat{x}$
We substitute the second equation for the second x in the first equation:
$D(\hat{x}) = D + G^T x + \frac{1}{2}(-H^{-1} \cdot G)^T H x$
The transpose of the Hessian is the Hessian itself, since it is a symmetric matrix, so:
$D + G^T x + -\frac{1}{2} H^{-1} \cdot G^T \cdot H x$
$H^{-1} H = 1$. So:
$D + G^T x - \frac{1}{2} G^T \cdot 1 x$
$D + G^T x - \frac{1}{2} G^T x$
Now if we again substitute the x-en with $\hat{x}$ then we get:
$D(\hat{x}) = D + \frac{1}{2} G^T \hat{x}$

3.9 We are in 3D, because we also need to look at the feature vector of every keypoint. This vector is a 3D vector, thus we need to be in 3D.

3.10 Changes in brightness that can be described in a linear fashion.