IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Implementation of Scale Invariant Feature Transform (SIFT) Algorithm Using Content Based Image Retrieval

Ms.S. Poovizhi*[1], Ms. Sandhiya[2]

*[1,2] Assistant Professor, RMK College of Engineering and Technology, Puduvoyal, India

poovs1@gmail.com

Abstract

Successful retrieval of relevant images from large-scale image collections is one of the current problem in the field of data management. Content-Based Image Retrieval (CBIR), also known as Query by Image Content (QBIC) is the application to solve image retrieval problem, that is, the problem of searching for digital images in large databases. SIFT (Scale Invariant Feature Transform) has been identified as a promising algorithm for this application.. The algorithm is specifically tested for its feasibility for finding matches between two different images.

Keywords: SIFT (Scale Invariant Feature transform), CBIR (Content Based Image Retrieval), NNS (Nearest Neighbour Search), QoM (Quality of Match).

Introduction

Image processing now a day finds its application in all fields around us. Database related to images is on increasing. SIFT is an image processing algorithm which can be used to detect distinct features in an image. Once features have been detected for two different images, one can use these features to answer questions like “are the two images taken of the same object?” and “given an object in the first image, is it present in the second image?” Thus, the feasibility of SIFT algorithm for CBIR is tested and used in image retrieval process.

Content Based Image Retrieval

Content-based image retrieval uses the visual contents of an image such as colour, shape, texture, and spatial layout to represent and index the image. In typical content-based image retrieval systems (Figure 1), the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors. The similarity distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. The indexing scheme provides an efficient way to search for the image database. Recent retrieval systems have incorporated users relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results.

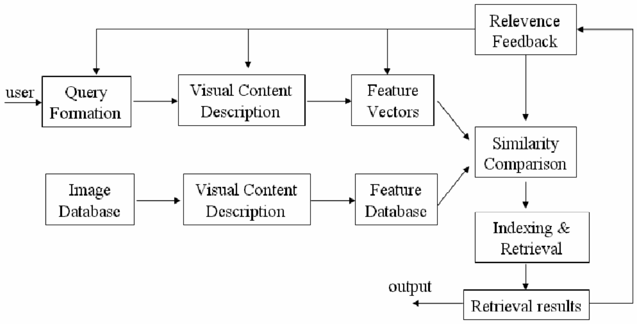

Figure 2.1 Diagram for content-based image retrieval system

Image Content Descriptors

General visual content include color, texture, shape, spatial relationship, etc. Domain specific visual content, like human faces, is application dependent and may involve domain knowledge. Semantic content is obtained either by textual annotation or by complex inference procedures based on visual content. A good visual content descriptor should be invariant to the accidental variance introduced by the imaging process (e.g., the variation of the illumination of the scene). However, there is a trade-off between the invariance and the discriminative power of visual features, since a very wide class of invariance loses the ability to discriminate between essential differences. Invariant description has been largely investigated in computer vision (like object recognition), but it is relatively new in image retrieval.

A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions or objects to describe the image content. To obtain the local visual descriptors, an image is often divided into parts first. The simplest way of dividing an image is to use a partition, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using region segmentation algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete object segmentation to obtain semantically meaningful objects (eg. ball, car, horse). Currently, automatic object segmentation for broad domains of general images is unlikely to succeed.

SIFT Algorithm

The SIFT algorithm identifies features of an image that are distinct, and these features can in turn be used to identify similar or identical objects in other images. SIFT has four computational phases. The reason for this being that some computations performed by SIFT are very expensive. The cost of extracting the keypoints is minimized by the cascading approach of SIFT. The more expensive operations are only applied on locations that pass an initial, cheaper test. The output of the SIFT algorithm is a set of keypoint descriptors. Once such descriptors have been generated for more than one image, one can begin image matching *(Figure 4.1)*. The image matching, or object matching, is not part of the SIFT algorithm. For matching we use a Nearest Neighbour Search (NNS), an algorithm that is able to detect similarities between keypoints. Thus, SIFT only makes matching possible by generating the keypoint descriptors.

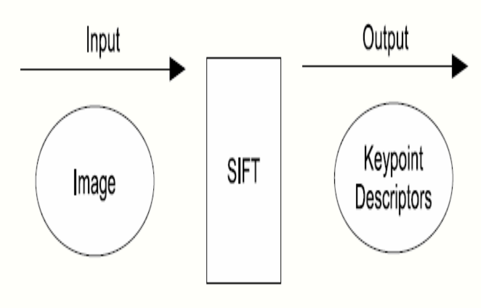

Figure 4.1 SIFT takes as input an image, and generate a set of keypoint descriptors. The keypoint descriptors may then be stored in a separate file.

Although the matching process is not part of SIFT, the results from image matches are used as an indicator of how well the SIFT algorithm is suited for image matching. *(Figure 4.2)* shows that in an image match checking, the two sets of keypoint descriptors are given as input to a nearest neighbour search algorithm. The output of the algorithm is a set of keypoint descriptors found to be very similar

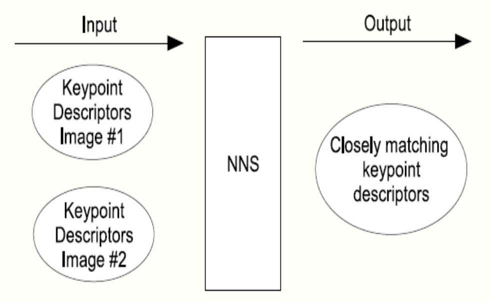
Input
Output
Keypoint Descriptors Image #1
Keypoint Descriptors Image #2
NNS
Closely matching keypoint descriptors

Figure 4.2 In an image match checking, the two sets of keypoint descriptors

Keypoints and Keypoint Descriptors

The SIFT algorithm produces keypoint descriptors. A keypoint is an image feature which is so distinct that image scaling, noise, or rotation does not, or rather should not, distort the keypoint. If one scales the image to half the size, or double the size, the keypoint would still be identifiable. The same goes for image rotation and noise. If an image is, for example, rotated clockwise, the keypoint would still persist.

A keypoint descriptor is a 128-dimensional vector that describes a keypoint. The reason for this high dimension is that each keypoint descriptor contains a lot of information about the point it describes. In Figure 4, blurry images in which SIFT has detected 240 keypoints (The image can also be found in the lower right hand corner of *Figure 5.1*). The blue circles represent a keypoint found, along with its scale.

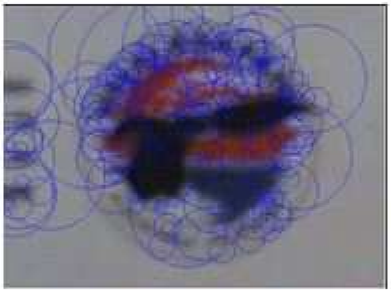

Figure 5.1 An image processed by the SIFT algorihm. 240 keypoints have been identified

Scale Space Extrema Detection

The first phase of the computation seeks to identify potential interest points. It searches over all scales and image locations. The computation is accomplished by using a Difference-of-Gaussian (DoG) function. The resulting interest points are invariant to scale and rotation, meaning that they are persistent across image scales and rotation. *Figure 6.1*shows how the DOG is calculated.

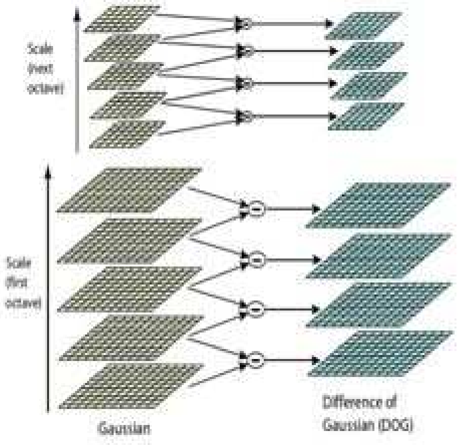

Figure 6.1 Explains how Difference of Gaussian is calculated

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

Where * is the convolution in x and y

$G(x, y, k\sigma)$ - variable-scale Gaussian

$I(x, y)$ input image

To detect stable keypoint locations, find the scale-space extrema in difference-of-Gaussian function. *Figure 6.2* shows the Scale space extrema detected for a particular image.

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

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

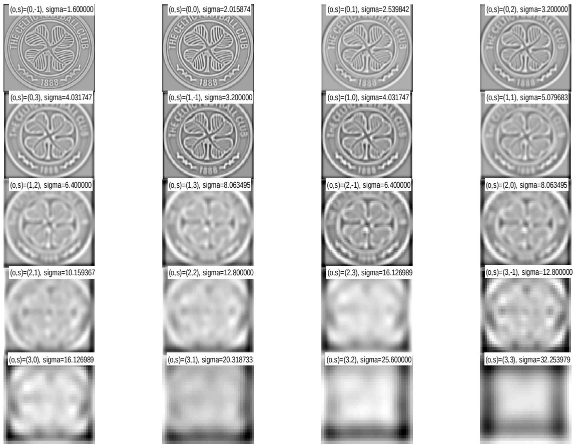
(o,s)=(0,-1), sigma=1.600000
(o,s)=(0,0), sigma=2.015874
(o,s)=(0,1), sigma=2.539842
(o,s)=(0,2), sigma=3.200000
(o,s)=(0,3), sigma=4.031747
(o,s)=(1,-1), sigma=3.200000
(o,s)=(1,0), sigma=4.031747
(o,s)=(1,1), sigma=5.079683
(o,s)=(1,2), sigma=6.400000
(o,s)=(1,3), sigma=8.063495
(o,s)=(2,-1), sigma=6.400000
(o,s)=(2,0), sigma=8.063495
(o,s)=(2,1), sigma=10.159367
(o,s)=(2,2), sigma=12.800000
(o,s)=(2,3), sigma=16.126989
(o,s)=(3,-1), sigma=12.800000
(o,s)=(3,0), sigma=16.126989
(o,s)=(3,1), sigma=20.318733
(o,s)=(3,2), sigma=25.600000
(o,s)=(3,3), sigma=32.253979

Figure 6.2 Scale Space Extrema detected for a particular image.

Keypoint Localization

For all interest points found in phase 1, a detailed model is created to determine location and scale. Keypoints are selected based on their stability. A stable keypoint is thus a keypoint resistant to image distortion. Take Taylor Series Expansion of scale-space function D(x,y,σ)
Use up to quadratic terms

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

origin shifted to sample point

$$x = (x, y, \sigma)^T$$

- offset from this sample point
- to find location of extremum, take derivative and set to 0

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

Orientation Assignment

For each of the keypoints identified in phase 2, SIFT computes the direction of gradients around. One or more orientations are assigned to each keypoint based on local image gradient directions.

Keypoint Descriptor

The local image gradients are measured in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. There are nine parameters one can assign to adjust what criteria SIFT uses on its four-step way to identify keypoints. Still, the parameters used are the ones Lowe set forth as optimal, and his recommendations are based on empirical studies.

ISSN: 2277-9655
Impact Factor: 1.852

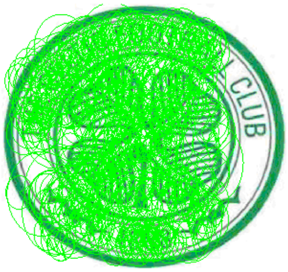

Figure 9.1 Keypoint descriptor generated for an image

Image Matching

The three possible image matches are,

1. A match where the whole of one image matches the whole of another image.
2. Part of one image matches the whole of another image.
3. Part of one image matches the part of another image.

The different matches will have different characteristics. If there is a match in case 1, a fairly large percentage of all keypoints are matched. In case 2, there is a large percentage match in the image with a whole match, and a small percentage match in the image partly matching. In case 3, there is a fairly low percentage of keypoints matching in both images.

Lines between matching keypoints have been drawn on top of the images for ease of understanding. The matches are not always correct, so some lines will point to non-matching locations. In *Figure 10.1*, there is a matching of two images that have a lot of similarity. As expected, a large number of keypoints were identified as matching. The image on top has been taken with a mobile phone camera, and the image below has been scanned.

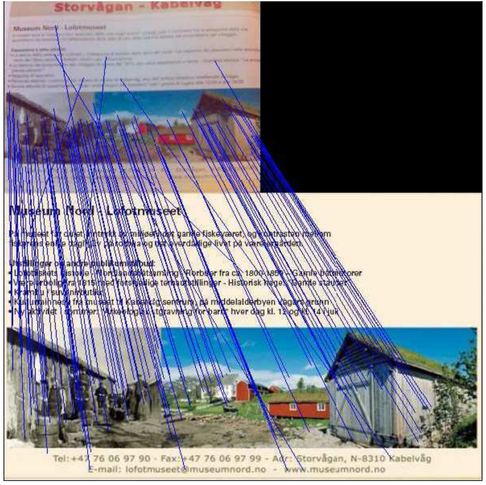
Storvågan - Kabelvåg
Museum Nord - Lofotmuseet
Museum Nord - Lofotmuseet
Tel:+47 76 06 97 90 - Fax:+47 76 06 97 99 - Adr: Storvågan, N-8310 Kabelvåg
E-mail: lofotmuseet@museumnord.no - www.museumnord.no

Figure 10.1 Two images with a large match percent. A total of 100 keypoints were identified as matching

An example of case 2 in figure 2.10. Again, the image on the left is taken with a mobile camera, and the image on the right has been extracted from PDF. Note that, although there are many hits, there is also one obvious miss. The almost vertical line going from the grass in the picture on the left and to the hills in the image on the right is falsely identified as a match.

To determine whether two images are similar, or contain a similar object, can perform a nearest neighbour search (NNS) to identify similar key point descriptors. A key point descriptor is considered a neighbour to another if they have many characteristics in common. Remember that a keypoint descriptor is a 128-dimensional vector. Thus, if two such vectors are similar, they are likely to be a description of two similar objects.

A common way to perform an NNS is to first create a kd-Tree, and then traverse it. It is not necessary to see how this tree is constructed, but it is worth mentioning that the running time of such a search is O(n log n). This is a lot better than the running time of a linear search, which is O(n^2). After all the neighbouring keypoint descriptors are identified, a Hough transform is performed on the set of matching keypoints. The purpose of the Hough-transform is to filter out false matches. Here only a few keypoint matches can be sufficient for identifying two images as a match, but this would be impossible be having a few keypoints been removed by a Hough-transform.

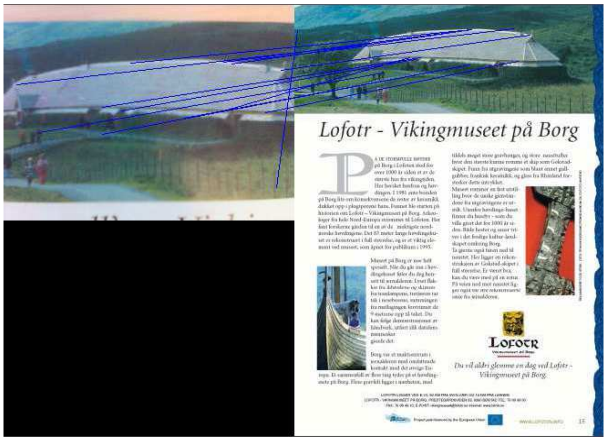
Lofotr - Vikingmuseet på Borg
Lofotr
Du vil aldri glemme en dag ved Lofotr - Vikingmuseet på Borg

Figure 10.2 Two images with a small match percent. A total of 15 keypoints were identified as matching.

RESULTS

The result of the project is classified into three different demos.

i) Keypoint generation for a particular image

ii) Matching of similar keypoints between two images

iii) Retreival of images based on the content of the query image.

Retreival of images based on the content of the query image

Keypoint Generation for a Particular Image

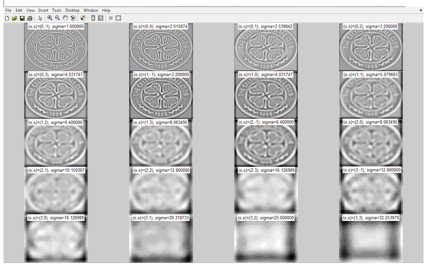

Figure12.1 Detection of Scale Space Extrema

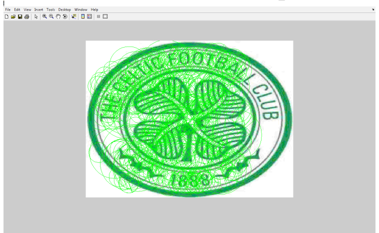

Figure12.2 keypoint generation for an image

Matching of Similar Keypoints between Two Images

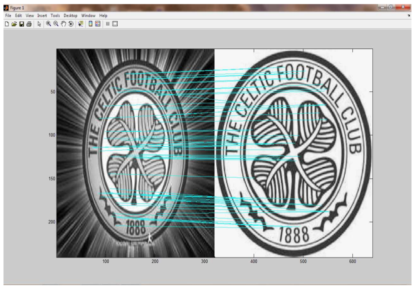

Figure 13.1 Output for two different images with more number of keypoints matching.

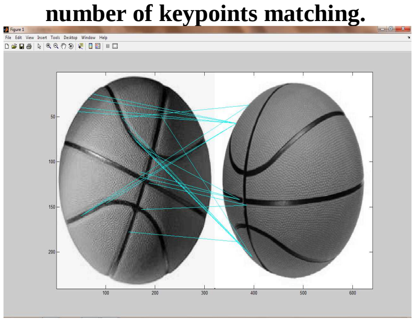

Figure 13.2 Output for two different images with less number of keypoints matching.

Retrieval of Images Based on the Content of the Query Image

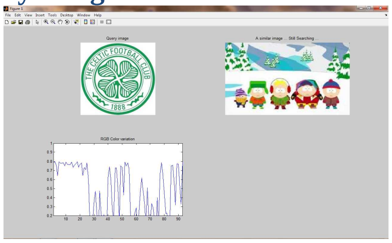

Figure 14.1 window shown while comparing query image with that of the all other images in the database.

Here, the closely related 4 images are retrieved from the group of images.

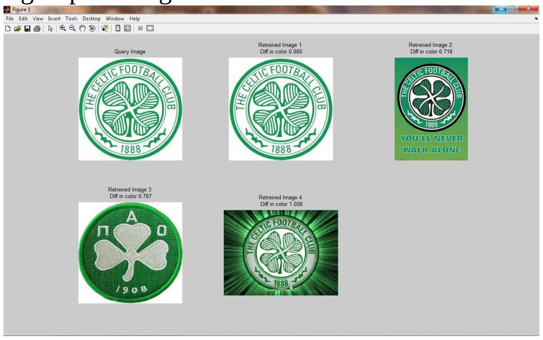

Figure 14.2 Four closely matched retreived Images.

Here, the closely related 10 images are retrieved from the group of images.

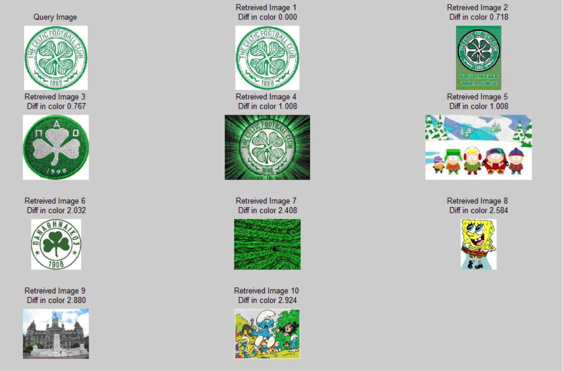

Figure 14.3 Ten closely matched retrieved Images.

The SIFT algorithm has gone through extensive testing, and here we will present all results and findings. The first category consists of image tests done for the reason of curiosity. The second category has to do with timing and keypoint generation. In the third category all image matches have been done on transformed images. An image transformation can be, for example, to remove colour, scale the image up or down, and so on.

Conclusion

The SIFT keypoints described in this paper are particularly useful due to their distinctiveness, which enables the correct match for a keypoint to be

ISSN: 2277-9655
Impact Factor: 1.852

selected from a large database of other keypoints. The keypoints have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. Computation of keypoint is efficient when all the three features (color, texture, shape) are taken into consideration, so that several thousand keypoints can be extracted from a typical image with near real-time performance on standard PC hardware. Thus, the SIFT algorithm can be used in real time content based image retrieval process.

References

[1] *Lowe, D.G, 2004, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 60.*

[2] *Schürmann, A, Aarbakke, A S, Egeland, E, Evjemo, B and Akselsen, S.2006. Let a picture initiate the dialog! A mobile multimedia service for tourists (pdf). Fornebu, Telenor Research & Innovation. Report R&I N33/2006.*

[3] *Brown, M. and Lowe, D.G. 2002. Invariant features from interest point groups. In British Machine Vision Conference, Cardiff, Wales, pp. 656-665.*

[4] *Akselsen, S, Bjørnvold, T A, Egeland, E, Evjemo, B, Grøttum, K J, Hansen, A A, Høseggen, S, Munkvold, B E, Pedersen, P E, Schürmann, A, Steen T, Stikbakke, H, Viken, A and Ytterstad, P. 2006. MOVE - a summary of project activities and results (2004-2006). Fornebu, Telenor Research & Innovation.*

[5] *Ballard, D.H. 1981. Generalizing the Hough transform to detect arbitrary patterns. Pattern Recognition, 13(2):111-122.*

[6] *Anne Staurland Aarbakke. 2007. M2S and CAIR: Image based information retrieval in mobile environments. Masteroppgave i informatikk. Mai 2007. Institutt for Informatikk, Det matematisk-naturvitenskapelige fakultet, Universitetet i Tromsø.*

[7] *Arya, S., and Mount, D.M. 1993. Approximate nearest neighbor queries in fixed dimensions. In Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'93).*

[8] *Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., and Wu, A.Y. 1998. An optimal algorithm for approximate nearest neighbor searching. Journal of the ACM, 45:891-923.*

[9] *Basri, R., and Jacobs, D.W. 1997. Recognition using region correspondences. International Journal of Computer Vision, 25(2):145-166.*

[10] *Baumberg, A. 2000. Reliable feature matching across widely separated views. In Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, pp. 774-781.*

[11] *Beis, J. and Lowe, D.G. 1997. Shape indexing using approximate nearest-neighbour search in highdimensional spaces. In Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 1000-1006.*

[12] *Carneiro, G., and Jepson, A.D. 2002. Phase-based local features. In European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp. 282-296.*

[13] *Hartley, R. and Zisserman, A. 2000. Multiple view geometry in computer vision, Cambridge University Press: Cambridge, UK.*

[14] *Hough, P.V.C. 1962. Method and means for recognizing complex patterns. U.S. Patent 3069654.*

[15] *Koenderink, J.J. 1984. The structure of images. Biological Cybernetics, 50:363-396.*

[16] *Lindeberg, T. 1993. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. International Journal of Computer Vision, 11(3): 283-318.*

[17] *Lindeberg, T. 1994. Scale-space theory: A basic tool for analysing structures at different scales. Journal of Applied Statistics, 21(2):224-270.*

[18] *Lowe, D.G. 1991. Fitting parameterized three-dimensional models to images. IEEE Trans. on Pattern Analysis and Machine Intelligence, 13(5):441-450.*

[19] *Lowe, D.G. 1999. Object recognition from local scale-invariant features. In International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157.*

[20] *Lowe, D.G. 2001. Local feature view clustering for 3D object recognition. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, pp. 682-688.*

[21] *Luong, Q.T., and Faugeras, O.D. 1996. The fundamental matrix: Theory, algorithms, and stability analysis. International Journal of Computer Vision, 17(1):43-76.*

[22]Matas, J., Chum, O., Urban, M., and Pajdla, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. In British Machine Vision Conference, Cardiff, Wales, pp. 384-393.

[23]Mikolajczyk, K. 2002. Detection of local features invariant to affine transformations, Ph.D. thesis, Institut National Polytechnique de Grenoble, France.

[24]Mikolajczyk, K., and Schmid, C. 2002. An affine invariant interest point detector. In European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp. 128-142.