# A FAST INTEREST POINT DETECTION ALGORITHM

by

AARON J. CHAVEZ

B.S., Kansas State University, 2006

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2008

Approved by:

Major Professor
David A. Gustafson

# Abstract

An interest point detection scheme is presented that is comparable in quality to existing methods, but can be performed much faster. The detection is based on a straightforward color analysis at a coarse granularity. A 3x3 grid of squares is centered on the candidate point, so that the candidate point corresponds to the middle square. If the color of the center region is inhomogeneous with all of the surrounding regions, the point is labeled as interesting. A point will also be labeled as interesting if a minority of the surrounding squares are homogeneous, and arranged in an appropriate pattern.

Testing confirms that this detection scheme is much faster than the state-of-the-art. It is also repeatable, even under different viewing conditions. The detector is robust with respect to changes in viewpoint, lighting, zoom, and to a certain extent, rotation.

# Table of Contents

# List of Figures

# List of Tables

# CHAPTER 1 - Introduction

Interest point detection, along with feature description, is part of the general object recognition problem. This problem, that of detecting arbitrary objects in real-world images, is well-documented. It is difficult in part because of the large amount of information present in even the most basic of images. Not all of this information is useful in object recognition, however, and so simplifying the data set before analysis is essential. This is the purpose of interest point detection.

Interest point detection is an important technique for reducing the complexity of image data. Most points in an image are homogeneous with their surroundings. Thus, two nearby points are unlikely to provide complementary information. If we only consider a set of points which are inhomogeneous with their surroundings, we significantly reduce the complexity of analysis. At the same time, it is hoped that most of the useful information in the image is preserved.

Another difficulty with object recognition is that there are often real-time constraints imposed on the problem. The speed which is necessary varies depending on the application. In some circumstances, a slower, more reliable algorithm is appropriate. In other cases a faster algorithm is desired if it does not sacrifice much in quality. Such an algorithm is presented in the following paper.

# CHAPTER 2 - Summary of Related Work

Common approaches to object recognition are divided into an interest point detector and a feature descriptor. The purpose of the interest point detector is to reduce the complexity of the information in the picture. The feature descriptor takes the reduced input (the interest points) and characterizes them in a robust way, one that is resilient to natural transformations.

## Interest Point Detectors

Interest point detection reduces the complexity of visual recognition, by determining the salient points in an image. A "point" actually refers to a region of the image with a regular shape, centered on a specific point. This regular shape may circular (or elliptical, if affine invariance is desired), or it may also be rectangular (a cruder approach that is more straightforward for calculations). Analyzing a region surrounding a point is necessary because a single pixel does not have enough information to be deemed interesting or uninteresting.

Only the points found with the interest point detector will be passed to the feature descriptor. This does not require, however, that every pixel fall into at least one interest point's area. Certain regions of the image may not be useful in object recognition, and thus should be ignored. Ideally, portions of the image that are ignored by the interest point detector do not contain useful information for object recognition.

It would seem that if a large percentage of the image is covered by the detected interest points, then little has been accomplished. The argument would be that only a small percentage of the information has been eliminated. This is not the case. A good detector will properly aggregate the pixels of the image, so that they are easily analyzed in the feature description phase.

Numerous algorithms for interest point detection have been proposed. The earliest method that is still widely used today is the Harris corner detector. Harris corners are found using the eigenvalues of the second-moment matrix. They are rotationally invariant, but not scale-invariant. [8]

Scale invariance can be achieved with automatic scale selection. Lindeberg experimented with both the determinant of the Hessian matrix and the trace (the Laplacian) for use in scale selection. The maxima and minima of both functions were found to be useful in scale selection. [14]

Mikolajczyk and Schmid combined these techniques to create more robust detection schemes. The Harris-Laplace and Hessian-Laplace methods both use the Laplacian for scale selection, but use the Harris function and Hessian determinant (respectively) for point selection. [10]

Because the Laplacian is unstable, it is common to apply a Gaussian smoothing filter before applying the Laplacian filter. These filters can be combined into the Laplacian-of-Gaussian in order to reduce the number of computations necessary to perform both operations. [10]

Lowe approximated the Laplacian-of-Gaussian using a difference-of-Gaussian function. This function performs two consecutive smoothings using a Gaussian, and finds the difference in the resulting images. This is efficient to compute because the smoothing is already necessary for building multiple scales in the image pyramid. [11]

Bay, Tuytelaars, and Van Gool developed a detection scheme that is faster than the previously mentioned approaches. It relies on integral images [18] and box filters to approximate the determinant of the Hessian matrix. This has been demonstrated to be comparable with previous methods in terms of repeatability. [1]

According to Mikolajczyk, repeatability is the measure of the performance of a detector [19]. Repeatability can be tested with respect to standard viewing deformations such as lighting, viewing angle, rotation, and scale.

To understand repeatability explicitly, consider two different images of the same object or scene. Ideally, each interest point found in one image should correspond to an interest point in the other image. But, in order to compare these images, some mathematical relationship between them must be established. The ground truth is such a relationship. It is a homography that projects points to the reference frame [19]. Using the ground truth, it is possible to determine which points in one image correspond to a set of points in another image.

If we compute a ground truth transformation, we can map the coordinate space of one image to the other and compare the overlap of the two interest points. If the error in overlap (calculated as *1 – intersection/union*) is below .4, then these two points are said to be corresponding. The repeatability score, then, is the percentage of points in either image that have a corresponding point in the other image. [19]

From a survey of the literature we observe that many of the more robust detection methods have efficient, highly discrete approximations. Such approximations are sometimes comparable in repeatability. A high level of repeatability is desirable; however, since object recognition problems frequently have real-time constraints, there are applications where the speed of the detector may be equally important.

## Feature Descriptors

A feature descriptor attempts to characterize a region in a robust way that is invariant to natural viewing changes. This may include scale, rotation, lighting, and affine/viewpoint variance. While many such descriptors have been proposed, the SIFT [11] and SURF [1] descriptors have been shown to outperform many existing approaches. [1]

The SIFT descriptor consists of image gradient histograms. The image gradient at every point in the region is calculated. Then, the region is divided into 16 sub-regions (4x4). For each sub-region, the gradients are reduced to eight directions and combined to form a histogram. The resulting 128 values (8 directional values for 16 regions) are the SIFT descriptor. Also note that interpolation is used to reduce the "boundary" effect; thus, values in the center of a sub-region are weighted higher than values on the edge. This makes the descriptor robust to small deformations in varying viewpoints. [11]

The SURF descriptor acknowledges the high matching performance of SIFT and attempts to replicate this performance while improving speed. The region of interest is divided into a square grid of 4x4 sub-regions, like SIFT. However, instead of using a histogram of image gradients to characterize each sub-region, SURF measures the response of Haar wavelets summed over the sub-region. Both a "vertical" and "horizontal" Haar wavelet response are calculated. The actual directions are relative to the interest point's orientation. There are four values in the descriptor for each region: the sum of the response for each wavelet, and the sum of the absolute value of the response for each wavelet. [1]

While both of these descriptors were developed along with a particular interest point detector, they are commonly used with other interest point detectors. Their high performance makes them attractive candidates for combining with other algorithms.

# CHAPTER 3 - Homogeneous Color Detector

This detector is built for applications where speed is of the utmost importance. As such our characterization of an interest point must be very basic, yet retain some discriminative quality. The focus is on comparing the average intensity of a region with the average intensity of surrounding regions, and looking for discrepancies.

## Homogeneous Color

Two regions are defined to be homogeneous if the difference between the average light intensity (color) values of each region is within a threshold. In general, many of the pixels in one region will have nearly same color, so the average color will be representative of the region as a whole. For similar reasons, if two regions are adjacent, they may likely have similar colors.

If, on the other hand, two adjacent regions have significantly different average colors, this could be indicative of an edge. If a region is inhomogeneous with numerous adjacent regions, then there are edge responses in multiple directions. This is somewhat similar to the machine vision definition of a corner, such as in Harris' corner detector. It is also the basis for defining our interest points in this detector.

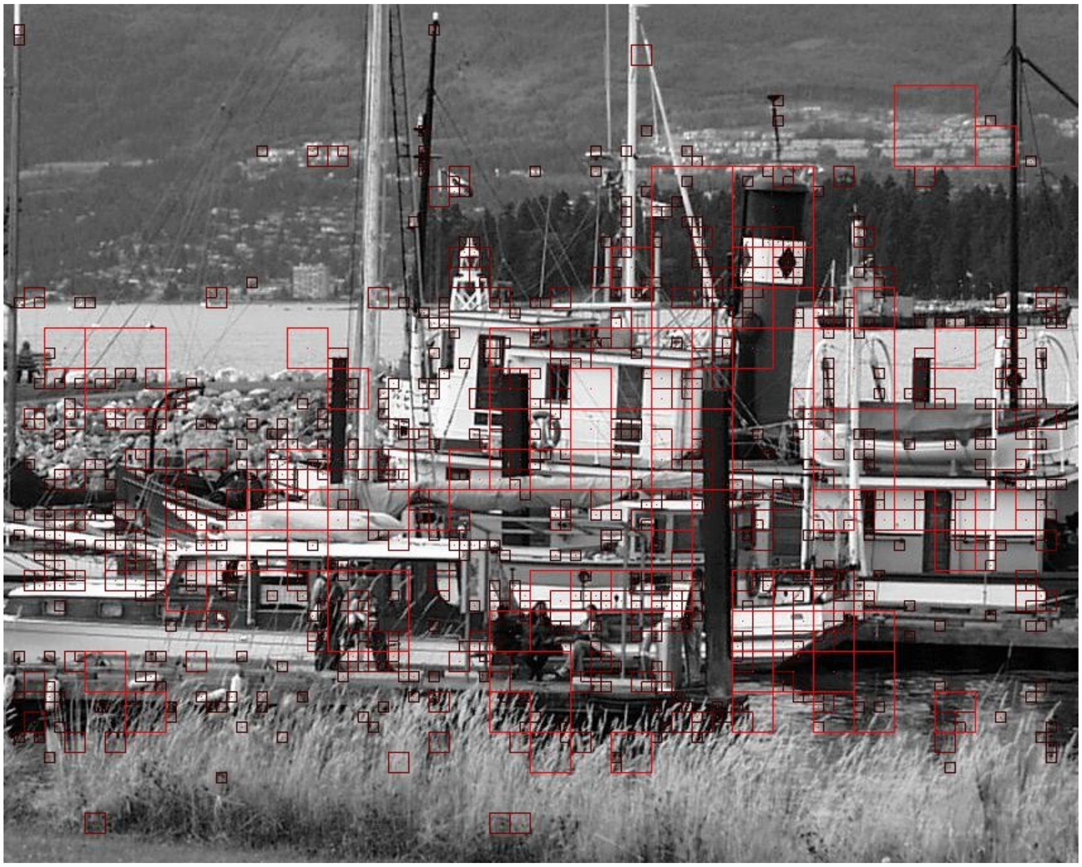

**Figure 1 - An Example Image With Detected Interest Regions**

This is an example image and the interest regions detected (test image from the Boat test set provided by Mikolajczyk). Boxes of differing sizes were found at differing scales during the selection process.

## Algorithm

To determine whether two regions are homogeneous, we must first define a threshold for homogeneity. The sample picture above was generated with a threshold of 20. To clarify, every region has an average intensity from 0 to 255, and two regions are homogeneous if their average intensities differ by less than 20. As is common in vision processing techniques, the optimal value of the threshold should be determined empirically for a given application.

For scale invariance, interest points must be detected at multiple scales in the image. Rather than use Gaussian smoothing to observe different scales, we simply reduce the size of the image by a factor of 2. Thus, at an arbitrary scale *n* (starting at 1 for the original image), one pixel represents a $2^{n-1}$ x $2^{n-1}$ region. This is significantly faster to compute than even Gaussian smoothing. Additionally, rather than using integral images to find the average intensity of a region, the regions of a given scale are represented by single pixel values.

Once the image has been scaled (this won't be necessary if we begin at scale = 1), a sweep over the image can be performed. For each candidate point, we observe a 3x3 grid of pixels centered on that particular point. Again, each pixel represents a $2^{n-1}$ x $2^{n-1}$ region (where $n$ is the scale). If the center pixel is homogeneous with 4 or more of the 8 surrounding pixels, then it is too similar to its surroundings and it is ignored.

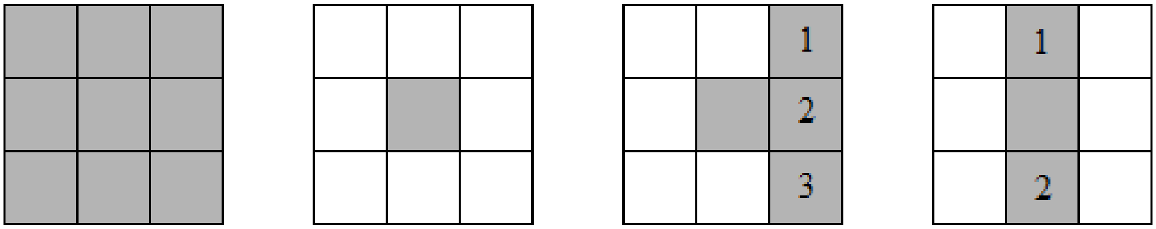

**Figure 2 - Interest Point Structure Examples**

If, however, the center pixel is homogeneous with 3 or fewer surrounding pixels, then it is probably an interest point. There is only one more check to reduce edge responses. If there are 2 or 3 homogeneous pixels, we require that they are adjacent to each other, either horizontally or vertically. This check is not applicable if there are 0 or 1 homogeneous pixels in the surroundings.

In the far left example, the point is rejected because the center pixel is homogeneous with all surrounding pixels. In the next example, the point is accepted because the center pixel is inhomogeneous with all surrounding pixels. In the third example, the point is accepted because the 3 inhomogeneous regions are adjacent (1 adjacent to 2, which is in turn adjacent to 3). In the final example, the point is rejected because 1 is not adjacent to 2. It is categorized as a potential edge and thrown out.

Because so few operations are necessary to confirm or reject an interest point, it is possible to check every pixel in a given scale. The algorithm does in fact check each point at a given scale, then reduces the image and repeats the process until the number of scales checked reaches the pre-determined  threshold.

## Theoretical Analysis of Running Time

This detector is fast compared with existing methods, because the key component of analysis (the average color of a region) is already found during the scaling process. This is similar to how SIFT's Difference-of-Gaussians is calculated using the values already necessary for Gaussian scaling. For each scaling, we require only 4 integer additions and 1 integer division per pixel in the new scale.

One of the significant improvements in speed comes from avoiding floating-point operations entirely. There are, in fact, only 8 integer subtractions (and use of absolute value) necessary to determine whether the surrounding pixels are homogeneous with the center. After these Booleans have been calculated, the remaining operations are simply Boolean checks.

Both the Difference-of-Gaussians and Fast-Hessian methods search for local maxima in order to localize an interest point. This step is omitted in the homogeneous color detector because candidate points are only checked at certain intervals. Points selected at a given scale cannot overlap.

# CHAPTER 4 - Performance Testing

As discussed in the literature review, repeatability is the primary measure of an interest point detector. Recall that repeatability is the percentage of correctly corresponding interest points found by a detector in two images of the same scene. A low repeatability score could indicate an inferior detector. However, a lower repeatability could also be justified in certain cases if it results in a significant increase in speed. This interest point detector has comparable repeatability to leading detectors in most cases, while offering noticeable improvements in the running time.

## Speed

Below is a comparison of the computation time of the Fast-Hessian detector versus the homogeneous color detector. It was performed on the first image in the Graffiti test set provided by Mikolajczyk.

| Detector | # points found | Time (ms) |
|---|---|---|
| Fast-Hessian | 1580 | 290 |
| Homogeneous Color | 3304 | 30 |

**Table 1 - Computation Time**

Clearly this interest point detector is significantly faster than the Fast-Hessian detector, which runs quicker than other current detectors. Furthermore, the homogeneous color detector was not fully optimized and was coded in Java as a proof-of-concept. The Fast-Hessian implementation tested was coded in C++. Minor improvements in the running time of the homogeneous color detector are likely possible with optimizations and with conversion to another programming language.

## Repeatability

The following figures represent repeatability testing of the homogeneous color detector, and direct comparison to the Fast-Hessian and Difference-of-Gaussian detectors used with SURF and SIFT (respectively). The testing software is provided by Mikolajczyk [19]. Note that the homogeneous color detector actually performs better than the other descriptors tested on the "Leuven" image set. Also note that for the following figures, the repeatability results for Difference-of-Gaussian and Fast-Hessian were found in [1].

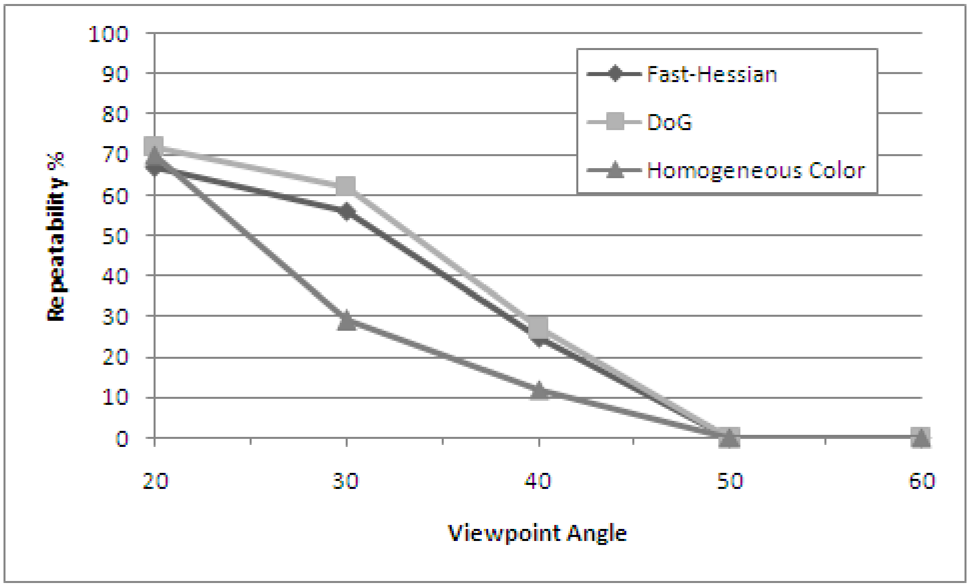

**Figure 3 - Repeatability Scores, Graffiti Test Set**

The "Graffiti" test set is the first of two sets to analyze the robustness of detectors with respect to viewing angle. The test set consists of six images with gradually changing viewing angles. Each value plotted represents the repeatability when comparing the first image against a subsequent image. For the fifth and sixth images, the affine transformation is too extreme for any of the detectors to correctly find corresponding interest points.

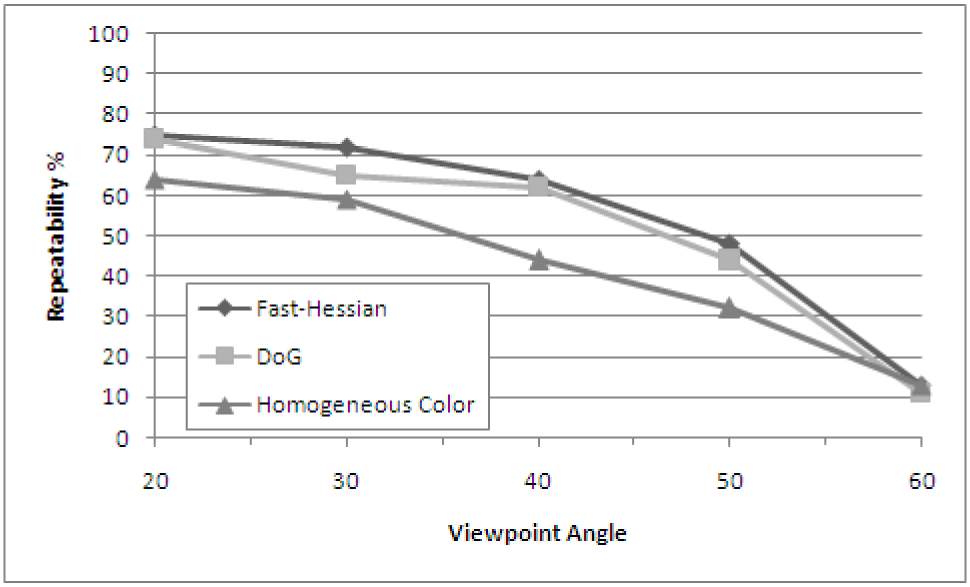

**Figure 4 - Repeatability Scores, Wall Test Set**

The “Wall” test set also analyzes the robustness of detectors with respect to viewing angle. All descriptors display a higher repeatability score in this test set than in the “Graffiti” set. This may be because the “Wall” set subjectively appears less interesting.

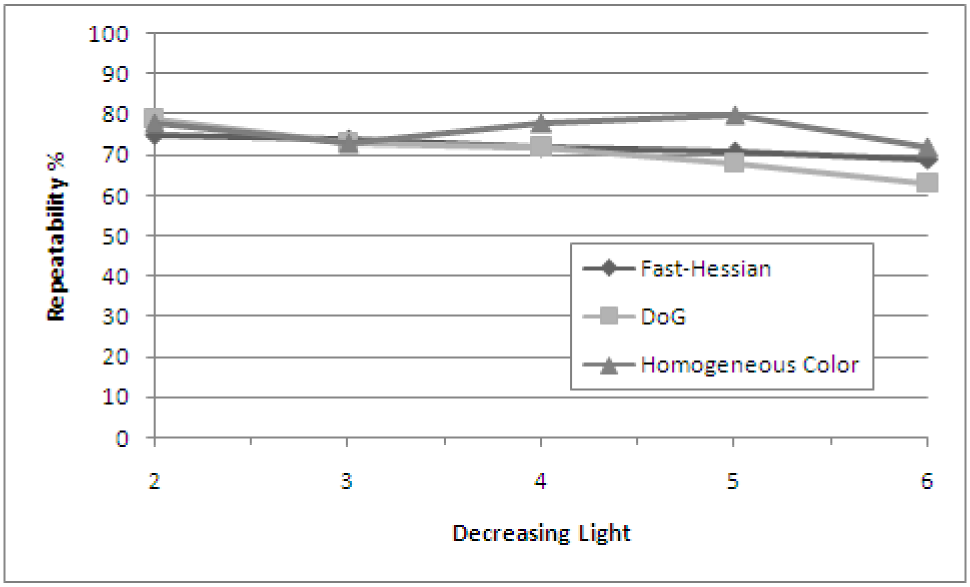

**Figure 5 - Repeatability Scores, Leuven Test Set**

The “Leuven” test set consists of pictures with gradually decreasing light. The homogeneous color detector displays higher repeatability on this test set than the Fast-Hessian and Difference-of-Gaussian detectors. This may be due to the type of scene depicted in the images, which places three cars prominently in the center. The sharp color contrasts of different features on the cars are easily recognized by the homogeneous color detector, even at low lighting.

# CHAPTER 5 - Conclusion

The interest point detector that has been presented is a suitable algorithm for time-critical applications. It is able to operate much faster than leading methods with only minor sacrifices in repeatability.

Future work should improve the robustness of the descriptor with respect to rotational invariance. Since the algorithm currently relies on axis-oriented regions, responses to rotations are sometimes favorable, but sometimes unpredictable.

In addition, research towards a complementary feature descriptor is also possible. Even the fastest current feature descriptors still require more computation time than this interest point detector. The total running time of an object detection algorithm that combines this detector with an existing descriptor will be dominated by the descriptor. An ideal complementary feature descriptor would run in approximately the same amount of time as the detector.

# Bibliography

1. Bay, H., Tuytelaars, T., and Van Gool, L. 2006. SURF: Speeded Up Robust Features. In Proceedings of the Ninth European Conference on Computer Vision.
2. Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. In International Journal of Computer Vision, 60(2): 91-110.
3. Mikolajczyk, K., and Schmid, C. 2002. An Affine Invariant Interest Point Detector. In: Proceedings of the Seventh European Conference on Computer Vision, 128-142.
4. Mikolajczyk, K., and Schmid, C. 2004. Scale & Affine Invariant Interest Point Detectors. In International Journal of Computer Vision, 60(1): 63-86.
5. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. 2005. A Comparison of Affine Region Detectors. In International Journal of Computer Vision, 65(1/2): 43-72.
6. Mikolajczyk, K., and Schmid, C. 2005. A Performance Evaluation of Local Descriptors. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10): 1615-1630.
7. Mikolajczyk, K., Leibe, B., and Schiele, B. 2005. Local Features for Object Class Recognition. In Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2, 1792-1799.
8. Harris, C., and Stephens, M. 1988. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, 147 – 151.
9. Lindeberg, T. 1998. Feature Detection With Automatic Scale Selection. In International Journal of Computer Vision, 30(2): 79-116.
10. Mikolajczyk, K., and Schmid, C. 2001. Indexing Based on Scale Invariant Interest Points. In Proceedings of the Eighth IEEE International Conference on Computer Vision - Volume 1, 525-531.
11. Lowe, D. G. 1999. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision - Volume 1, 1150-1157.
12. Kadir, T., and Brady, M. 2001. Scale, Saliency and Image Description. In International Journal of Computer Vision, 45(2): 83-105.
13. Jurie, F., and Schmid, C. 2004. Scale-Invariant Shape Features for Recognition of Object Categories. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, 90-96.
14. Lindeberg, T. 1998. Feature Detection with Automatic Scale Selection. In International Journal of Computer Vision, 30(2): 79-116.
15. Matas, J., Chum, O., Urban, M., and Pajdla, T. 2002. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In Proceedings of the Thirteenth British Machine Vision Conference, 384-393.
16. Tuytelaars, T., and Van Gool, L. 2004. Matching Widely Separated Views Based on Affine Invariant Regions . In International Journal of Computer Vision, 59(1): 61-85.
17. Kadir, T., Zisserman, A., and Brady, M. 2004. An Affine Invariant Salient Region Detector. In Proceedings of the Eighth European Conference on Computer Vision,

404-416.

18. Viola, P., and Jones, M. 2001. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, 511-518.
19. http://www.robots.ox.ac.uk/~vgg/research/affine/