2012

# A method to improve interest point detection and its GPU implementation

Prabakar Karuppannan Gunashekhar
*Louisiana State University and Agricultural and Mechanical College*, prabakarkg@gmail.com

# A METHOD TO IMPROVE INTEREST POINT DETECTION AND ITS GPU IMPLEMENTATION

A Thesis

Submitted to the Graduate Faculty of
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical and Computer Engineering

by
Prabakar Karuppannan Gunashekhar
B.E., CEG, Anna University, Chennai, India 2009
December 2012

## ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Dr.Gunturk for his invaluable support and guidance. This research would not have been successful without his help throughout my research. I would like to thank Dr. Li for his generous support and guidance in learning C++ programming. Without his help the GPU implementation of my algorithm might not have been possible. I would like to express my gratitude to Dr. Suresh Rai, for sharing his thoughts and experiences during the neural network course. I have learned to analyze and optimize the algorithms during his course and that experience was vital in my research. Also I am deeply grateful to all of them for being my committee members. I also want to thank Dr. Eurico D'Sa for financially supporting me for nearly two and half years. His support was very important for my degree.

Finally, I would like to thank my family and friends for their continuous support and appreciation through out my life. Without them nothing would have been possible.

# TABLE OF CONTENTS

# LIST OF FIGURES

# ABSTRACT

Interest point detection is an important low-level image processing technique with a wide range of applications. The point detectors have to be robust under affine, scale and photometric changes. There are many scale and affine invariant point detectors but they are not robust to high illumination changes. Many affine invariant interest point detectors and region descriptors, work on the points detected using scale invariant operators. Since the performance of those detectors depends on the performance of the scale invariant detectors, it is important that the scale invariant initial stage detectors should have good robustness. It is therefore important to design a detector that is very robust to illumination because illumination changes are the most common. In this research the illumination problem has been taken as the main focus and have developed a scale invariant detector that has good robustness to illumination changes.

In the paper [6] it has been proved that by using contrast stretching technique the performance of the Harris operator improved considerably for illumination variations. In this research the same contrast stretching function has been incorporated into two different scale invariant operators to make them illumination invariant. The performances of the algorithms are compared with the Harris-Laplace and Hessian-Laplace algorithms [15].

# CHAPTER 1 - INTRODUCTION

Computer vision has a wide area of applications ranging from simple smile detection in cameras to a highly complicated humanoid robot that can able to respond to its surroundings. Figure 1 shows the Asimo humanoid robot, designed and built by HONDA, receiving some customers [8]. It is still a challenge to develop an algorithm with very high performance under different noisy situations. It is clear that in future the application of computer vision is enormous and it might be a part of most of the embedded systems. The important initial stage of many computer vision algorithms is feature detection. The features can be regions, interest point, corners, edges etc.

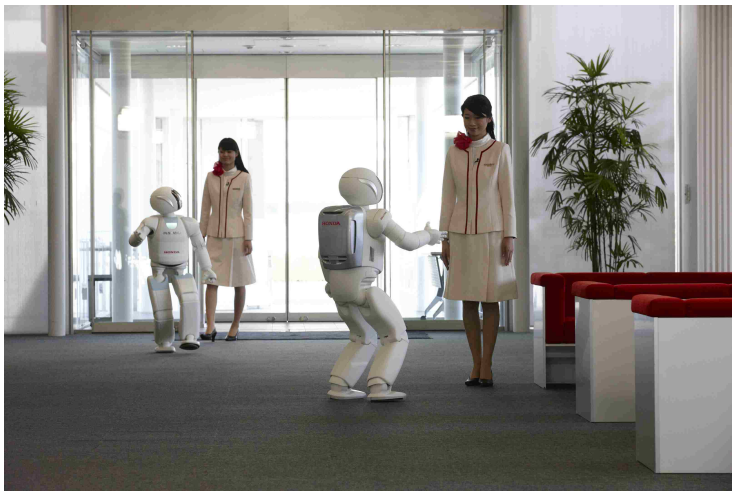


Figure 1: Image of ASIMO the robot, guiding customers [8].

Illumination is an important factor that affects the performance of the feature point detectors and feature descriptor algorithms. The images taken in our daily life will be highly dependent on the illumination conditions. Figure 2 shows an image taken during the night under high illumination conditions [3]. It can be observed that the objects and the people's face are difficult to identify. Under these conditions the algorithms will have low performance without any preprocessing done.

Figure 2: Effects of high illumination [3].

Interest point detection is a low-level computer vision technique that has wide range of applications like matching, object recognition, object tracking. In [15] an efficient affine invariant interest point detector has been introduced. The algorithm is a two-fold process in which the initial points are found using scale invariant operators to localize the points and then an iterative affine correction algorithm is applied. In the same paper two scale invariant operators namely Harris-Laplace and Hessian-Laplace were proposed. The points found by these scale invariant operators are further processed for affine corrections. For the algorithm to work well the initial points have to be located as close as possible to the actual location and the best approximation of the scale should be found. Thus the whole performance of the affine correction procedure depends on the initial position given by the scale invariant detectors. Many of the affine invariant interest point detectors and region descriptors are dependent on the performance of the initial interest point operators. It is therefore crucial that the initial points lie almost close to the actual points for the further procedures to give better results. Under strong illumination conditions the error in the initial position of the points will be high. In [6] an illumination invariant detector was proposed which is based on contrast stretching technique. The operator is

named as Illumination Robust Feature Extraction Transform (IRFET). The input image is contrast stretched around several contrasts and then the Harris detector [7] is applied on each of the contrast stretched images. The output measure of the detector from each of the contrast stretched image is summed up to give the final measure that is used to find the interest points. The idea behind this algorithm is that by contrast stretching the images, some of the image structures that might not be of good contrast in the original image might become easy to detect in the contrast stretched image, for example the low intensity areas might become visible when the image is stretched around low contrast values. It has been proved that the performance of the Harris detector increases considerably under high illumination changes, by using the contrast stretching technique.

In this research the multi-scale Harris detector [6] and the Hessian detector [16] has been incorporated with the illumination invariant IRFET technique to develop methods that are invariant to scale and illumination changes. In the proposed methods the multi-scale Harris or the Hessian operator is applied on all the contrast stretched images and finally the interest points are chosen from the summed up responses from all the contrast stretched images. The performance of the proposed method has been validated against Harris-Laplace and Hessian-Laplace that were proved to have good robustness to scale changes. The following report is organized as follows: In chapter 2 the literature review of the interest point detectors are discussed briefly with some examples. Then the interest point detectors used in the research are detailed in chapter 3. In chapter 4 the proposed methods and the algorithm to detect the points are discussed in detail. In chapter 5 the results have been discussed using datasets with different variations. Also the scale validity is proved using SIFT features and the research is concluded with possible future improvements in chapter 5..

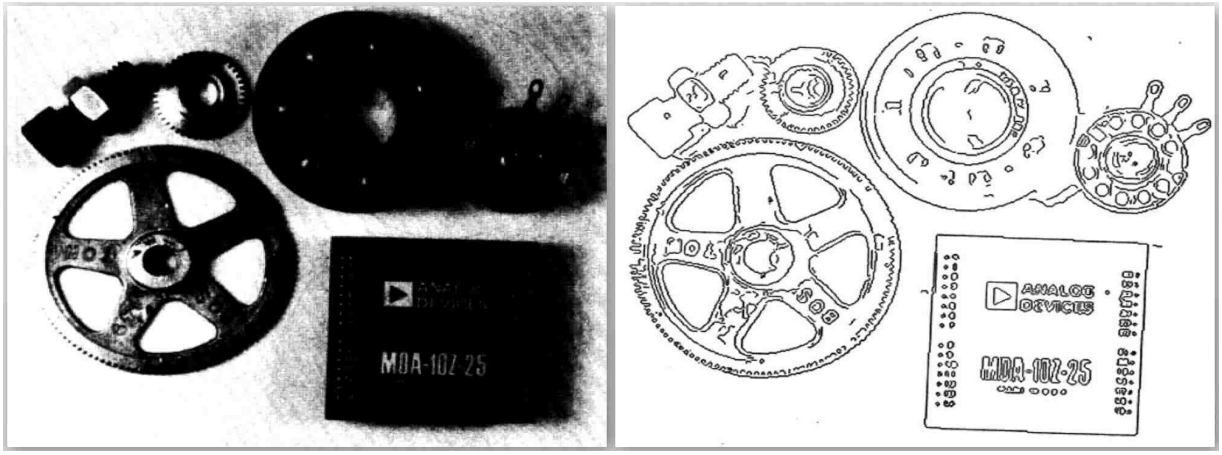
# CHAPTER 2 – LITERATURE REVIEW

## 2.1 Interest Point Detection

Interest points are intensity variations in two or more directions and the point represents some structure at one or more scales. Such points are robust and can be used to produce better retrieval performance than the other features like edges and blobs. There are many algorithms to extract such points. They can be broadly classified as variant and invariant detectors.

## 2.2 Variant Detectors

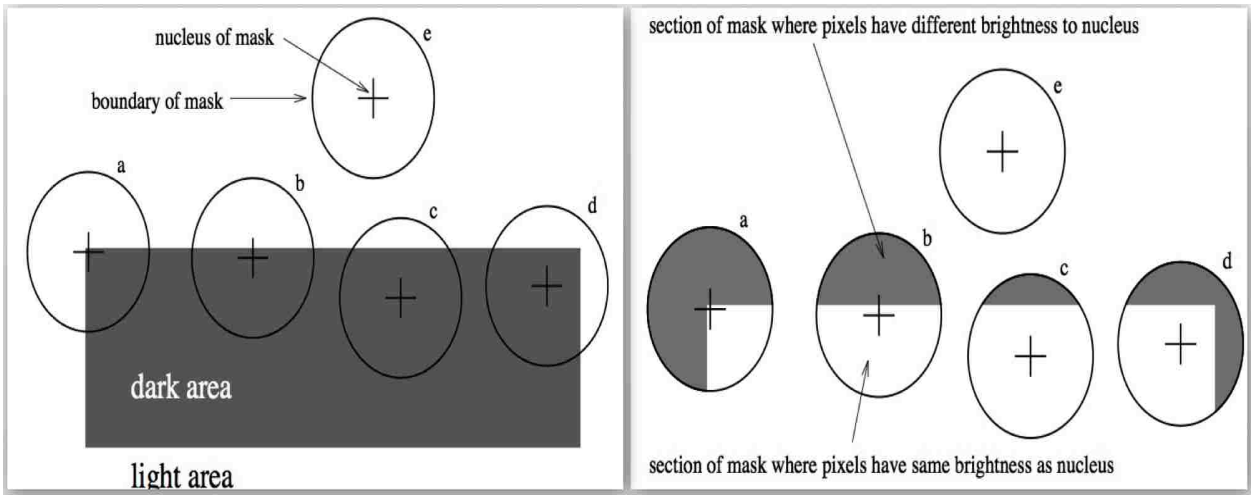
These detectors depend on the scale and affine transformations of the images. One such detector is the canny edge detector [4], which is a multi-stage operator. It uses the edge gradient's magnitude and the corresponding direction calculated using four different filters that can detect the horizontal, vertical and the diagonal edges, to trace the edges. As the algorithm is based on the first order derivatives it is highly susceptible to noise and it uses Gaussian filtering on the image as a preprocessing stage to reduce noise. Figure 3 (a) shows the reference image and (b) shows the edge detectors output [4].




(a) (b)

Figure 3: a - Reference image, b - Canny operator output [4].

Another widely used feature detector was Smallest Univalue Segment Assimilating Nucleus (SUSAN) [20]. It uses a circular mask and calculates a feature measure which is proportional to the number of pixels in the neighborhood of a pixel within the circular mask that are similar to the center pixel (nucleus). If that number is around one quarter of the total number of pixels that the circular mask contains, then the pixel is a corner or if the number of pixels is somewhere near half the number of pixels contained in the mask, then the center pixel is on an edge. Figure 4 below provides an insight of how this algorithm works [20].




(a) (b)

Figure 4: SUSAN operator [20].

Figure 4 shows that if the position of the center of the mask is at the corner the number of pixels similar to the nucleus will be as low as possible. If it is on the edge, the number of pixels similar to the nucleus will be almost half of the total number of pixels covered by the mask. The performance relies on the size of the mask and under affine variations it fails to perform well. Also the edge features detected by canny detector are not reliable for retrieval because they are highly affected by the affine variations making the retrieval process more difficult.

The most widely used operator and that forms the basis of many other operators is the Harris operator [7]. Harris operator detects both edges and corners and it uses the second

moment matrix to calculate the feature strength. The Harris detector was formulated by removing the problems associated with the Moravec's detector [19] given by $E_{x,y} = \sum_{u,v} w_{u,v} \left| I_{x+u,y+v} - I_{x,y} \right|^2$. In the operator $w$ is the window which has a value 1 inside the window and 0 elsewhere, $I$ is the intensity of the image. $(x,y)$ determines the shift and its values are from the set {(1,0),(1,1),0,1),(-1,1)}. $(u,v)$ is the position of the current pixel. The final response of the operator at each pixel is the minimum value of $E_{x,y}$ among the four shifts. The response will be low if the region is flat and it will be high for corners. A point is considered as corner if it is a local maximum of minimum of $E_{x,y}$ and it should be above some threshold. The problems with the Moravec's detector is that the detector's response is anisotropic because of the fixed shifts used and it is affected by noise because of the binary and rectangular window and also it might respond to edges readily because only the minimum of the operator $E_{x,y}$ is considered as a measure [8].

The problems associated with Moravec's operator were eliminated in the Harris detector by using the Taylor approximation of the operator. The approximation of the Moravec operator is given by $E_{x,y} = Ax^2 + 2Cxy + By^2$, where $E_{x,y}$ is the response at pixel $(x,y)$. $A$, $B$ and $C$ are the parameters of the second moment matrix given by $\mu(\mathbf{x}) = \begin{bmatrix} L_x^2 & L_x L_y \\ L_x L_y & L_y^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$, where $L_x$ is the 1st order image derivative in x direction, $L_y$ is the 1st order image derivative in y direction and $\mu(\boldsymbol{x})$ is the second moment matrix at pixel $\boldsymbol{x}=(x,y)$. By examining the eigenvalues of $\mu(\boldsymbol{x})$ the interest points can be detected. The cornerness measure for the Harris operator is given by $det(\mu(\mathbf{x})) - k\, tr(\mu(\mathbf{x}))$, where $det(.)$ is the determinant of the Harris matrix, $tr(.)$ is the trace of the matrix and the constant $k$ is assigned a value of 0.4. The measure will be high for a corner,

low for a flat region and negative at an edge. In this research, scale adapted Harris operator is used as one of the primary operator.

In [22] color transformations to invariant color spaces where applied to improve the performance and such transformations were applied on the Harris-Laplace and SUSAN to prove that the color is important for interest point detectors. But under high illumination conditions the detector's performance drops low. Also the detector adds complexity to the algorithm in the means of transformations.

### 2.2.1 Invariant Detectors

There are number of algorithms that are robust to scale changes. The widely used algorithm is the Harris detector that detects edges and corners. Since the detector uses a symmetric Gaussian kernel in finding the derivatives and also for smoothing, the detection algorithm is invariant to rotation and also to slight illumination changes [17]. The detector is not scale invariant by itself because it uses one scale to find the points. In [5] an approach to make the Harris corner detector invariant to scale was proposed. The Harris operator was adapted to scale by adapting the second moment matrix to scale. The matrix is given by $\mu(\boldsymbol{x}, s\sigma_D, s\sigma_I) = \sigma_D^2 G(s\sigma_I) * \begin{bmatrix} I_u^2(\boldsymbol{x}, s\sigma_D) & I_{u,v}(\boldsymbol{x}, s\sigma_D) \\ I_{u,v}(\boldsymbol{x}, s\sigma_D) & I_v^2(\boldsymbol{x}, s\sigma_D) \end{bmatrix}$ where $\sigma_D$ is the derivative scale, $\sigma_I$ is the integration scale $I_u$ is the image derivative in $u$ direction, $I_v$ is the image derivative in $v$ direction, $s$ is the parameter used to create the scale space, $G$ is the Gaussian integration kernel and $\boldsymbol{x}$ is the position vector $(x,y)$. The matrix is multiplied by a factor $\sigma_D^2$ to make the measure comparable across scales. The detector has low performance under affine and high illumination changes, because of the symmetric Gaussian kernel and limited number of scales used. In this research this operator

is used to adapt to high illumination changes and it has been the important part of the proposed detector.

In [10] a scale invariant detector was proposed in which the Laplacian of Gaussian (LoG) filter was used as the detection operator. Since the LoG operator is based on the second order derivatives they are highly sensitive to noise and have direct influence on performance of the detector. The solution for this problem was given by [18] and a method similar to Harris detector was proposed by detecting the points for which the determinant and the trace of the Hessian matrix given by $\mathbf{H}(\mathbf{x}) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma_D) & L_{xy}(\mathbf{x}, \sigma_D) \\ L_{xy}(\mathbf{x}, \sigma_D) & L_{yy}(\mathbf{x}, \sigma_D) \end{bmatrix}$ attains an extrema. In the matrix $L_{xx}$ is the 2nd order image derivative in $x$ direction and $L_{yy}$ is the 2nd order image derivative in $y$ direction, $\sigma_D$ is the scale of the derivative kernel and $\mathbf{x}$ is the position vector $(x,y)$. The derivatives are performed on the image and then smoothed by the Gaussian integration kernel with scale $\sigma_I$. The scale for the derivative kernel has to be carefully scaled respective to the current integration scale by a factor. One interesting part of the operator is that the trace of the matrix gives the LoG response. By simultaneously detecting the determinant and trace extremum, the detector is less affected by noise [18]. In this research the Hessian matrix is also an important part in developing the illumination invariant operator.

In [15] an affine invariant interest point detection algorithm was proposed. In this two stage algorithm, the first stage is to detect interest points and their corresponding scale in a scale invariant way. The detected points are then iteratively corrected for affine variance. The initial scale invariant operators are the Harris-Laplace and Hessian-Laplace operators. These scale invariant stages in turn has two stages. In the first stage a set of points are detected using multi-scale Harris operator or multi-scale Hessian operator and then the detected points are checked for

weakness using LoG operator. Though the operators are invariant to scale and rotation they fail to perform well under high illumination changes. The affine invariant iterative procedure that depends on the points and scale detected by the initial scale invariant point detector is in turn not invariant to high illumination changes. Figure 5 shows the affine regions detected using the detector [15].

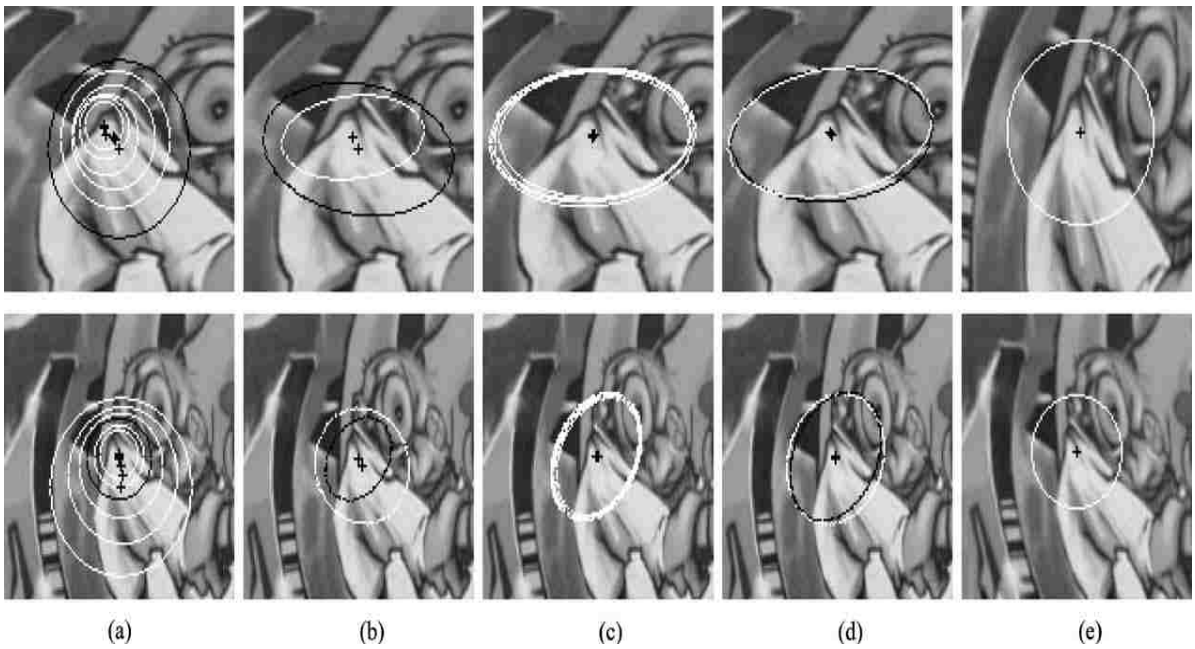



Figure 5: Iterative affine corrections for interest points [15].

In figure 5, the top and bottom part of the images are the regions taken from two different images but corresponding to the same local structure. Image (a) shows the initial points detected using multi-scale Harris detector with their scales in white. The point selected using the Harris-Laplace is shown in black. It is clear that the multi-scale Harris operator finds multiple points corresponding to the same structure and the LoG stage of Harris-Laplace removes those duplicate points. Image (b) shows the affine region for the initial point detected by Harris-Laplace before the affine corrections in black. The actual affine region from the corresponding image is shown in white. Image (c) shows that all the initial points detected by the multi-scale

Harris operator converge to the same point and region. Image (d) shows the average affine region selected in black and the actual region projected from the corresponding image in white. Image (e) shows the region after the affine corrections.

There are also algorithms to describe regions of interests so that the descriptions can be used to match the objects. One such algorithm is Scale Invariant Feature Transform (SIFT) [12]. In this method a Differentiation of Gaussian (DoG) operator that is an approximation of LoG operator is used to find the initial points and the scale. Then a set of features is determined for each point using gradient's magnitude and orientation of pixels around the interest point in an affine and illumination invariant way. In [2] a very efficient algorithm was proposed to find region descriptions called as Speeded Up Robust Features (SURF). The algorithm makes use of *integral images* to efficiently calculate the approximation of Hessian measure to find the initial interest points. The features are based on the response of Haar wavelet response. The algorithm is several times faster than the SIFT and is robust to many affine transformations. Again in these algorithms the initial points is found by some scale invariant operators that are not invariant to illumination.

All the detectors that are scale invariant fail to perform well under high illumination changes. The reason for those detectors to have poor performance is that they don't have any specific operations in them to improve their performance under illumination changes. This problem is taken as the primary consideration and it is the motivation for this research, because of the necessity for robust scale and illumination invariant detectors. The scale invariant operators have been adapted to illumination invariance and a method has been developed to improve their performance.

# CHAPTER 3 – INTEREST POINT DETECTORS

## 3.1 Harris-Laplace

### 3.1.1 Multi-Scale Harris

The Harris Laplace consists of two stages. In the first stage the interest points are detected using the multi-scale Harris operator and then in the second stage, the selected points are further refined using the LoG measure. For an input image a scale-space is created and the corner strength at different scales is measured. In this research the initial scale used to test the performance of Harris-Laplace operator is 1.5 and a scale step size of 1.4 is used to get the scale for the higher levels. To make the search space for the derivative and integration scales simple, the derivative scale is chosen as a factor of the integration scale. In this research a factor of 0.7 is used to get the derivative scale from the integration scale (i.e. $\sigma_D = 0.7\sigma_I$). Figure 6 shows a good picture of how the scale space is created [2]. The image shows that the derivative of the input image is successively determined and smoothed with kernels of increasing size to create the scale-space.

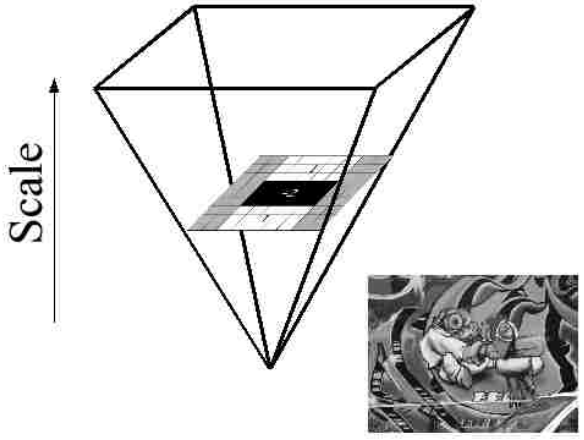



Figure 6: Scale space creation. Left image – integration kernel of varying size, right image input image [2].

After the scale space is created the interest points are detected. A point is considered as an interest point if it is a local maximum and has corner strength greater than a threshold to ensure that the points are strong. The search for interest points is done at every level of the scale space and the corresponding characteristic scale is assigned to the point. The scale selected for an image structure is characteristic in quantitative sense, since the measure relates the derivative operator and the image structure. Note there might be more than one characteristic scale corresponding to the same point but corresponding to different structures [15]. The points found by the multi scale Harris detector is then further processed based on the LoG measure to reject points that might be weak. Also the point location and scale are refined.

### 3.1.2 Laplacian of Gaussian (LoG)

The input to this stage is the points found by the multi-scale Harris operator. It has been found that the Harris measure rarely attains maximum over scales and also LoG has been proved to select highest percentage of correct characteristic scales [15]. Figure 7 shows the plot of a 2D LoG filter with scale 9 given by $\left|\mathrm{LoG}(\mathbf{x},\sigma_{n})\right| = \sigma_n^2\left|L_{xx}(\mathbf{x},\sigma_n) + L_{yy}(\mathbf{x},\sigma_n)\right|$, where $\sigma_n$ is the scale of the kernel, $L_{xx}$ is the 2nd order image derivative in $x$ direction and $L_{yy}$ is the 2nd order image derivative in $y$ direction at point $\boldsymbol{x}=(x,y)$.

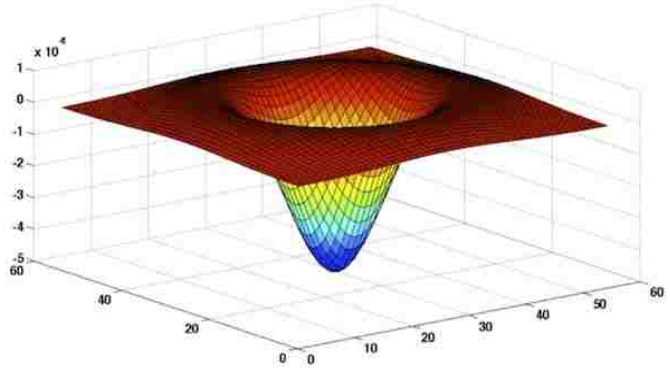



Figure 7: 2D LoG Kernel.

The LoG measure is perfectly suitable for detecting blobs because of its perfect symmetry but it also has been proved to be a good measure to detect interest points such as corners, edges, multi-junctions etc. [15].

Two different methods have been proposed to eliminate weak points in the LoG stage. One is the iterative method that simultaneously refines the position and the scale of the detected interest point. The other method is the simplified one that rejects or retains the point but does not modify the scale or the position. The iterative method is accurate but the simplified method is computationally efficient. The simplified method is a tradeoff between the accuracy and execution time. However in our experiment the LoG stage is not used because of the use of contrast stretching technique to reject weak points.

**Simplified Method:**

In the simplified method, the LoG measure for the characteristic scale is checked for its maximum with the nearest finer $\sigma_{n-1}$ and the coarser scale $\sigma_{n+1}$, where $n$ is the current scale-space level. If the response does not attain a local maximum in the selected scale or if it is less than the nearest scales or if it is lower than a threshold the point is rejected.

**Iterative Method:**

In the iterative method the selected scale from the multi-scale Harris stage is modified by a factor $c\sigma$, where $\sigma$ is the scale corresponding to the point from the multi scale Harris detector. The factor $c$ ranges from 0.7 to 1.4. The interval (0.7 to 1.4) spans the space between the successive coarser and the finer scale. At each stage the position of the point is refined along with its scale. When there is no change in scale and the position then the algorithm stops and the current position and scale are updated for the corresponding interest point. Thus the scale and the

position of the initial interest point are iteratively modified until the scale and the position does not change anymore. Below is the algorithm briefing the different steps to calculate LoG operator's response [15].

**Algorithm**

- If the point $\mathbf{x}^k$ attains a maximum over LoG scales the point is retained otherwise rejected. The scales are limited to $\sigma^{k+1}=c\sigma^k$, where $c$ is between 0.7 and 1.4
- Neighboring pixel with high Harris measure than the current pixel is searched for the current scale $\sigma^{k+1}$
- Go to step 1 if $\sigma^{k+1}\neq\sigma^k$ or $x^{k+1}\neq x^k$
- The selected points from the LoG detector are then further processed for affine correction. Since the affine correction is out of scope for this research it is not discussed in this documentation.

Figure 8 shows the output of the multi-scale Harris detector and the scale selected by LoG operator. Top row of the figure shows points selected using multi-scale Harris operator and the bottom row shows points selected using Harris-Laplace operator.

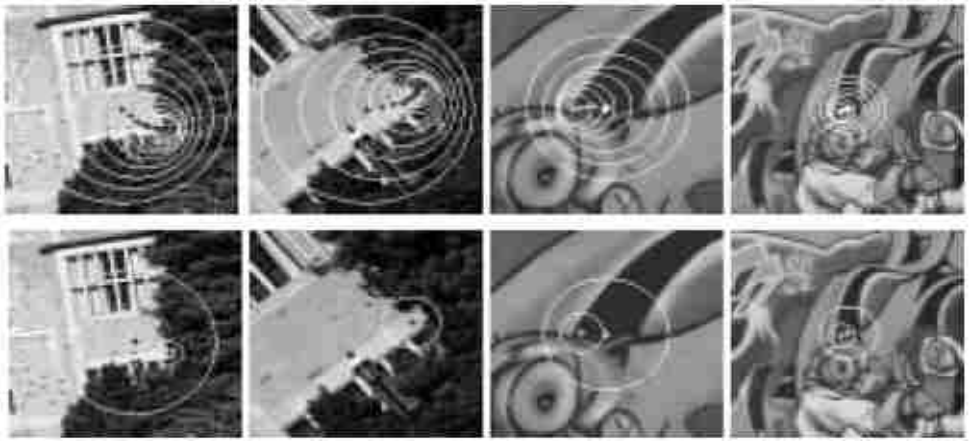


Figure 8: Operator responses – multi-scale Harris (top), Harris-Laplace (bottom) [15].

### 3.2 Hessian-Laplace

The Hessian-Laplace algorithm is similar to Harris-Laplace except for the second moment matrix. Hessian-Laplace uses a different measure based on second derivatives of the image as discussed in section 2. The operator uses the determinant of the Hessian matrix given by $\det(\boldsymbol{H}) = \sigma_I^2 \left(L_{xx}L_{yy}(\mathbf{x}) - L_{xy}^2(\mathbf{x}, \sigma_D)\right)$ and the trace of the Hessian matrix given by $tr(\boldsymbol{H}) = \sigma_I\left(L_{xx} + L_{yy}\right)$. The determinant and the trace are multiplied by a factor of $\sigma_I^2$ and $\sigma_I$ respectively to make the measures comparable between scales. A point is considered as an interest point if the determinant and the trace of the second moment matrix attains maximum. Since it uses the second derivatives, the points may have changes in only one direction. By checking the maximum this way the operator penalizes points that have changes in a single direction [17].

The procedure is the same as described for Harris-Laplace. A scale space is created and a set of interest points is detected at each level by selecting points that attains local maximum in determinant as well as trace and if the corner strength is above some threshold. The selected points are further checked for weakness by using simplified LoG or iterative LoG operators.

Figure 9 shows the multi-scale hessian operators response to an image for different scales. The first image (top-left) is the reference image (Note: The operators work on the gray scale image. All the color images are converted to gray scale and then the operators are applied) and the following 12 images are the responses of the Hessian operator for different scales. The 2nd last image is the multi-scale response (i.e. the scale-space maximum of each pixel) of the Hessian operator. The final image is the output of the non-maxima suppression applied on the multi-scale response.

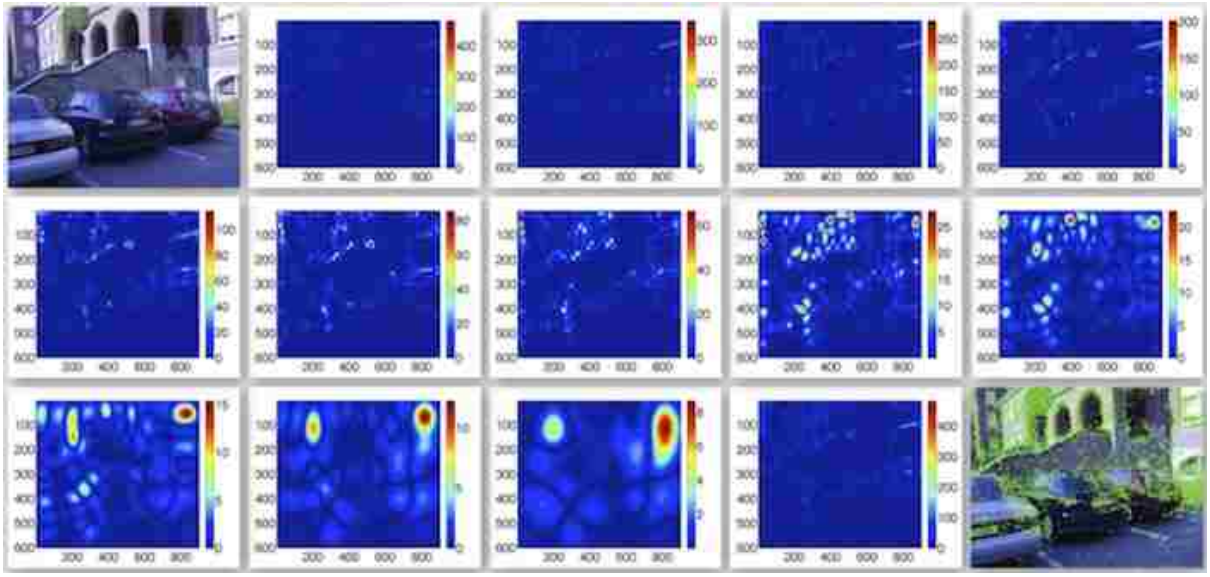


Figure 9: Overview of multi-scale Hessian operator.

Figure 10 shows the LoG responses to the input reference image shown in the first image of the figure. The 12 images following the reference correspond to the LoG response at 12 different scales. The 2nd last image is the interest points detected by the multi-scale Hessian and the last image is the output of the LoG operator. We can see that the number of points in the last image is less because the LoG operator rejects the weaker points.

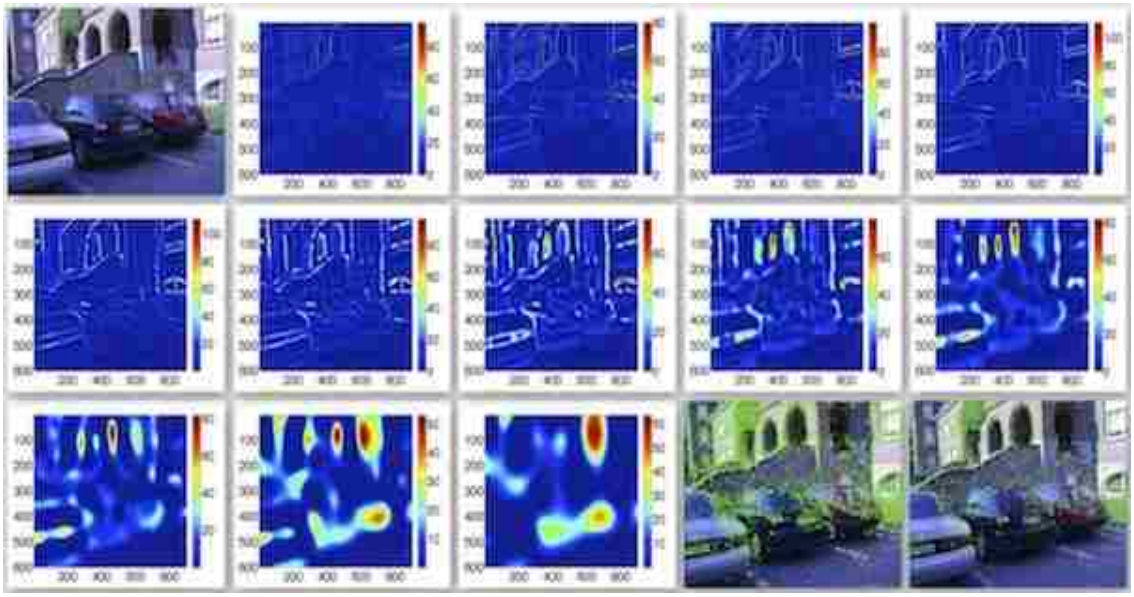


Figure 10: Overview of the LoG stage of Hessian-Laplace.

### 3.3 Illumination Robust Feature Extraction Transform (IRFET)

The IRFET [6] accounts for illumination problem by using the contrast stretching technique by stretching the input image around certain contrasts and applying the detection process on those images. This enhances the possibility of detection of interest points that might be under low-contrast regions, which might not be detected in the actual image. The contrast stretching function used is given by $f_c(\mathrm{I}(\mathrm{x},\mathrm{y})) = {}^{1}/_{1+e^{-\gamma(I(x,y)-c)}}$. In this formula, $c$ is the contrast around which the stretching is done, the parameter $\gamma$ (gamma) controls the slope of the function i.e. the range of values around the contrast that are stretched, *I(x,y)* is the normalized intensity value of the pixel at location (*x,y*). The function with different gamma values is shown in the figure 11 [6]. In the paper 50 different contrasts and a gamma value of 25 has been used.

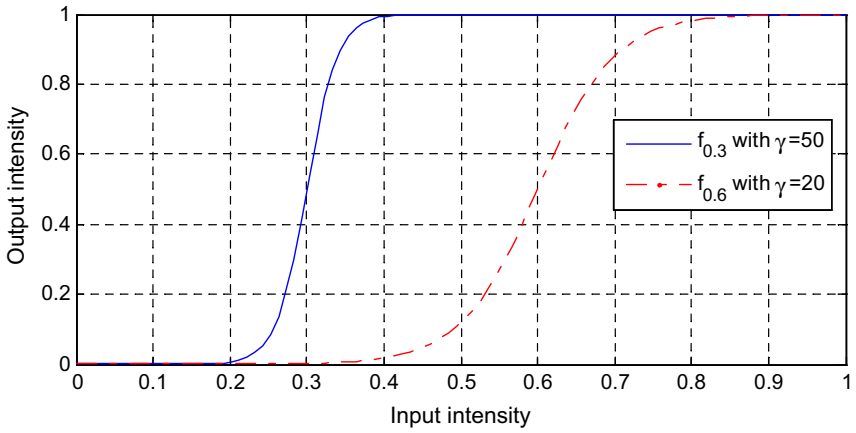



Figure 11: Contrast stretching function [6].

The idea behind the IRFET operator is that the points having high variability will be detected over many contrast stretched images. The input image is contrast stretched around some preselected contrasts. Then any of the interest point detection operators are applied on the contrast stretched image to obtain the scores. We will call this as *contrast-space* throughout this documentation. The point is selected as an interest point if it was detected across many of the

contrast stretched images. One way to detect points is to select interest points in each of the contrast stretched image by applying non-maxima suppression with thresholding and then to check for points that were detected in many levels of the contrast-space. But that is a costly operation. Instead if we sum up the scores across the contrast-space and apply the non-maxima suppression with thresholding on the summed up measure the output will be the same. By doing this way the computation has been reduced a lot. The latter method of measure calculation has been used in this research. Figure 12 below shows how a point would respond for the interest point operator across the contrast-space [6].

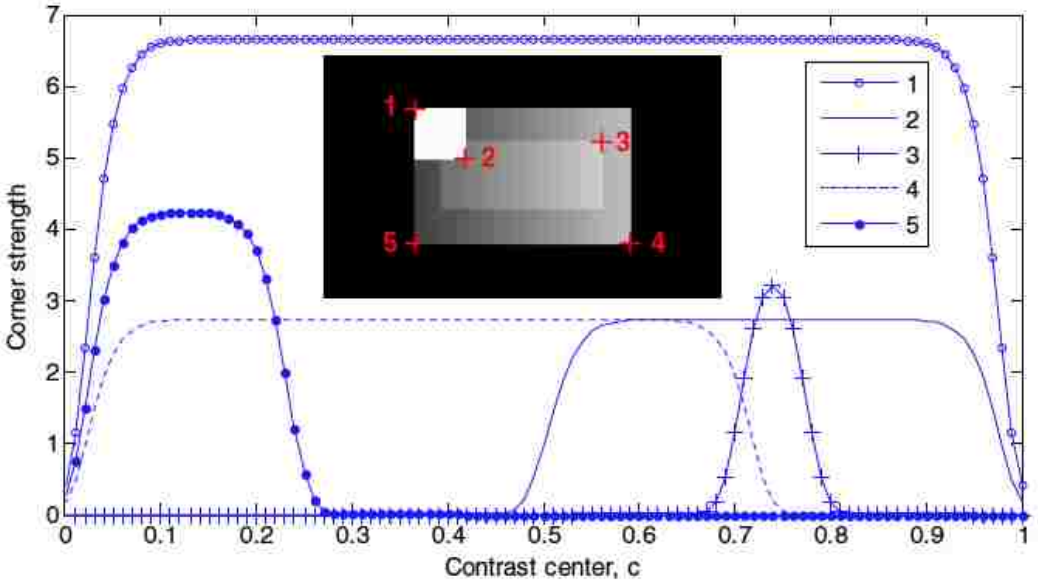



Figure 12: Harris operator's response on contrast stretched images.

This idea of contrast stretching can be incorporated into any of the interest point detection algorithms to make it illumination invariant. It has been shown that this method has improved the performance of Harris detector considerably under high illumination changes. The performance is validated using the point repeatability score. A drawback of the algorithm is that the detection algorithm has to be applied on each of the contrast stretched image. Thus the detection time is increased and it depends on number of contrast stretched images used.

# CHAPTER 4 – PROPOSED METHODS

## 4.1 Operators

The Harris-Laplace and Hessian-Laplace are robust to slight illumination changes. But under high illumination variations the detector's performance decreases, which is evident from the results discussed later in this report. On the other hand, IRFET algorithm has been shown to be robust under high illumination variation. In this research it has been proved that the proposed methods are better in point localization than the Harris-Laplace and Hessian-Laplace and so they can be used as a preprocessing stage for affine invariant detectors [15] that are based on the output of the scale invariant detector or can be used as a preprocessing stage for region descriptors that work on the points found by some interest point detectors. Thus the whole interest point detector or the region descriptor algorithm will be robust to high illumination changes. The proposed methods are discussed below.

The IRFET contrast stretching technique can be incorporated into any algorithm to make it illumination invariant. One way to make the affine invariant algorithm [16] robust to illumination is to apply the whole algorithm on contrast stretched images. But there is an important factor to be considered in this method. The affine detection algorithm detects the elliptical affine region using the second moment matrix. Since interest points are detected and affine regions are calculated in all the contrast stretched images, the problem arises in choosing the elliptical region for a point. One possible solution for this is to select the region for which the Harris measure is a maximum across the contrast space (2D space created by stretching the images around some contrasts). But this method will be highly prone to noise and also the scale of a region corresponding to a particular structure might be different in different contrast stretched images. A better way to implement is to make the preprocessing stage of the affine

detection algorithm robust to illumination. Thus to replace the Harris-Laplace or the Hessian-Laplace with the proposed algorithm, became a better solution. The multi scale Harris and the Hessian detectors are incorporated with the IRFET to make the detection, scale and illumination invariant.

The proposed algorithms are named as Harris-IRFET that uses the multi-scale Harris operator and Hessian-IRFET that uses multi-scale Hessian.

### 4.1.1 Stage 1 – Creating Contrast-Space

Let us assume that the number of contrasts used is $n$ (i.e. n equally spaced contrasts between 0 and 1), $\gamma$ be the slope of the stretching function, $I$ be the input image. First step in the proposed methods is to preselect a set of contrasts around which the input image can be stretched to form the contrast-space. The number of contrasts used has a direct impact on the detectors execution time. Therefore the number of contrasts used has to be carefully selected. After carefully studying the responses of the operators for different $\gamma$ and $n$ it has been found that the minimum value for n is 9 (i.e. 9 equally spaced contrasts between 0 and 1) for Hessian-IRFET and 12 for Harris-IRFET. The value for $\gamma$ is set as 30 for both the operators. After fixing the parameters of the contrast stretching function the next step is to apply the stretching function to the input image $I$. A contrast-space is created by contrast stretching the image around different contrast using the contrast stretching function.

Figure 13 shows the reference image (top-left) stretched around contrasts 0.2, 0.5 and 0.8 with a gamma value of 30 respectively shown by the following images. In the image (top-right) the stretching has made the number tag of the cars more visible than they were in the original image and improved the contrast in certain regions of the image. This property of contrast stretching technique makes the operator illumination invariant.

Figure 13: Contrast space (contrasts used – 0.2, 0.5, 0.8).

### 4.1.2 Stage 2 – Multi-Scale Operation

On each of the images in the contrast space the multi-scale Harris or the multi-scale Hessian operator is applied to get the corner strength of each pixel in the image. In using the multi-scale Harris operator the response is calculated similar to the one discussed earlier in the section 3.1 for each of the contrast stretched images and they are summed up to give the final measure.

In using the Hessian operator a slightly different approach was used. In [2] it has been experimented and found that the trace of the Hessian matrix did not have considerable effect on the performance and so they used the determinant in building their operator. Also our experiments comply with the latter result and in this research the idea of using the determinant was adapted and instead of using the determinant and trace simultaneously, only the determinant

of the Hessian matrix is considered to get the response score. By incorporating the operator by this way the operator had better or comparable performance with the Hessian-Laplace.

For the multi-scale detectors a range for the integration $(\sigma_I)$ and derivative $(\sigma_D)$ scales has to be chosen. In our research the smallest scale is chosen as 1.5 and the scale is increased in the consecutive levels with a step size of 1.4 ($\sigma_I = 1.5(1.4\text{^}level)$, where the initial level is considered as $0^{th}$ level and the symbol ^ denotes the power operator). The number of levels is chosen as 11 and these are used to determine the scale of a structure centered at a particular pixel. The range of scales chosen entirely depends upon the application where the operators are used. In using the detectors to detect objects that are very big in very high-resolution images the longest scale should be sufficiently large in turn the step can be set to some high value. It is impossible to use all combination of derivative and integration due to limited hardware resources. The derivative scale is chosen as a fraction of the integration scale similar to multi-scale Harris. Figure 24 shows the Gaussian integration and derivative kernels at a scale.

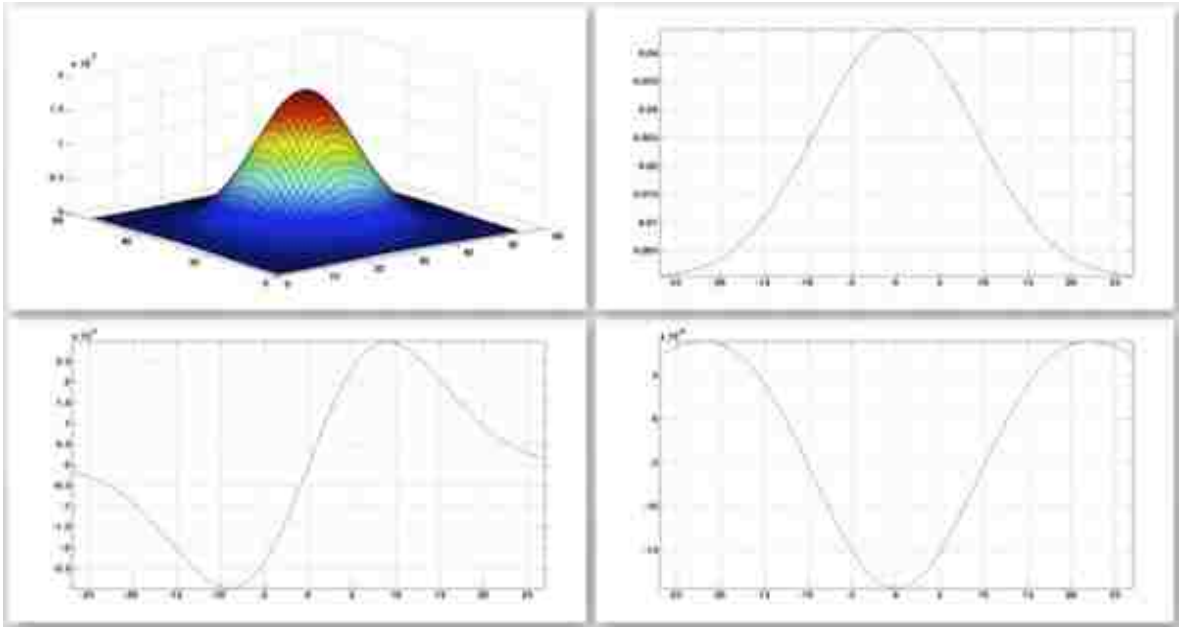


Figure 14: Gaussian kernels (upper left – Gaussian 2D, upper right – Gaussian 1D, lower left – 1D Gaussian $1^{st}$ order derivative, lower right – 1D Gaussian $2^{nd}$ order derivative).

Figures 15 and 16 shows the image derivative in x-direction and y-direction respectively for a reference image. The 1st image is the reference image and the successive images are the reference image convolved with a derivative kernel of increasing scale (scales used are 3, 9, 15, 30, 45 respectively). The derivatives are smoothed with appropriate Gaussian integration kernel of slightly larger scale. As the scale is increased the information we get decreases due to the reason that there are just few structures with such low spatial frequency change. (NOTE: Low spatial changes respond to derivative kernel with large scales).

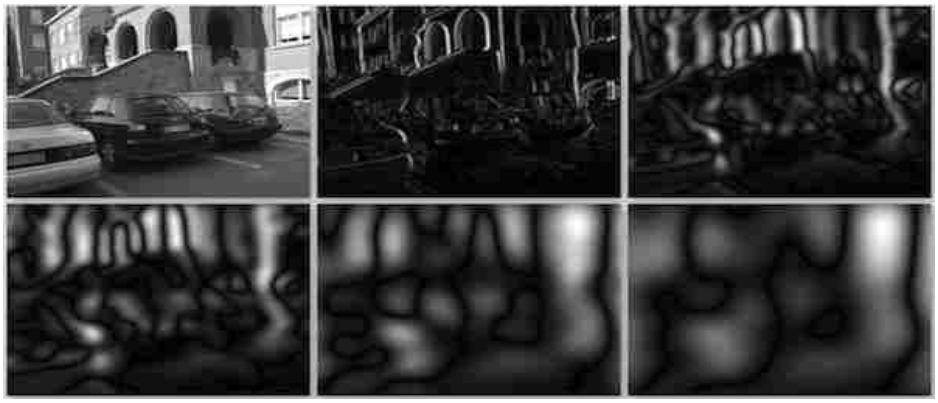


Figure 15: Image derivative in x-direction.

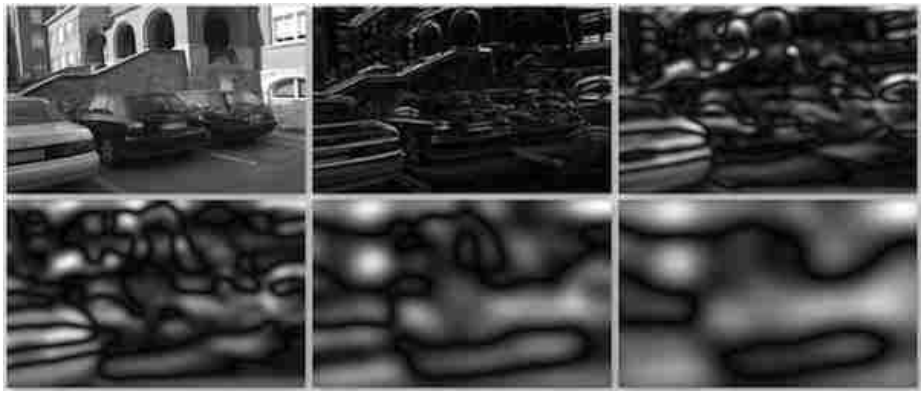


Figure 16: Image derivative in y-direction.

Figures 17 and 18 show the second derivatives of an image in x and y-direction respectively. These measures are used in calculating the Hessian measure. The scales used for the derivative kernels are the same as used for the first order derivatives. The second derivative

gives small values exactly at the interest point and peaks near the actual position of the interest point. But this does not affect the calculation of the performance of the operators. Because the effect is the same in all the images and the points found in different images should be relatively the same.

Figure 17: Second order image derivative in x-direction.

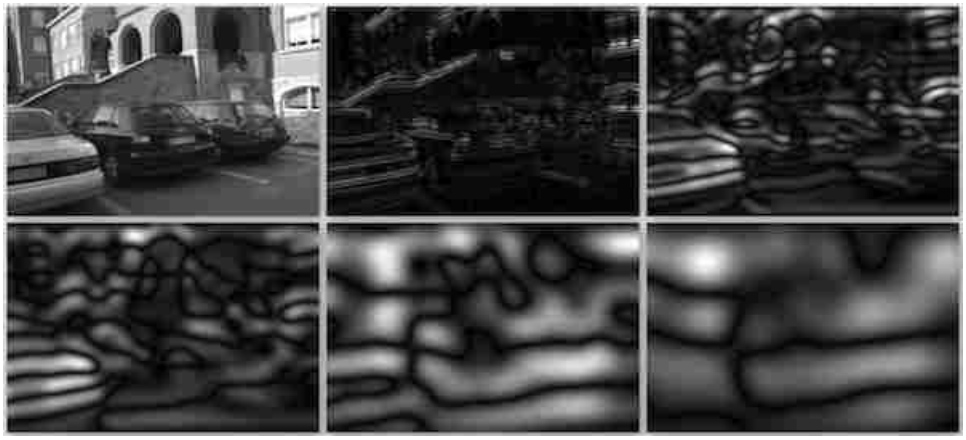


Figure 18: Second order image derivative in y-direction.

Thus after determining all the necessary image derivatives the multi-scale Harris or the Hessian measure for each of the contrast stretched image is found by selecting the scale-space maximum of the cornerness measure. The figures 19 through 27 show the response of the Hessian operator to the input contrast stretched image at different scales. There are 9 contrast stretched images and 11 different scales used. The first image in each of the figures is the

contrast stretched reference image of the illumination dataset shown in section 5. The following images in each figure are the Hessian responses at the 11 successive scales between 1.5 and 45. The last image is the cornerness measure of each pixel corresponding to the scale-space maximum. Since at the boundaries the convolution assumes 0 for the pixels that are not inside the convolution kernel, the response that we get will not be accurate. So in practice some of the values are ignored along the image boundaries. By doing this we can make sure that the convolution values are correct.

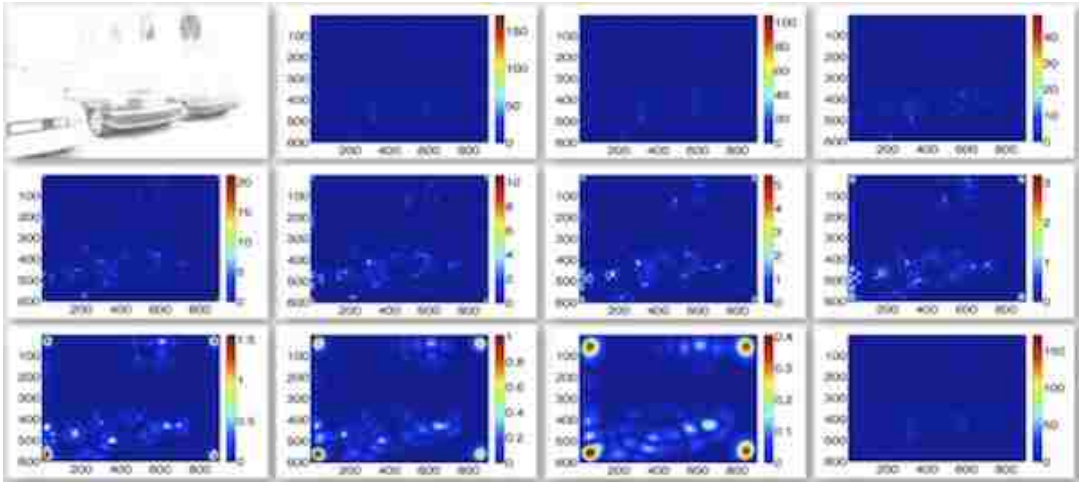


Figure 19: Multi-scale Hessian response to image stretched around the contrast 0.03.

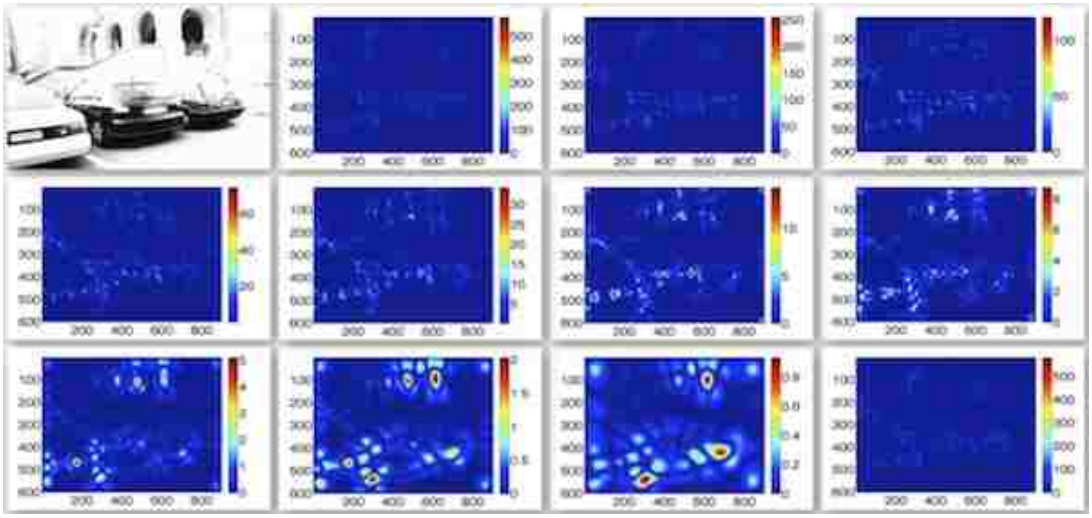


Figure 20: Multi-scale Hessian response to image stretched around the contrast 0.15.

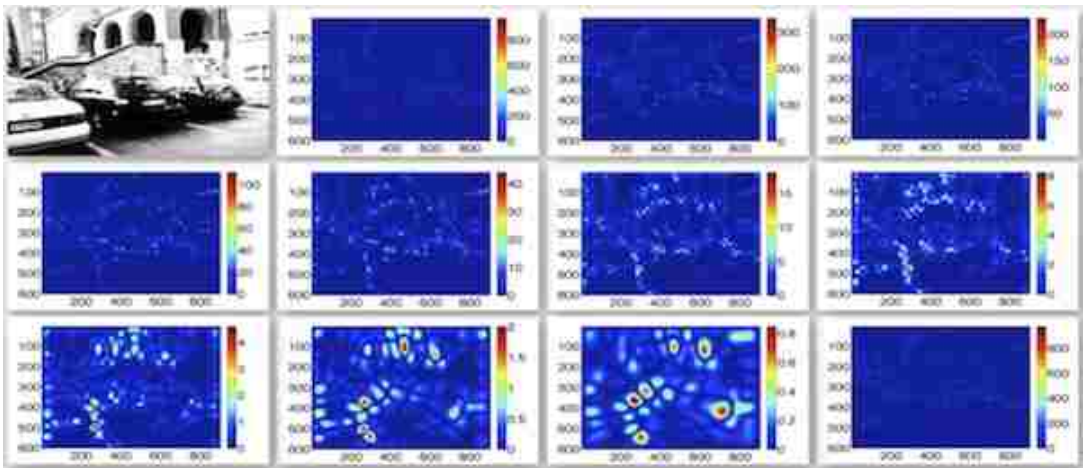


Figure 21: Multi-scale Hessian response to image stretched around the contrast 0.27.

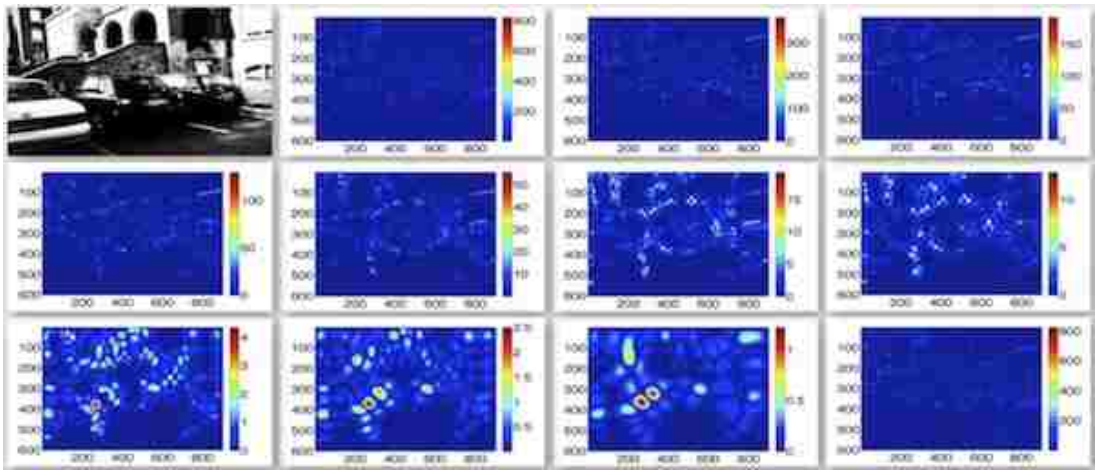


Figure 22: Multi-scale Hessian response to image stretched around the contrast 0.38.

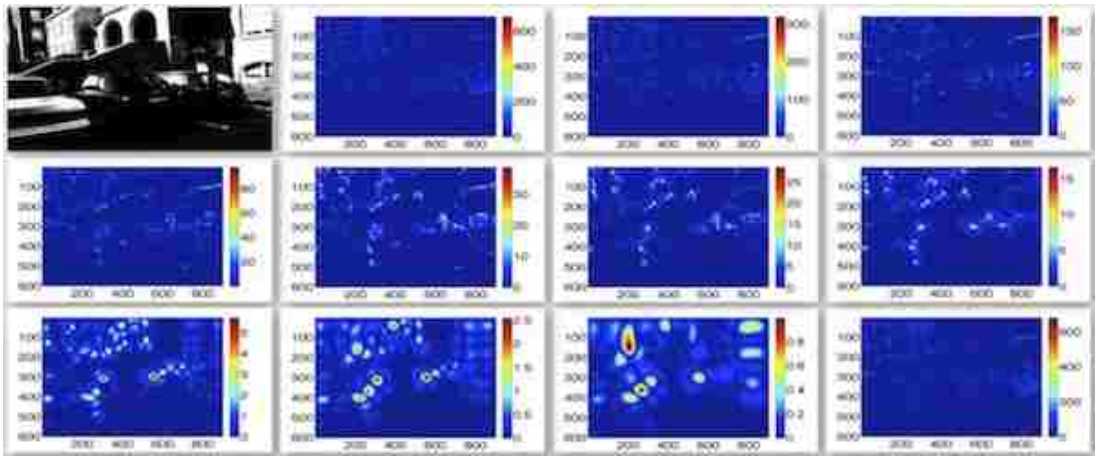


Figure 23: Multi-scale Hessian response to image stretched around the contrast 0.38.

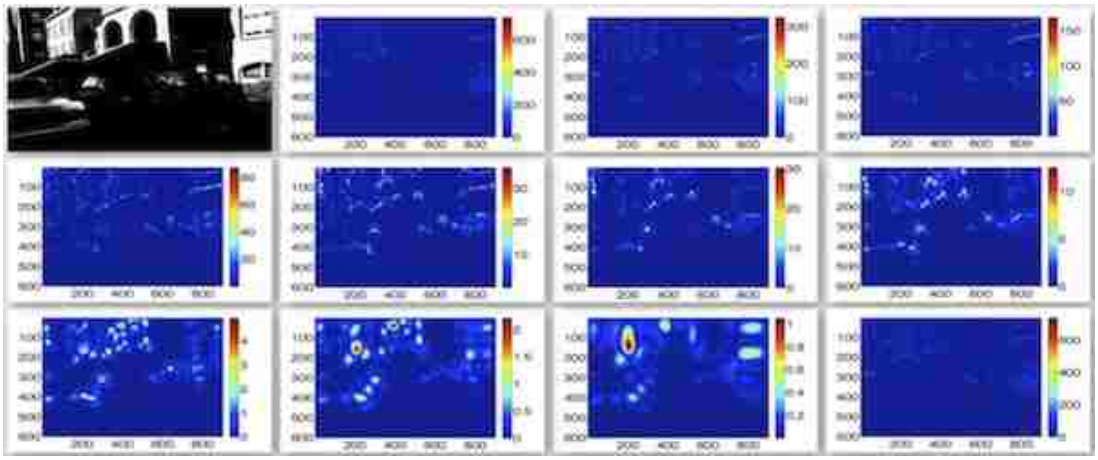


Figure 24: Multi-scale Hessian response to image stretched around the contrast 0.62.

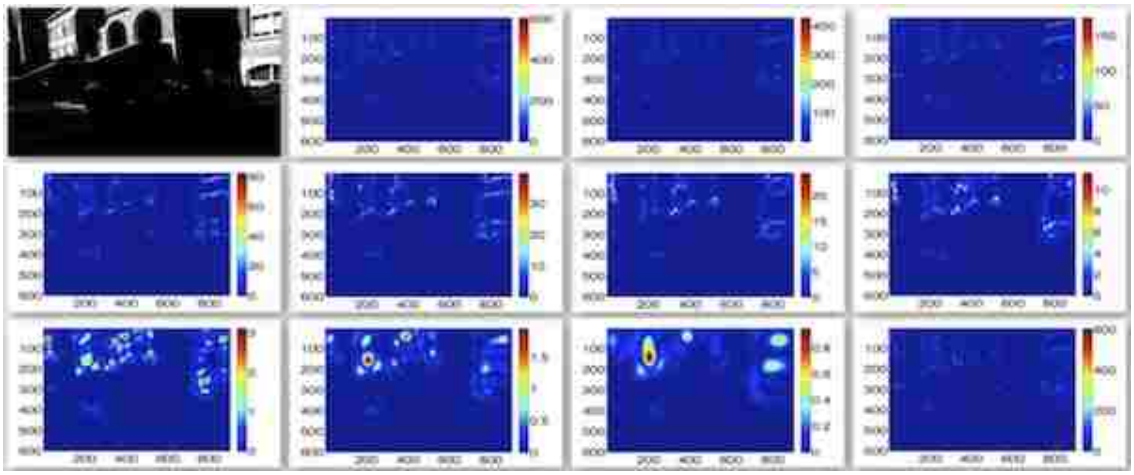


Figure 25: Multi-scale Hessian response to image stretched around the contrast 0.74.

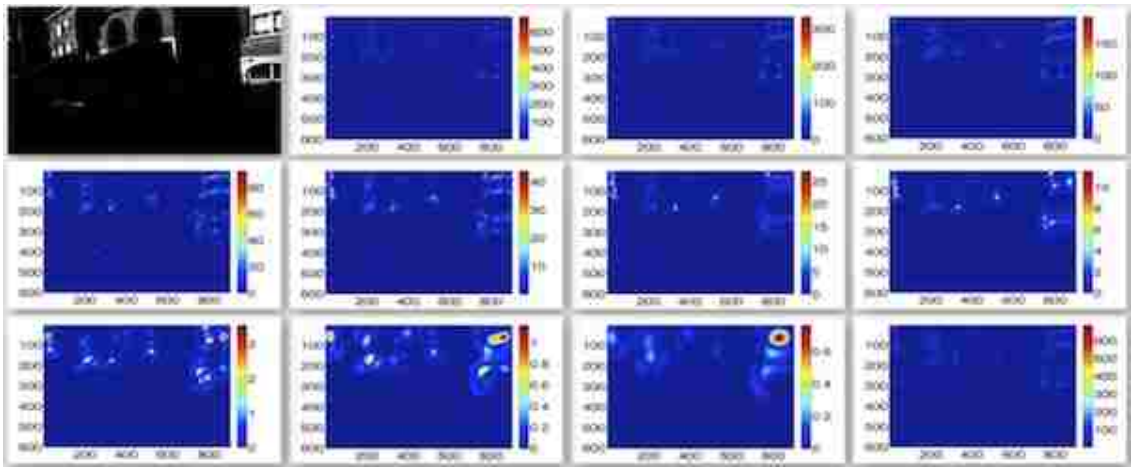


Figure 26: Multi-scale Hessian response to image stretched around the contrast 0.85.

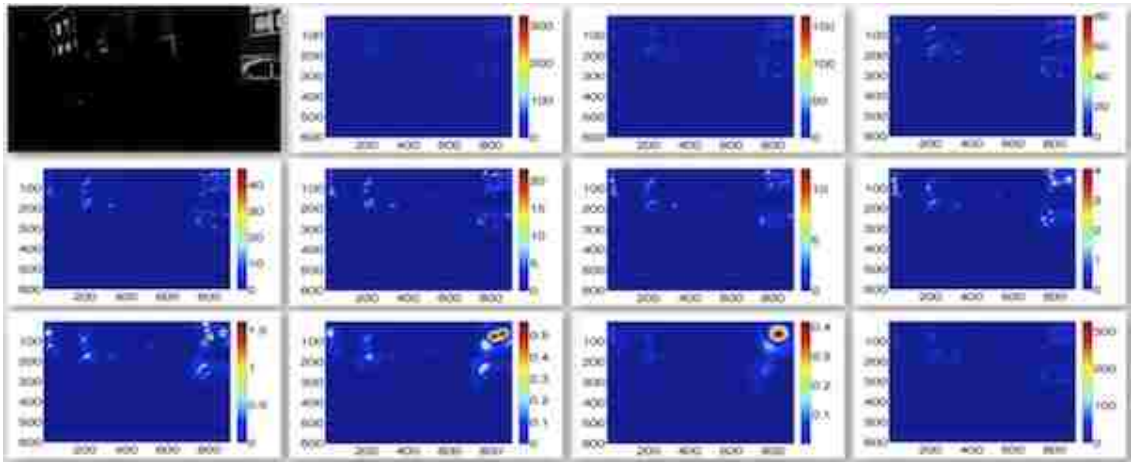


Figure 27: Multi-scale Hessian response to image stretched around the contrast 0.97.

To understand how the operators respond to the corners and other regions the operators response to some selected points on different contrast stretched images should be analyzed. Figure 28 shows the response of the multi-scale Hessian across the contrast-space to four manually selected points. The responses are the scale-space maximum of the points in each contrast stretched image. Image (a) shows the corner strength of the selected corner pixel. Image (b), (c) and (d) shows the corner strength of three other points near the corner pixel corresponding to (a). It is clearly seen that the operator gives high response at the corner and the pixels near have low responses over the contrast-space.

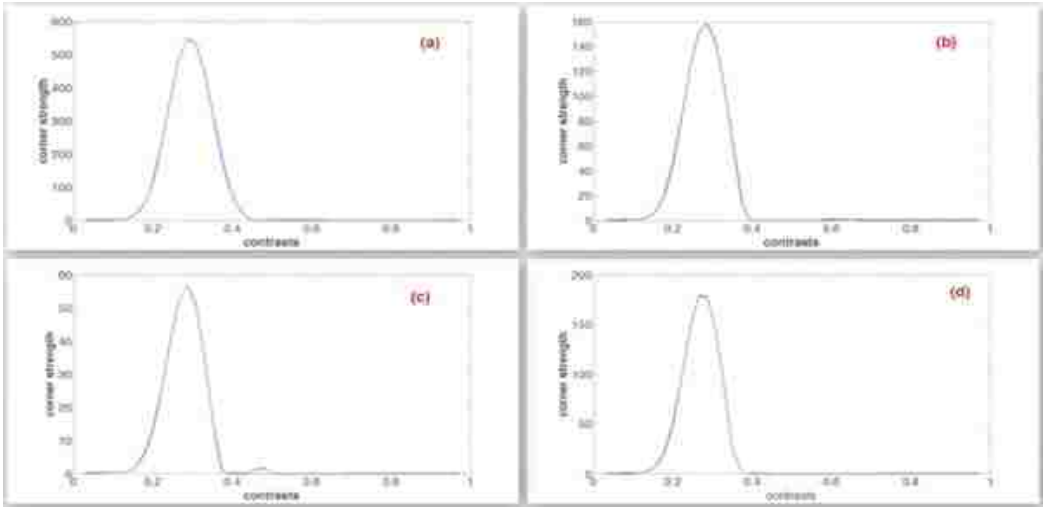



Figure 28: Corner strength of pixels across contrast-space.

The figure shown is the response of the operators to each of the contrast stretched images. The actual response used in this research is the sum of the responses across the contrast space. By summing up the cornerness the corner strength will be sufficiently high and can be easily distinguished from the rest of the weak points. Intuitively the Harris-IRFET and Hessian-IRFET operators apart from finding points in an illumination invariant way they also boost the responses of the multi-scale Harris and multi-scale Hessian respectively to separate out the good interest points.

### 4.1.3 Stage 3 – Point Detection

After determining the multi-scale measure for each of the contrast stretched images, the sum of all the responses is calculated. Then the last step is to eliminate weak points using non-maxima suppression with thresholding. In this experiment non-maxima suppression of radius 3 has been used to test and validate all the operators. The threshold used for the non-maximum suppression stage should be carefully chosen because they play a vital role in eliminating noisy points. Experimental results showed that the Harris-IRFET and the Hessian-IRFET operators performed at their best when the threshold was chosen as a function of the maximum value of the final measure instead of a constant threshold for all the contrast stretched images. For the Harris-IRFET the threshold was chosen as 1% of the maximum measure and for the Hessian-IRFET the threshold was chosen as 5% of the maximum cornerness value.

Figure 29 shows the reference image followed by the output of the multi-scale Hessian on each of the contrast stretched images. The 2$^{nd}$ image from the last is the total sum of all the measures from the contrast stretched images. The last image shows the points detected using non-maxima suppression and thresholding operations.

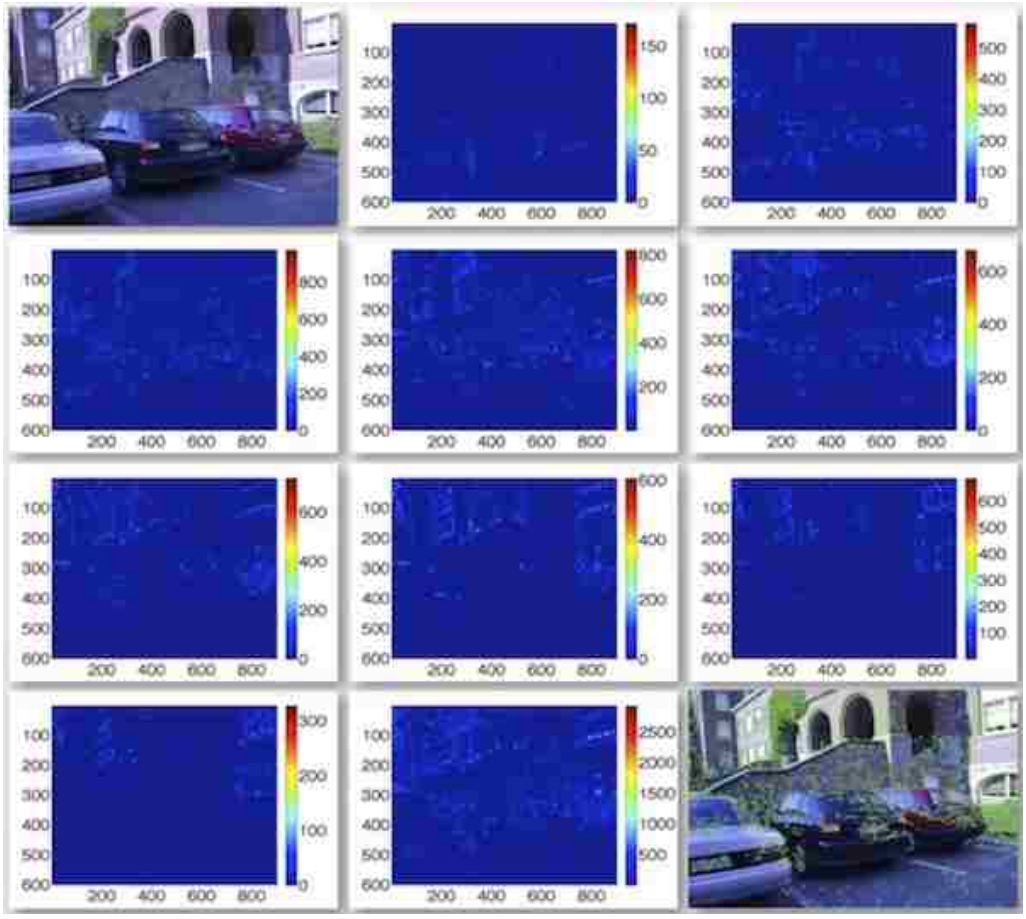


Figure 29: Overview of Hessian-IRFET operator.

## 4.2 Implementation

### 4.2.1 MATLAB and C++

The algorithm is implemented in MATLAB and C++. MATLAB implementation of Harris-Laplace, Hessian-Laplace and the proposed methods is used in validation. Execution time in MATLAB for the proposed algorithms is in average 75 seconds per image. The proposed method has also been implemented in C++ using the OpenCV libraries to make practical application possible. The execution time in C++ is in average 45 seconds per image. The execution time is tested on Intel dual core i7 processor with a clock speed of 2.7GHz. The implementation can be optimized further, by using multithreaded programming.

### 4.2.2 GPU - OpenCL

Still the implementation will be slow for images with very high resolution. One way to tackle this problem is by parallelizing the operatins. By using Graphics Processing Units (GPU), Digital Signal Processors (DSP) and multi core CPU's the implementation can be parallelized to a great extent. These performance accelerators have several processing units that are capable of doing basic arithmetic operations in parallel. By efficiently using these units we can considerably improve the performance. To motivate the future improvements of the proposed algorithm, the algorithm has been implemented using OpenCL API, which is an interface between the application and the above mentioned performance accelerators. Figure 30 shows the platform model used by OpenCL [9]. The host is the machine to which the accelerators are connected. A compute device is a GPU device, multi-core processor or any accelerator device. A host can be connected to any number of such devices. A compute device is subdivided into compute units that are further divided into processing elements. The processing elements are the units capable of performing arithmetic operations.

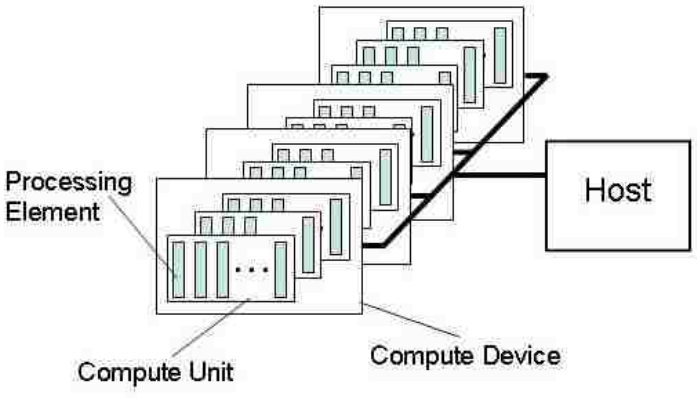



Figure 30: OpenCL platform model [9].

Figure 31 shows the execution model [9]. *NDRange* specifies how the data is organized and executed. At present the maximum number of dimension supported is 3 and the size depends on

the device. Input data is organized as workgroups which in turn divided into workitems. Each workgroup executes on a compute unit and each workitem executes on a processing element.

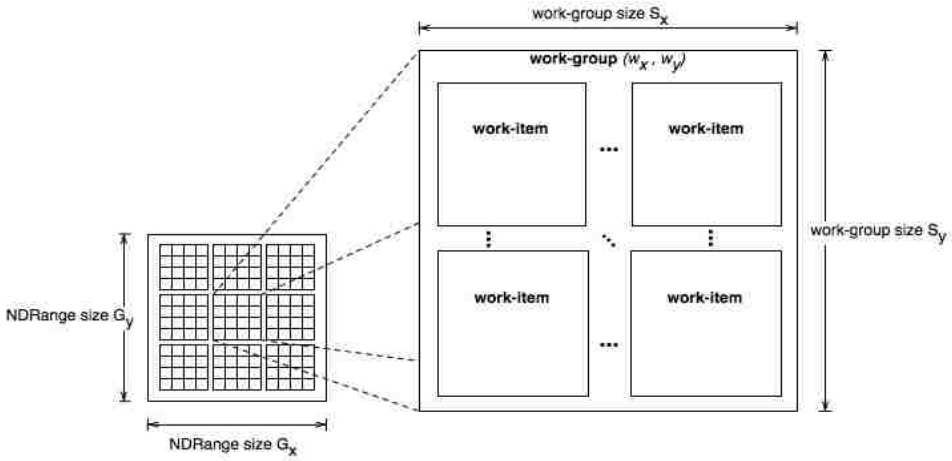



Figure 31: OpenCL execution model [9].

Figure 32 shows the memory model of an OpenCL device [9]. Global memory is the device memory and can be accessed by any workgroup. Global memory access is the slowest. Local memory is local to a workgroup and can be accessed by any work item inside a workgroup. Local memory access is faster than the global access. Private memory can only be accessed by work item and it is the fastest memory access.

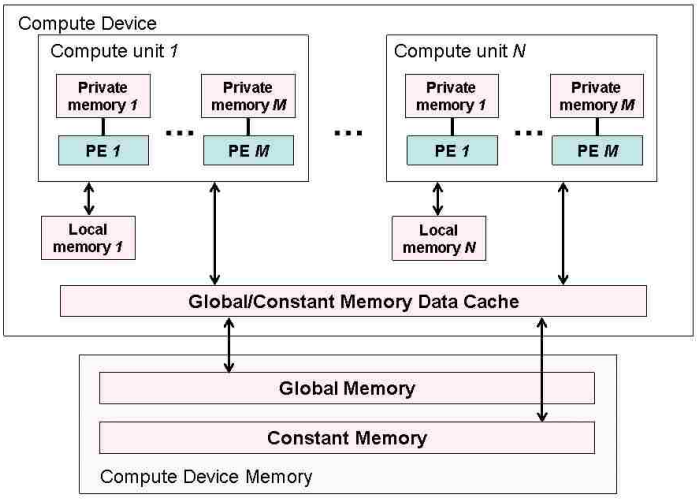



Figure 32: OpenCL memory model [9].

The important factor that affects the performance of the GPU implementation is that the data transfer overhead between the main memory and the accelerators memory. To perform computations on the accelerators, the data has to be transferred to the memory of the accelerators and then retrieve the data after the process is done. To efficiently implement an algorith minimum data transfer should occur between the systems main memory and accelerators memory. Also by using the better accelerators it is possible to achieve very high speed for very high-resolution images. The reason for concentrating on very high-resolution images is that for low resolution images the regular implementation might work better than the GPU implementation because of the data transfer issue mentioned before.

The efficiency can be achieved by data parallel or task parallel implementation. In data parallel implementation a set of instructions is applied to each of the input elements, in our case to each pixel. In task parallel implementation execution processes are distributed across the different nodes i.e. the processors. Since convolution is extensively used and convolution is highly data parallel, the performance can be improved by data parallel implementation. The algorithm has been implemented and optimized using OpenCL API to use the GPU resource and a *3 times* speedup has been achieved over the OpenCV implementation on the same machine which had higher end GPU and quad core processor. The total time taken by the algorithm is 4.5 seconds per image. The benchmark was tested on a machine with NVIDIA GeForce video card with 512 cuda cores (processing elements). The execution time depends on the architecture of the GPU and the number of cores it has. So with higher end GPU's a tremendous improvement in the execution time can be achieved. Figure 33 shows a flow chart that gives a pictorial representation of the proposed algorithms.

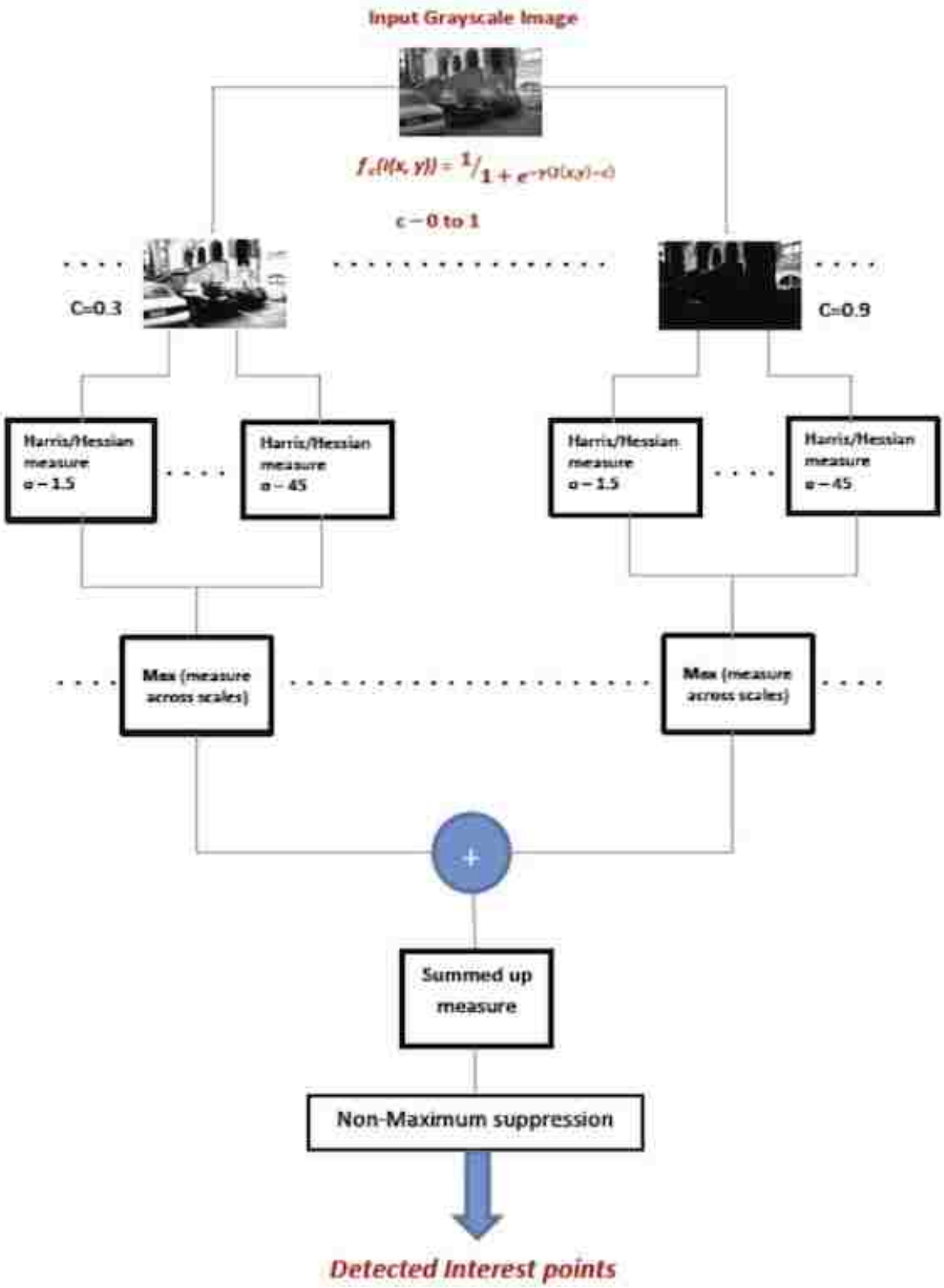



Figure 33: Overview of the proposed algorithms.

# CHAPTER 5 – RESULTS

## 5.1 Dataset

The dataset used to validate the performance of the algorithm is obtained from the website [14]. In validating the algorithms five image datasets with five different variations (illumination, blur, compression, viewpoint and scale) is used. Each set has a reference image and 5 other images with affine or illumination variations. The homography (mapping) between the reference image and the other images are known, which is used to calculate the repeatability score. Figure 34 shows the dataset used. In the figure row 1 is the illumination dataset, row 2 is the blur dataset, row 3 is the compression dataset, row 4 is the viewpoint dataset and row 5 is the dataset with scale variations. The 1$^{st}$ image in each dataset is the reference image and the following images have successively increasing variations from the reference.

Figure 34: Dataset [14].

## 5.2 Repeatability Score

In this research the point repeatability score is used as the performance measure to validate the proposed algorithms. There are many ways to define the repeatability of a detector. Here a straightforward algorithm to calculate the repeatability is used. A point in one image is said to be a match if there is an interest point in the 3x3 neighborhood of the other image surrounding the projected point. Let us assume *n* interest points were detected in the reference and *m* points in the other test image. The repeatability score is given by $\frac{number\ of\ feature\ matches}{\min(n,m)}$ x 100%, where *number of feature matches* is the number of matched interest points common to both the images.

There might be changes in camera position while registering the images i.e. in the case of scale and viewpoint change, a part of scene will not be present in the other. Thus the points lying in the area common to both the images has to be determined and the interest points that lie in this common area should be taken into account for calculating the repeatability measure. The features that are common to both the images can be obtained by projecting both the reference and the test image onto each other and to find the outliers by detecting which pixels went out of bound in the corresponding projected image. Steps followed to find the repeatability score are listed below.

STEPS:

1. Project the interest points detected on the images onto one another.
2. Reject the points that are projected out of boundary in the other image.
3. For each point in one image check whether there is a point in another image inside a 3x3 neighbourhood.
4. Calculate the repeatability score using the formula mentioned above.

### 5.2.1 Illumination

The illumination dataset contains six images taken under different lighting conditions. Since the proposed method is focused on images with illumination changes, the operator was expected to perform at its best. Figure 35 and 36 shows the feature points detected using the operators. Figures 37 and 38 shows the repeatability score and the number of features detected on the images. The results prove that the Harris-IRFET and Hessian-IRFET perform better than Harris-Laplace and Hessian-Laplace.

Figure 35: Feature points: Harris-IRFET (left), Harris-Laplace (right).

Figure 36: Feature points: Hessian-IRFET (left), Hessian-Laplace (right).

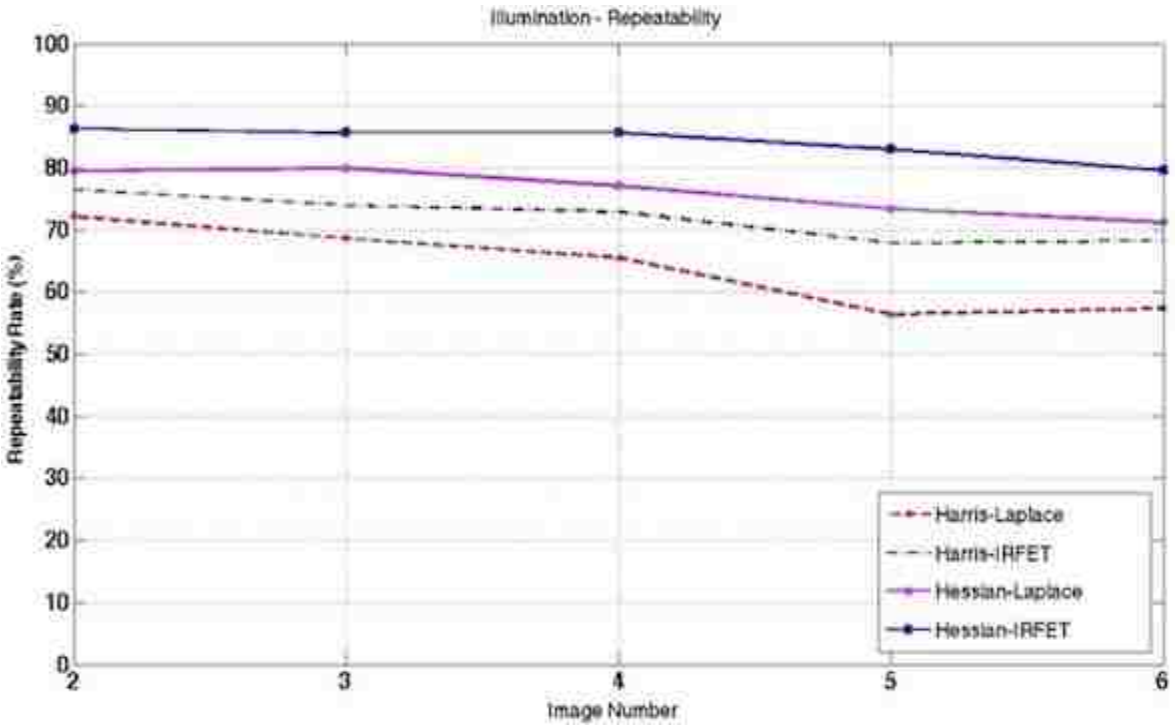



Figure 37: Repeatability plot - Illumination dataset.

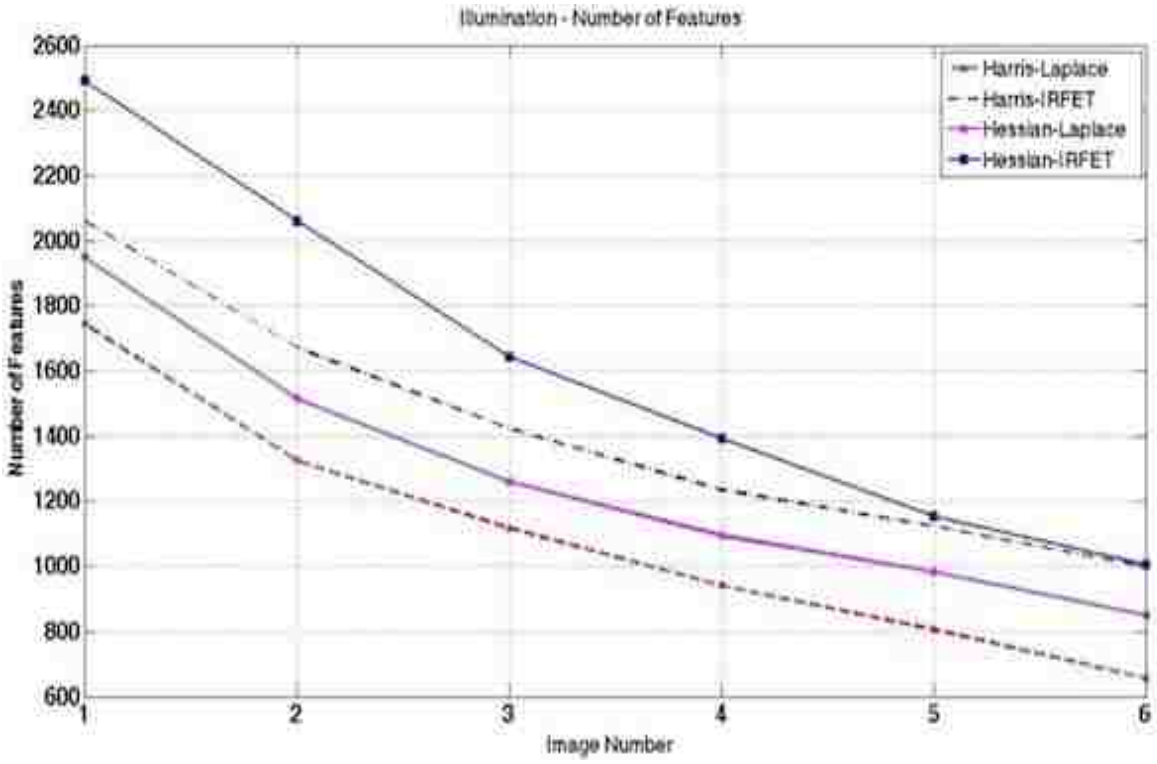



Figure 38: Number of features - Illumination dataset.

### 5.2.2 Blur

From the repeatability plot shown in figure 39 it can be seen that the performance was better in most cases. Since the dataset contains images with low contrast regions that will be difficult to detect without contrast stretching the images the proposed method performs well for some images. Figure 39 and 40 shows feature points detected using the operators. Figure 41 and 42 shows the repeatability and the number of features detected for the blur dataset.

Figure 39: Feature points: Harris-IRFET (left), Harris-Laplace (right).

Figure 40: Feature points: Hessian-IRFET (left), Hessian-Laplace (right).

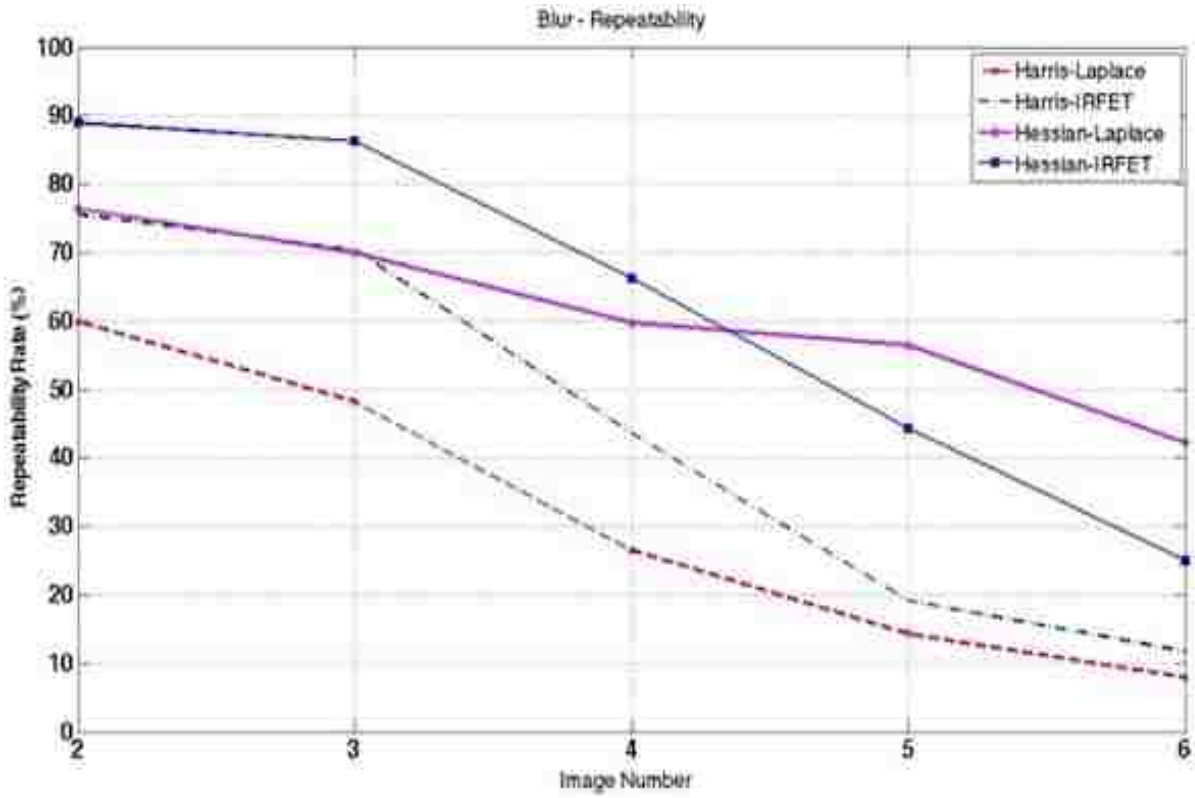



Figure 41: Repeatability plot - Blur dataset.

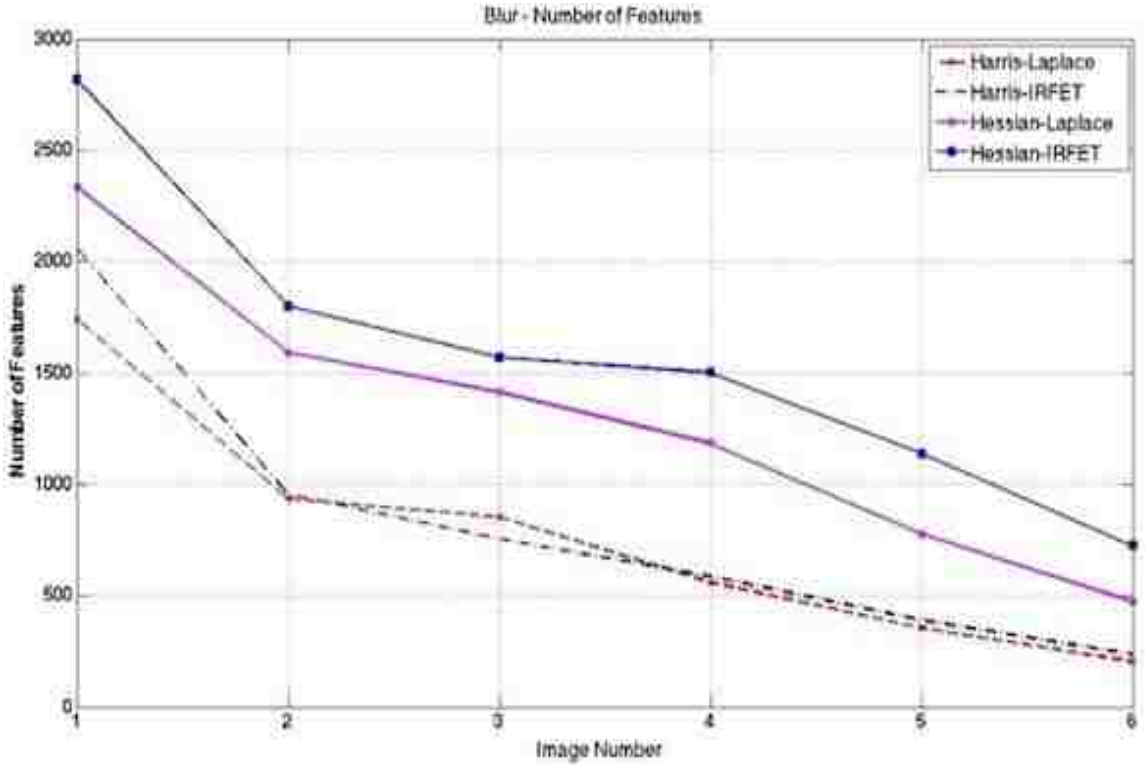



Figure 42: Number of features – Blur dataset.

### 5.2.3 Compression

The proposed method has comparable performance with the Harris and Hessian Laplace methods for images with compression. Figures 45 and 46 shows the repeatability rate and number of features detected on the images in compression dataset. Figure 43 and 44 shows the interest points detected using the operarors.

Figure 43: Feature points: Harris-IRFET (left), Harris-Laplace (right).

Figure 44: Feature points: Hessian-IRFET (left), Hessian-Laplace (right).

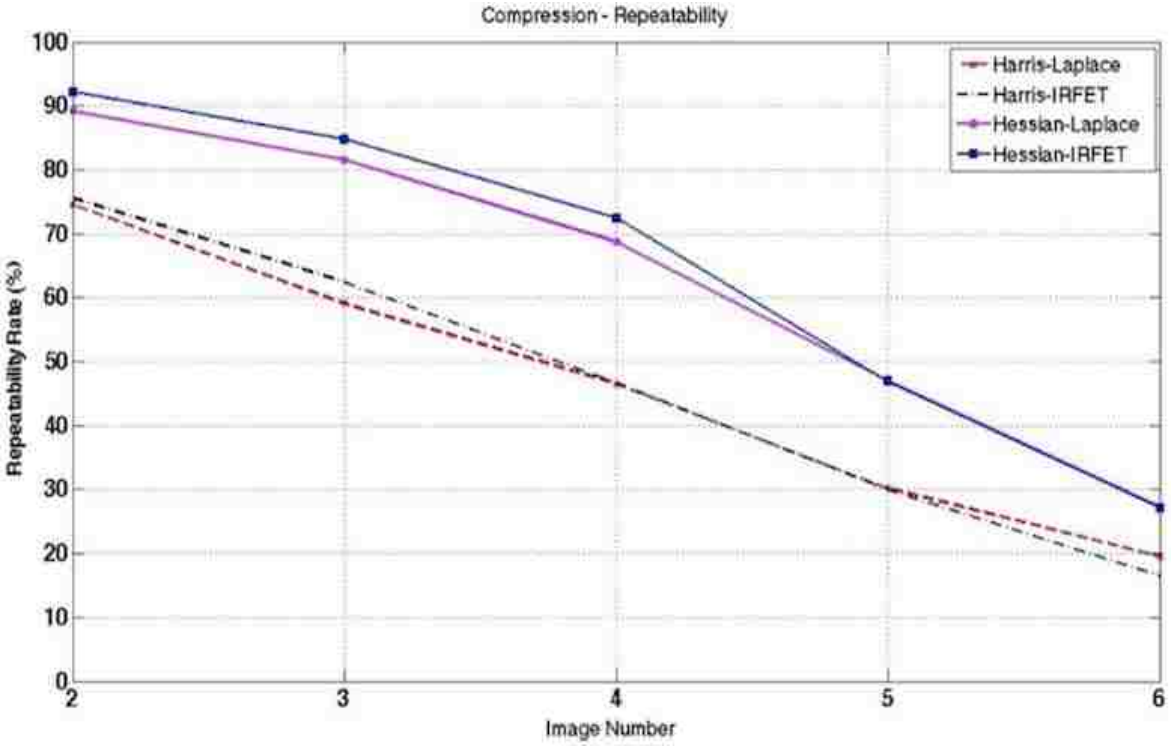



Figure 45: Repeatability plot - Compression dataset.

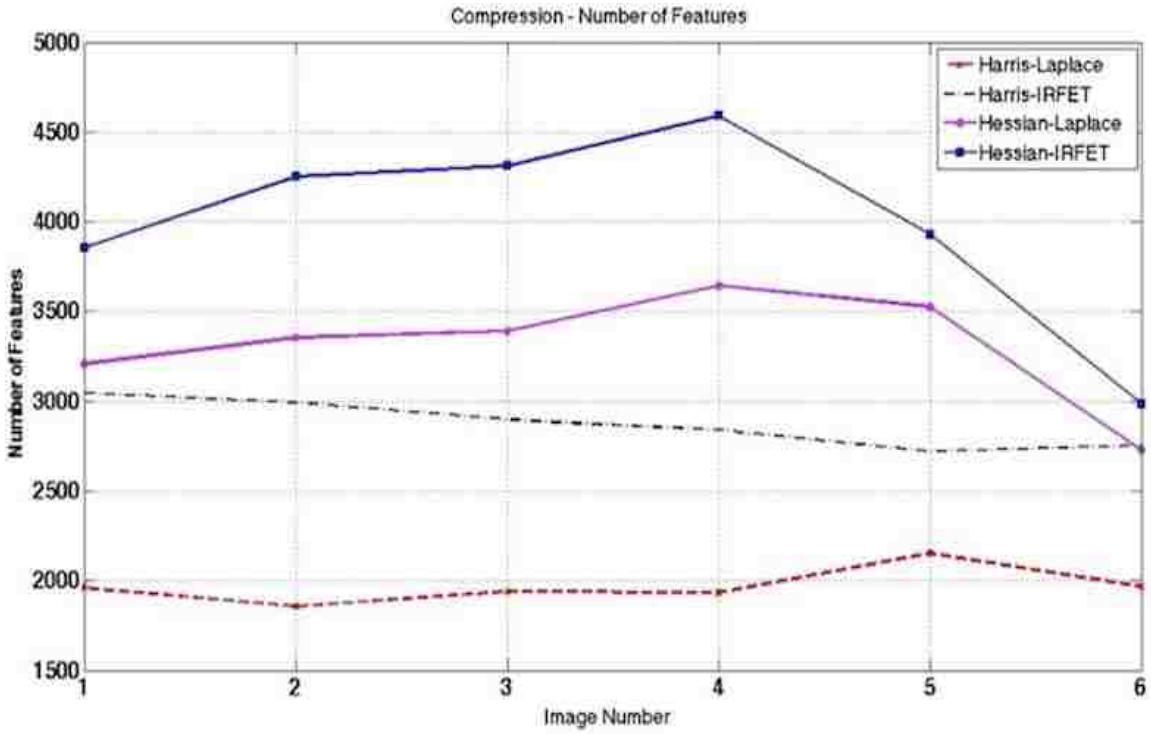



Figure 46: Number of features - Compression dataset.

### 5.2.4 Viewpoint

In the case of viewpoint change the proposed methods performed better than Harris-Laplace and Hessian-Laplace. Viewpoint changes are usually accompanied by illumination changes and due to this reason, the proposed operators performed better. Figures 49 and 50 shows the repeatability rate and number of features on the images in viewpoint dataset. Figure 47 shows the feature points detected using Harris-IRFET (left) and Harris-Laplace (right). Figure 48 shows the feature points detected using Hessian-IRFET (left) and Hessian-Laplace (right).

Figure 47: Feature points: Harris-IRFET (left), Harris-Laplace (right).

Figure 48: Feature points: Hessian-IRFET (left), Hessian-Laplace (right).

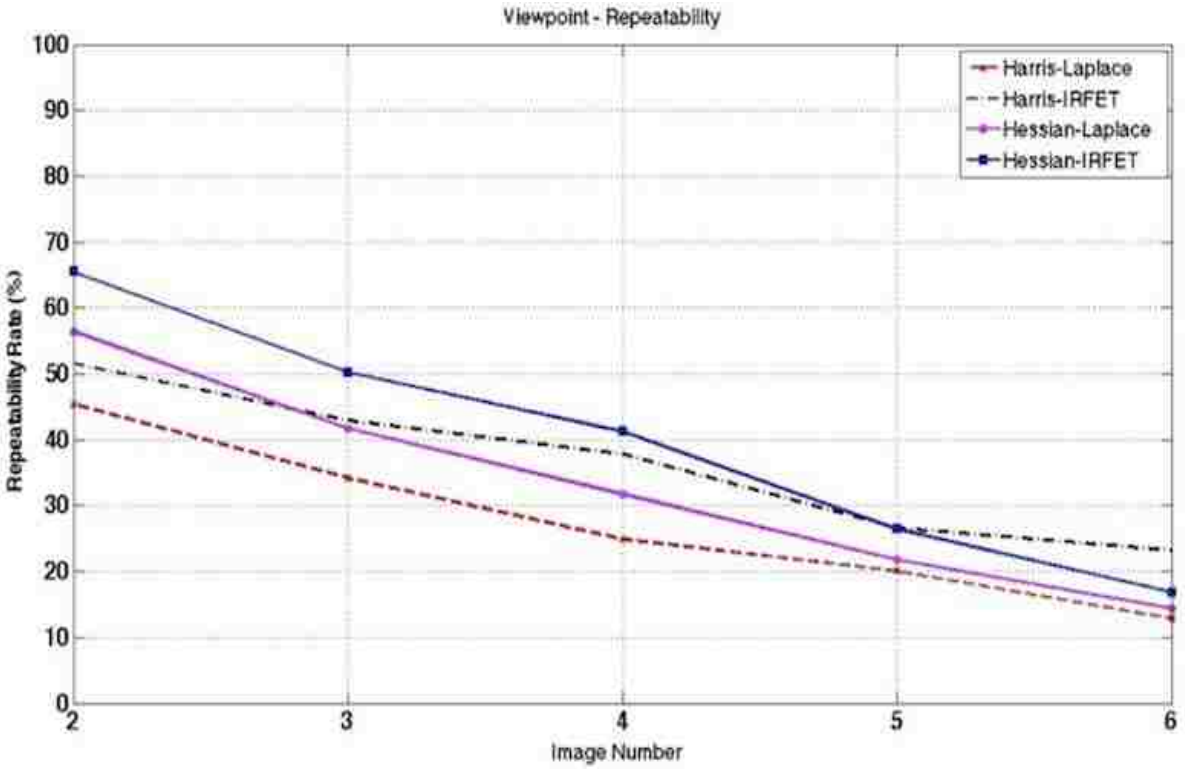



Figure 49: Repeatability plot – Viewpoint dataset.

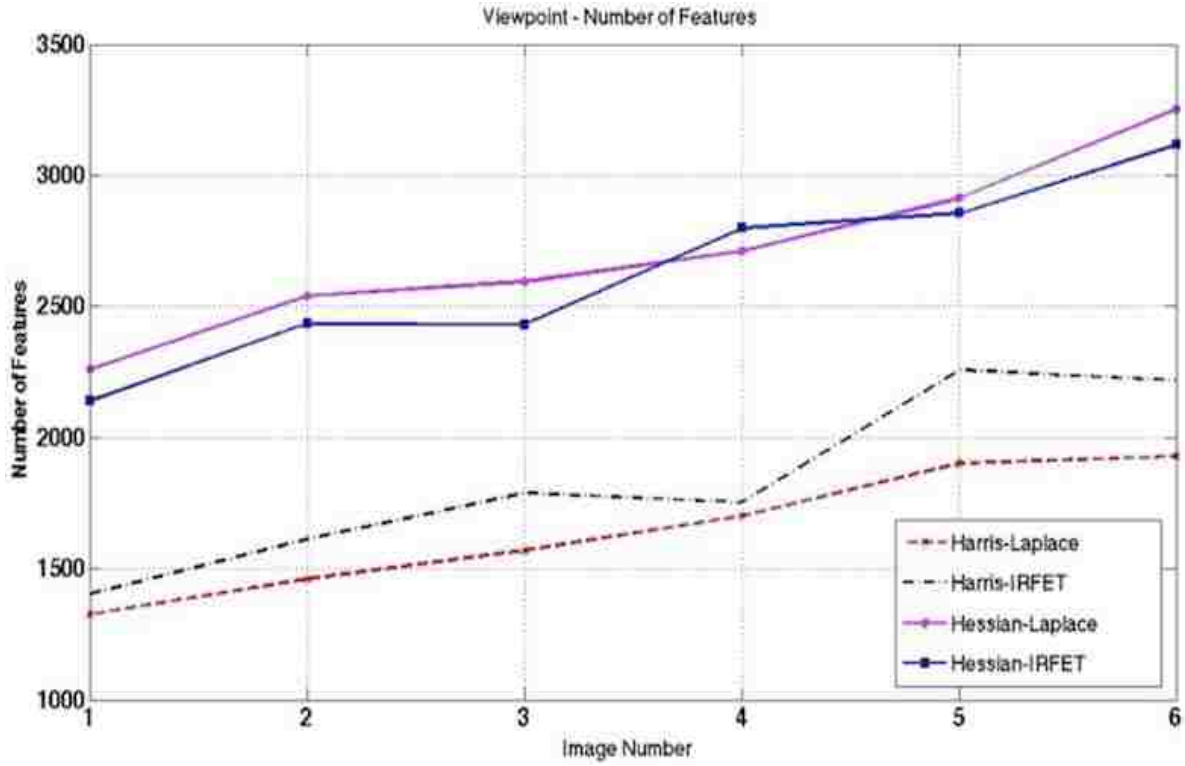



Figure 50: Number of features - Viewpoint dataset.

### 5.2.5 Zoom and Rotation

The proposed operators have comparable performance with the Harris and Hessian Laplace operators in the case of zoom and rotation changes. All the operators perform low under high variations. This is due to the reason that we use discrete number of scales and the variations are high. Figure 53 and 54 shows the repeatability rate and the number of features detected using the four operators. Figure 51 shows the feature points detected using Harris-IRFET (left) and Harris-Laplace (right). Figure 52 shows the feature points detected using Hessian-IRFET (left) and Hessian-Laplace (right).

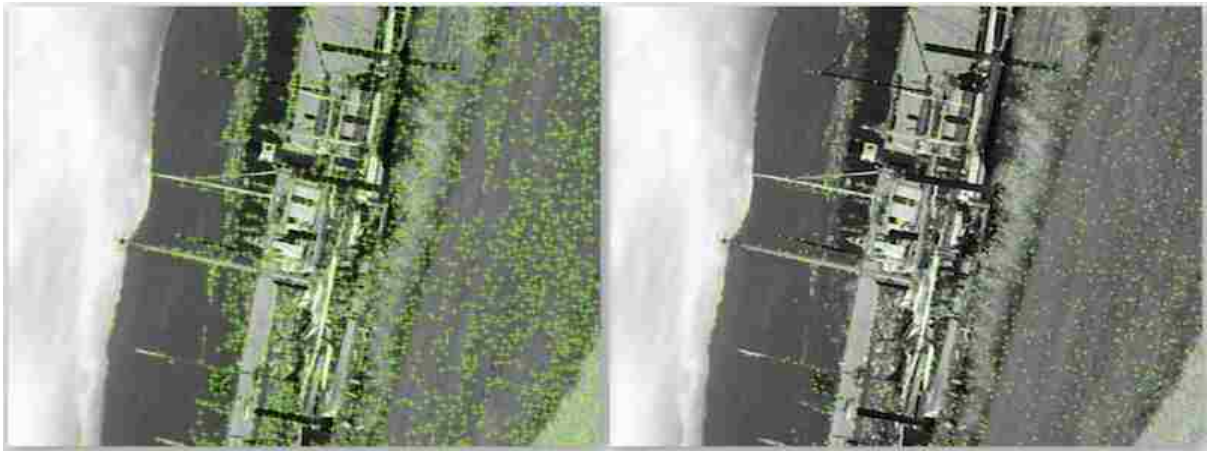


Figure 51: Feature points: Harris-IRFET (left), Harris-Laplace (right).

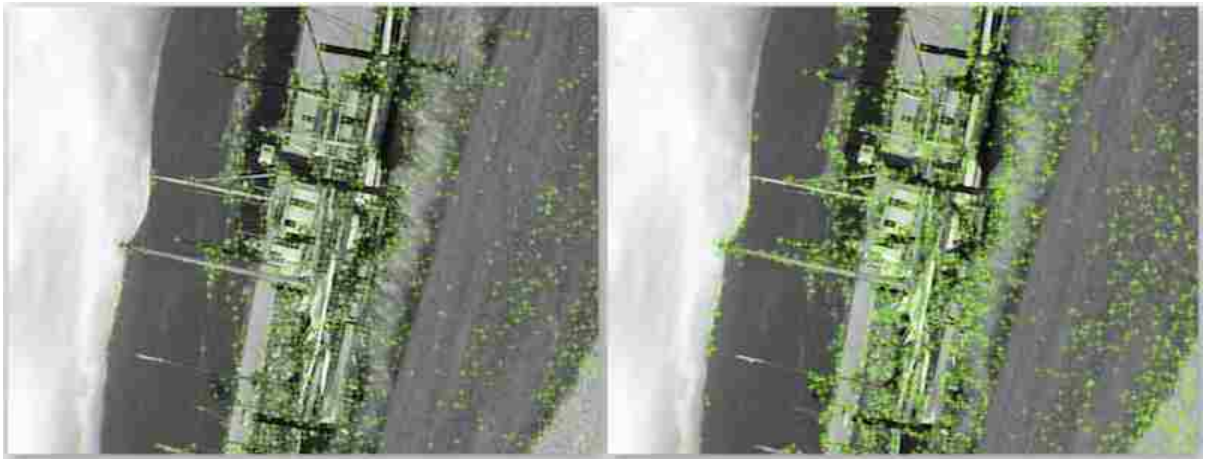


Figure 52: Feature points: Hessian-IRFET (left), Hessian-Laplace (right).

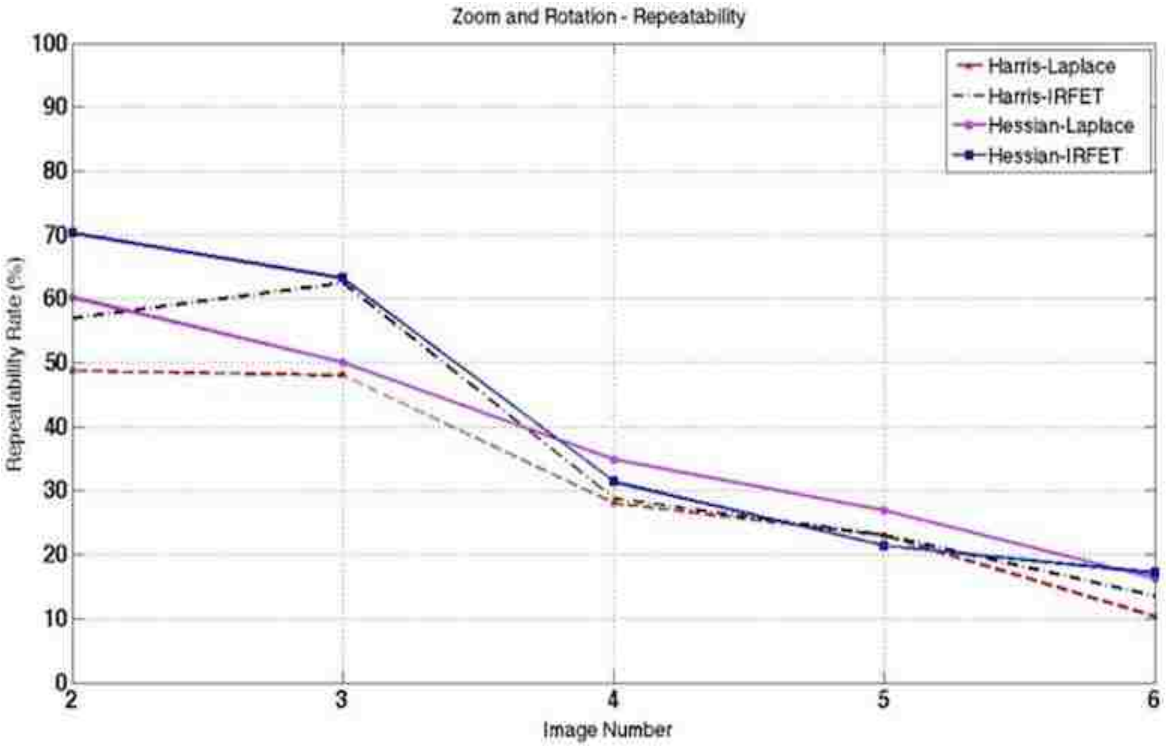



Figure 53: Repeatability plot – Zoom and Rotation dataset.

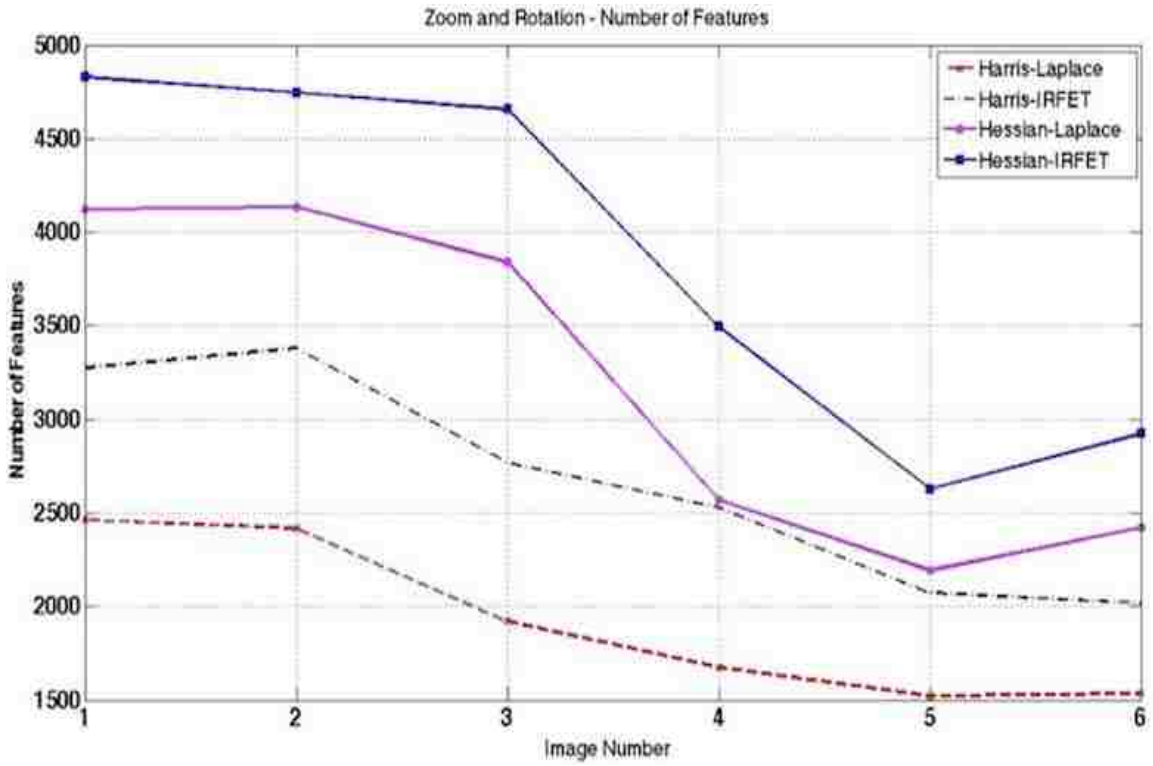



Figure 54: Number of features - Zoom and Rotation dataset.

### 5.3 Scale Efficiency

Scale selection is an important part of the scale invariant operators. Operators like SIFT[12], SURF [2], affine invariant Hessian and Harris [15] depend on the scale and position of the interest points found by the initial stage scale invariant interest point detectors. One way to detect the scale for the proposed detectors is to select the scale corresponding to the scale-space maximum in the contrast stretched image that has the highest cornerness measure (i.e. the scale corresponding to the scale-space maximum of the image that is also a contrast-space maximum). Another way is to get the scales corresponding to scale-space maximum from all the contrast stretched images and then do some normalization based on the cornerness measure. Experiments showed that these methods did not work well. This is due to the reason that the contrast stretching the image does not preserve the scale of its objects. Objects might be detected at different scales in different contrast stretched image.

One possible solution is to determine the scales on the original image after detecting the interest points using the Harris/Hessian-IRFET methods. This idea has been studied and validated against the scale selection procedure of Hessian-Laplace and Harris-Laplace. SIFT feature descriptor is a rotation and illumination invariant feature descriptor algorithm that depends on the scale of the interest points detected by its initial stage. In our experiment the scale and interest points detected by the proposed methods are given as input to the SIFT descriptor to get the feature descriptions. By using these descriptions from different images the feature matching is done and tested for the accuracy using the match rate.

The illumination dataset has been used to study the proposed operators scale selection property. After getting the feature descriptions from the images the features from the reference image were matched with the features from other images using FLANN (Fast Library for

Approximate Nearest Neighbors) OpenCV library. Figure 55 shows few of the matched features between the reference image and one of the other images from the illumination dataset.

Figure 55: Features mapped from the reference (left) to the test image (right).

The matched features are further checked for the point locality using the known mapping between the reference and other images. The mapping is nothing but the homography information between the images. Using the homography information the matched features can be checked whether they are matched to the same point in the other image or to some other similar objects in other parts of the image. Figure 56 shows the match rate of Hessian-IRFET and Hessian-Laplace for the features detected on images from the illumination dataset. Match rate is given by $\frac{correct\ matches}{total\ matches} \times 100\%$ where *total matches* is the sum of *correct matches* and mismatches. The match rate is almost same for both the operators. This is because the scale detected was not illumination invariant and the procedure to select the scales for the proposed operator was almost the same as the Hessian-Laplace. Figure 57 shows the number of correct matches. It can be seen that even though the match rate is the same for both operators Hessian-IRFET finds more correct matches than the Hessian-Laplace. This is due to the reason that Hessian-IRFET finds more good interest points that may lie in a low contrast region.

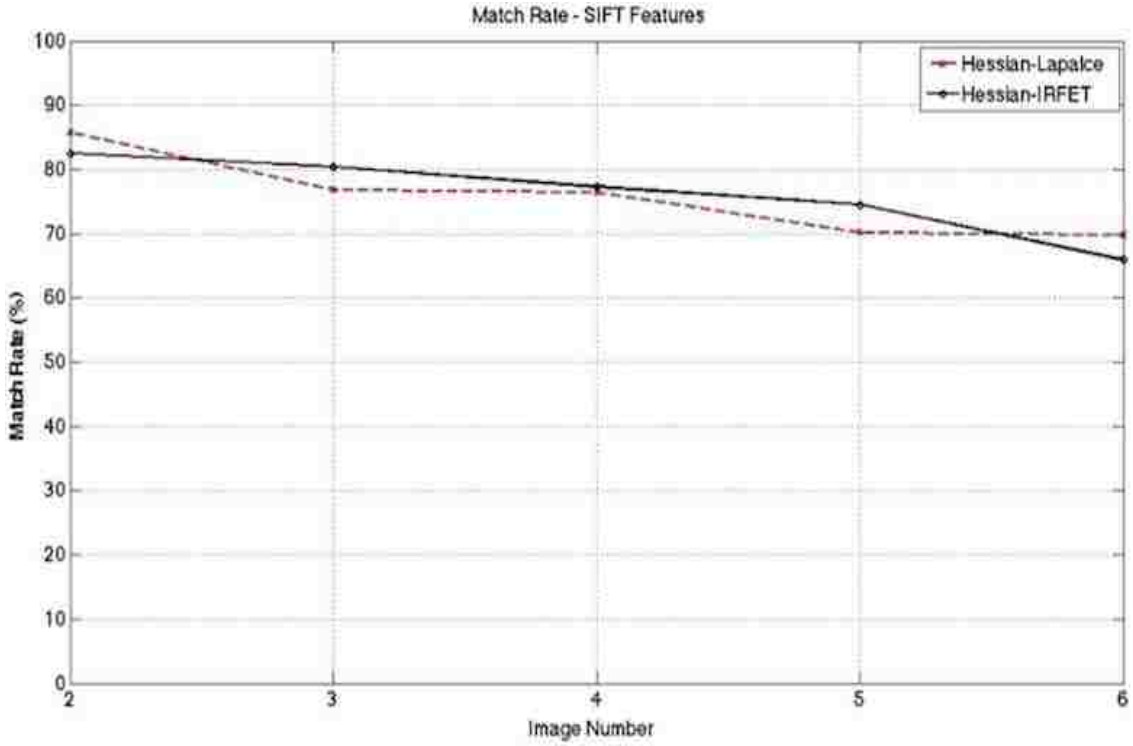



Figure 56: Match rate for the features between the reference and other images.

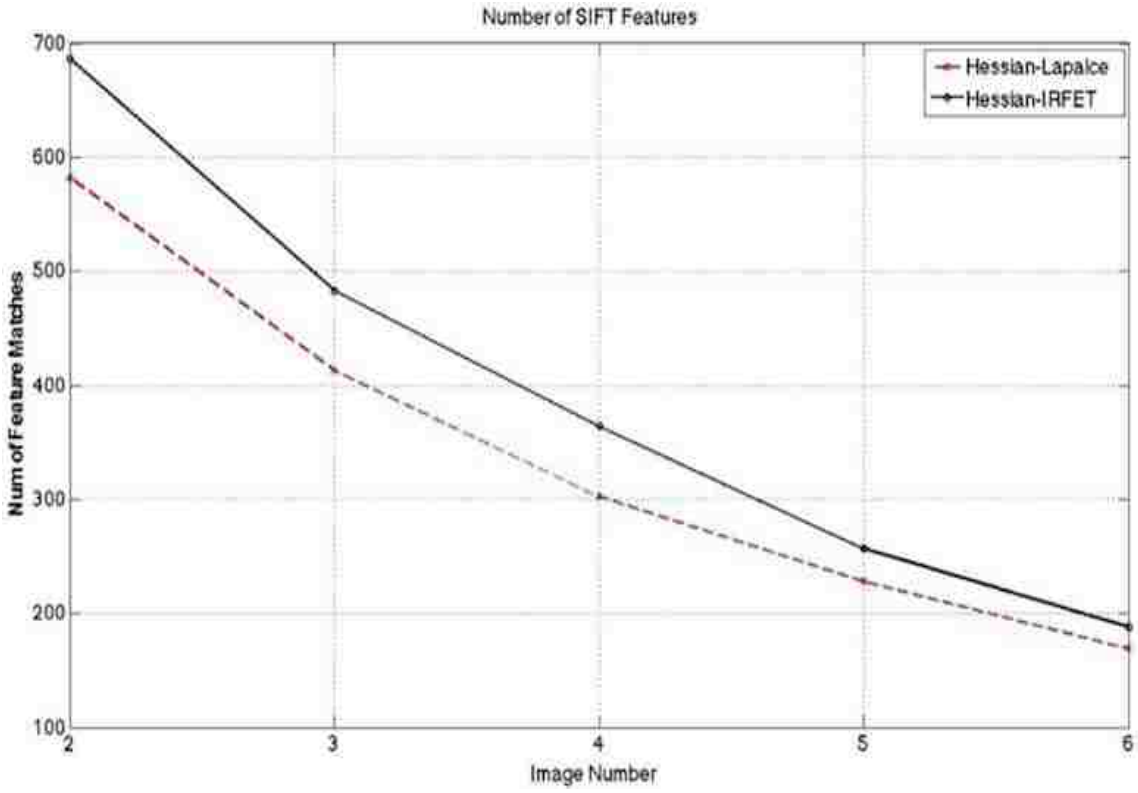



Figure 57: Number of features matched between the reference and the other images.

## 5.4 Conclusion

The proposed method improved the performance of Harris-Laplace and Hessian-Laplace for datasets having illumination, viewpoint and blur variations and thus proving to be a better scale invariant detector. Overall Hessian-IRFET proved to be the best operator outperforming others. The number of contrasts around which the image has to be stretched has been reduced in this implementation to just 12 contrasts for Harris-IRFET and to 9 contrasts for Hessian-IRFET. Still the method takes 12 or 9 times more time than the actual algorithm that is incorporated which in this case is the multi-scale Harris/Hessian operators. Since the algorithm is highly parallel the GPU implementation can be used to speed up the detection process. Still a lot of optimization can be done in the algorithm and the implementation to improve the execution speed. The algorithm fails to perform well under high affine variations. It is due to the limited number of scales used and the algorithm is not affine invariant by itself because of its symmetric derivative and integration kernels. In future improvements the algorithm can be a preprocessing stage for the affine invariant algorithm like those discussed in [15] and thus making it illumination and affine invariant feature extraction operator. Also the scale selected during the detection is not illumination invariant and so future work can be focused on scale selection procedure so that the scale found is illumination invariant.

# REFERENCES


1. Baumberg, A. (2000). Reliable feature matching across widely separated views. *Computer Vision and Pattern Recognition*, (pp. 774-781).

2. Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding , 110* (3), 346-359.

3. Bright Gmh & Co. KG. (n.d.). *News*. Retrieved September 2012, from Bright Tradeshow: brighttradeshow.com/en/news.html

4. Canny, J. F. (1983). *Finding Edges and Lines In Images.* Massechusetts Institute of Technology, Artificial Intelligence Laboratory.

5. Dufournaud, Y., Schmid, C., & Horaud, R. (2000). Matching Images with Different Resolutions. *IEEE*, *1*, pp. 612-618.

6. Gevrekci, M., & Gunturk, B. K. (2008). Illumination Robust Feature Extraction Transform. *Computer Vision and Image Understanding* , 565-571.

7. Harris, C., & Stephens, M. (1988). A Combined Corner and Edge Detector. *Alvey Vision Conference*, (pp. 147-151).

8. Honda. (n.d.). Retrieved September 2012, from Asimo - The World's Most Advanced Humanoid Robot: asimo.honda.com

9. Khronos OpenCL working group. (2009, June). The OpenCL Specification. (A. Munshi, Ed.)

10. Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision , 30* (2), 79-116.

11. Lindeberg, T., & Garding, J. (1997). Shape-Adapted Smoothing in Estimation of 3-D Depth Cues from Affine Distortions of Local 2-D Brightness Structure. *Image and Vision Computing , 15* (6), 415-434.

12. Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal on Computer Vision , 2* (60), 91-110.

13. Matas, J., Chum, )., Urban, M., & Pajdla, T. (2002). Robust wide baseline stereo from external regions. *13th British Machine Vision Conference*, (pp. 384-393).

14. Mikolajczyk, K. (2007, July). *Affine Covariant Features*. Retrieved August 2011, from Robotics Research Group: http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

15. Mikolajczyk, K., & Schmid, C. (2002). An affine invariant interest point detector. *Seventh European Conference on Computer Vision*, (pp. 128-142).

16. Mikolajczyk, K., & Schmid, C. (2001). Indexing based on scale invariant interest points. *Eighth nternational Conference on Computer Vision*, (pp. 525-531).

17. Mikolajczyk, K., & Schmid, C. (2004). Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision , 60* (1), 63-86.

18. Mikolajzyk, K. (2002). *Interest point detection invariant to affine transformations.* Institut National Polytechnique de Grenoble.

19. Moravec, H. P. (1980). *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* Stanford University.

20. S.M.Smith, & J.M.Brady. SUSAN - a new approach to low level image processing. *International Journal of Computer Vision , 23* (1), 45-78.

21. Sebe, N., Gevers, T., Weijder, J., & Dijkstra, S. (2006). Corner Detectors for affine invariant salient regions: Is color important? *CIVR.* Phoenix, USA.

22. Sur, F. (2007). *Invariant Image Descriptors and Affine Morphological Scale-Space.*

# VITA

Prabakar K. Gunashekhar was born in 1987 in Coimbatore, a city in India. He did his schooling in his hometown and joined College of Engineering Guindy, one of the most esteemed institutions in India, during 2005 to pursue his bachelor's degree in Electronics and Communications Engineering. He completed his bachelor's degree in the year of 2009 and later joined Louisiana State University in the year 2010 to pursue his master's degree in Electrical and Computer Engineering from which he will be graduating in December 2012