EFFICIENT FEATURE DETECTION USING OBALoG: OPTIMIZED BOX APPROXIMATION OF LAPLACIAN OF GAUSSIAN

by

VINAYAK REDDY JAKKULA

B.E., Osmania University, India, 2007

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2010

Approved by:

Major Professor
Dr. Chris Lewis

Copyright

Abstract

This thesis presents a novel approach for detecting robust and scale invariant interest points in images. The detector accurately and efficiently approximates the Laplacian of Gaussian using an optimal set of weighted box filters that take advantage of integral images to reduce computations. When combined with state-of-the art descriptors for matching, the algorithm performs better than leading feature tracking algorithms including SIFT and SURF in terms of speed and accuracy.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BA - Bundle Adjustment

CenSurE - Center Surround Extremas

DoB - Difference of Boxes

DoG - Difference of Gaussian

DTM - Digital Terrain Model

HiRISE - High Resolution Imaging Science Experiment

KLT - Kanade-Lucas-Tomasi

LoG - Laplacian of Gaussian

M-SURF - Modified - Speeded Up Robust Features

MU-SURF - Modified Upright - Speeded Up Robust Features

NNDR - Nearest Neighbor Distance Ratio

OBALoG - Optimized Box Approximation of Laplacian of Gaussian

RANSAC - Random Sample Consensus

SIFT - Scale Invariant Feature Transform

SSD - Sum of Squared Distances

SURF - Speeded Up Robust Features

Acknowledgments

I would like to take this opportunity to express my gratitude to some important people who have inspired me to complete this work. Firstly, I thank my Major Professors, Dr.Chris Lewis and Dr.Dale Schinstock, for their continued support and guidance throughout my Masters. I thank them for their patience, for always being supportive and guiding me in the right direction. I am grateful to Dr.Dwight Day for his support and guidance.

I would like to thank Zachary Moratto for his ideas on this topic and also helping me in the implementation. Finally, I would like to thank my family and friends for their unrelenting support and encouragement.

Chapter 1

Introduction

The detection and matching of corresponding features from a sequence of images of a scene is a basic problem for computer vision applications. It arises in object recognition, image registration, panoramic mosaic creation, terrain extraction, and simultaneous localization and mapping. Robust feature matching algorithms find correspondences in images even with various deformations such as scaling and rotation[1]. The leading algorithms first detect features or interest points, then use descriptors to match corresponding features. The Harris Detector[2], SUSAN detector[3] and Kanade-Lucas-Tomasi (KLT) Feature Tracker[4] are classic examples. More recent algorithms such as the Scale Invariant Feature Transform (SIFT)[5], the Speeded Up Robust Features (SURF)[6] and the Center Surround Extrema (CenSurE)[7] algorithms provide both enhanced correspondence accuracy and computational speed for real time applications. Both SIFT and SURF algorithms are protected by either copyrights or by patents that restrict their use in commercial development.

In this thesis an alternate algorithm is developed that is more computationally efficient and equally accurate to these with the intent of providing an unprotected, yet state-of-the art, feature tracking capability. The main contribution is in the detection phase. Similar to the SIFT algorithm, this method approximates the Laplacian of Gaussian (LoG) filter for detecting interest points. The Optimized Box Approximation of the LoG filter(OBALoG), takes advantage of integral images using a set of box filters similar to the Fast approximated SIFT[8], SURF and CenSurE detectors. The box sizes and weights are optimized to improve

accuracy. For the description and matching phase, basic elements from various other implementations are combined. The descriptor vector is similar to that described in CenSurE, but includes an additional orientation step from SURF. The nearest neighbor distance ratio (NNDR) detects matches between images and outliers are rejected using RANSAC[9].

Chapter 2

Background

There has been significant research developing feature detectors that find stable, unique and scale-invariant interest points. One of the first and most common edge detectors is the Laplacian of Gaussian (LoG) filter[10]. It is a combination of low and high pass filters, a Gaussian kernel to eliminate high frequency noise and a Laplacian operator to find edges. Edges are located as zero-crossings of the image formed by convolving the input image with a LoG operator. One of the important characteristics of the LoG operator is that it enables scale space manipulations. Corresponding edges occur at different scales when one image is captured closer to the feature than the other. Features observed at different scales are matched by applying LoG operators with different standard deviations to each image. The embedded Gaussian filter selects the range of scales over which there is a change in intensity.

Moravec described a method based on the autocorrelation of an image patch to detect corners in images[11]. He used the sum of squared distances (SSD) between the patch around a candidate corner and patches shifted in distance and direction as a criterion to select corners. Later, Harris and Stephens built on this approach by approximating the second order derivatives appearing in the SSD[2]. The weighted SSD is given by

$$S(x,y) = \sum_u \sum_v w(u,v)(I(u,v) - I(u+x, v+y))^2 \tag{2.1}$$

where I is the image, w is the weight and (u,v) define the image patch that is shifted by

(x, y). Approximating $I(u + x, v + y)$ with its Taylor series expansion the SSD becomes:

$$S(x, y) \approx \sum_u \sum_v w(u, v)(I_x(u, v)x - I_y(u, v)y)^2 \tag{2.2}$$

$$= \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.3}$$

where I_x and I_y are partial derivates of I and A is given by,

$$A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.4}$$

Shi and Tomasi provided mathematical proof that the smallest Eigen value of A indicates the corner response[4]. Instead of computing the Eigen values directly, Harris defined the corner response as:

$$C = \det A - k(trace(A))^2 \tag{2.5}$$

where k is a tunable sensitivity parameter. Similarly, the Hessian can be defined with the second partial derivatives as

$$H = \sum_u \sum_v w(u, v) \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{2.6}$$

When $w(u, v)$ is a circular Gaussian window the trace of H is the LoG filter.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{2.7}$$

Many of the recent algorithms make use of integral images to speed up computations[12]. An integral image (also known as a summed-area table) is an algorithm for quickly generating the sum of values in a rectangular subset of a grid. For an image I, the integral image can be created efficiently in a single pass over the image. The value at any point (x, y) in the integral image is just the sum of all the pixels above and to the left of (x, y) in the original image. As, shown in Fig. 2.1, once the integral image has been computed, any rectangular area sum can be evaluated in constant time with just four references.

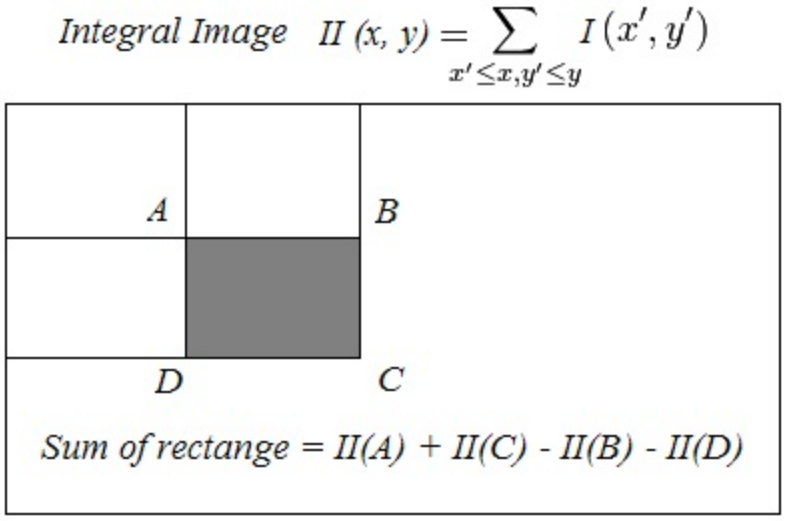


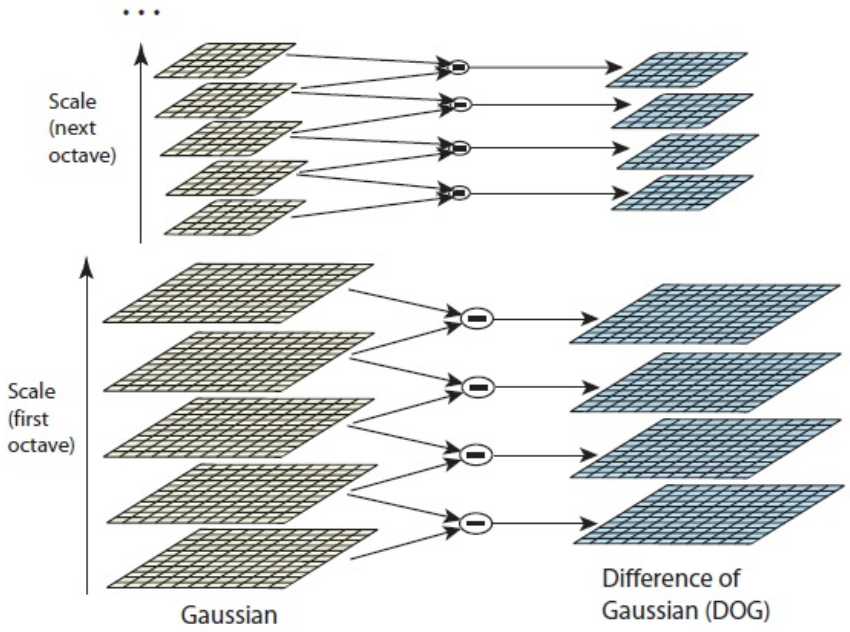
Figure 2.1: *Each point in the integral image is the sum of pixels above and left of the that point in the original image. Once integral image is computed, the sum of a rectangular region of any size can be computed in just four additions.*

2.1 SIFT

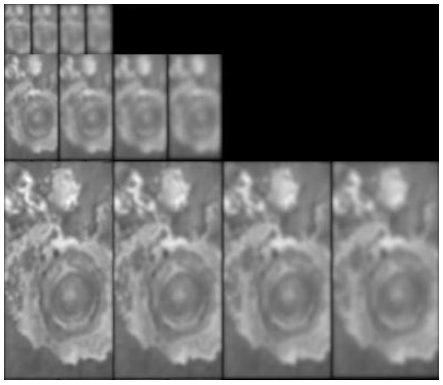
Lowe proposed the Difference of Gaussian (DoG) approach for approximating the LoG filter efficiently[5]. The input images are convolved with Gaussian filters at different scales. Scale-space is formed as the difference between consecutive filter responses, as shown in Fig. 2.2 . Stable extrema across the images and across scale are selected as interest points. However, at higher scales in the scale-space, the images are sub-sampled to save computation time with the loss of location accuracy for interest points.

2.2 SURF

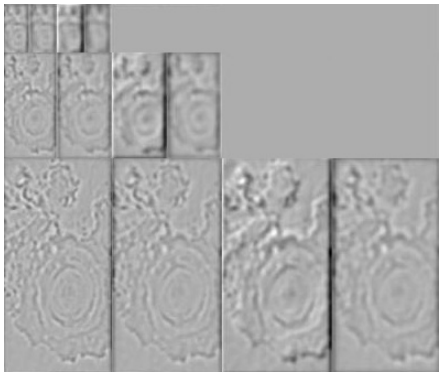
In 2006, Bay et al. proposed the SURF algorithm which extracts interest points using an approximation to the determinant of the Hessian[6]. Instead of constructing the Hessian using Gaussians and second order partial derivatives, Bay et al. approximated this operation with encapsulated rectangular box filters, as shown in Fig. 2.3. Rectangular filters are efficiently evaluated using integral images, because the convolution of the original image



(a)



(b)



(c)

Figure 2.2: *Scale space setup of SIFT(a)[5] (Figure from SIFT paper by D.Lowe). Gaussian image pyramid (b) and the Difference of Gaussian pyramid (c) of SIFT on a sample image.*

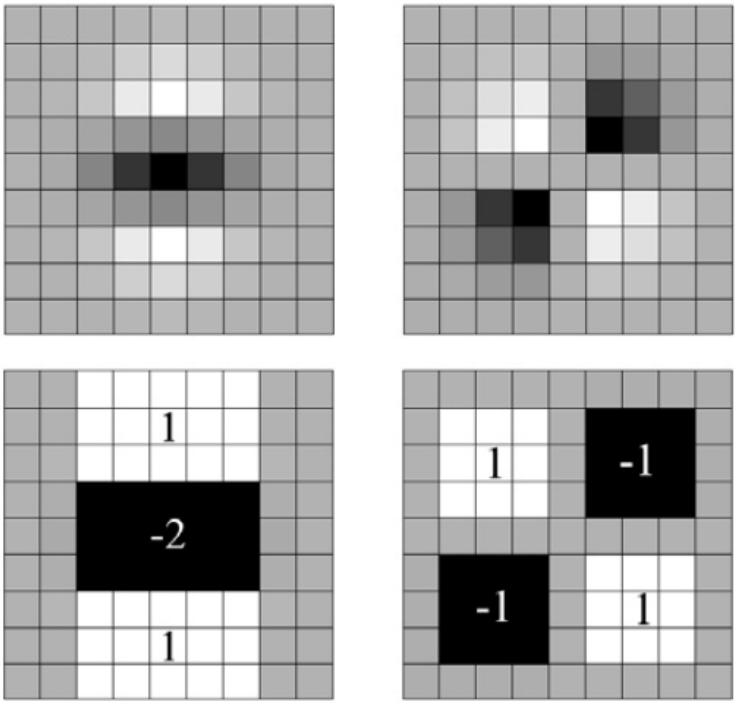


Figure 2.3: *Gaussian second order partial derivatives in y-direction and xy-direction(top), and the corresponding SURF box approximations(bottom)*[6] *(Figure from SURF paper by Bay et.al).*

with a rectangular box filter is found with just four additions, irrespective of the box filter's size.

2.3 CenSurE

The CenSurE algorithm developed by Agarwal et al. uses a Difference of Boxes(DoB) approach to approximate the LoG[7]. Here again, the usage of box filters and integral images speeds up computation. Fig. 2.4 shows the 2 box filters used to build the scale space in CenSurE algorithm.

This thesis presents a detector which also approximates the LoG with a set of overlapping box filters (Fig. 2.5), but which does so more accurately than previous methods and remains computationally efficient.

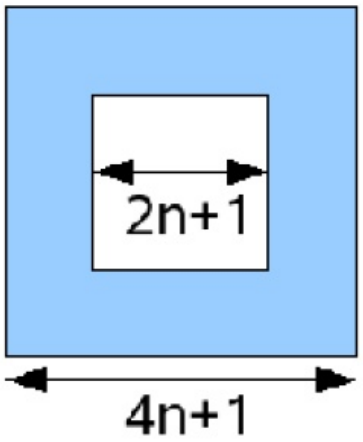
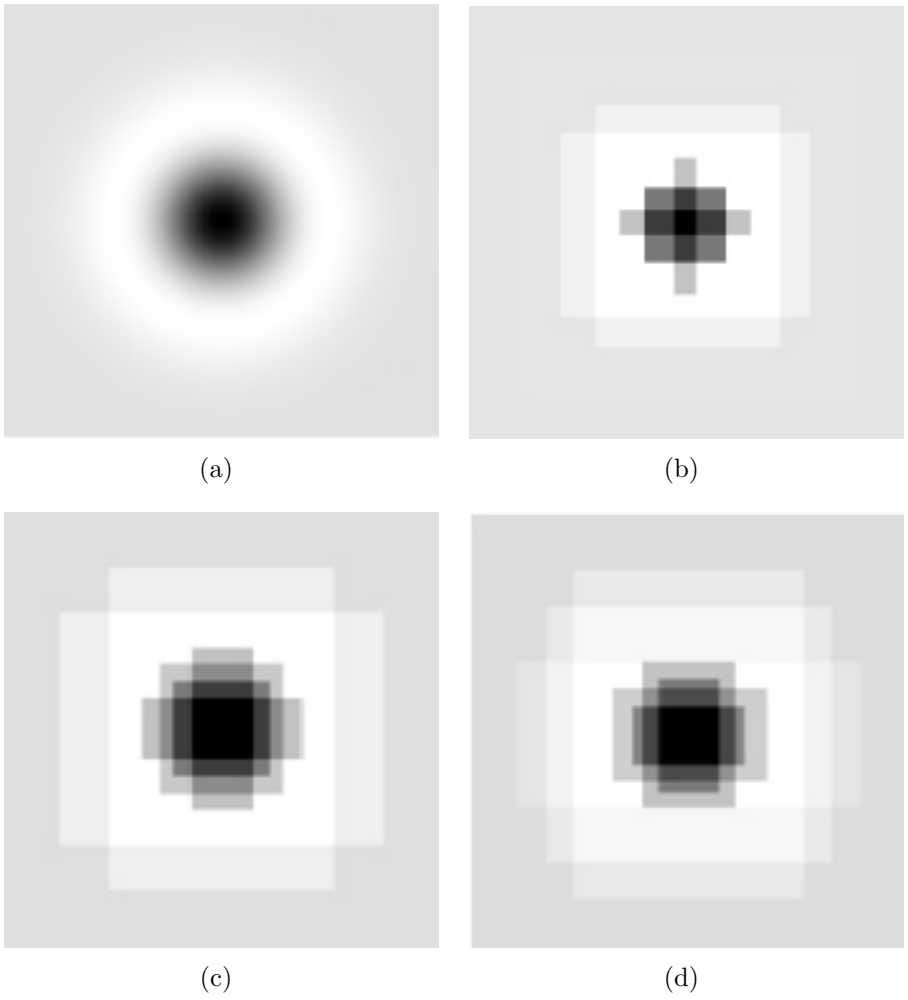


Figure 2.4: *CenSurE Difference of Boxes Approach*[7] *(Figure by Agarwal et.al).*



(a) (b)

(c) (d)

Figure 2.5: *LoG operator (a) approximated using 6 (b), 7 (c) and 8 (d) weighted boxes.*

Chapter 3

OBALoG approach

The box filters of OBALoG are a discrete approximation of the truncated LoG filter[13] shown in Fig. 3.1, described by

$$L(x, y, \sigma) = \frac{1}{2\pi\sigma^4}\left(2 - \frac{x^2+y^2}{\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$
$$\forall(x, y) \in [-3\sigma, 3\sigma] \tag{3.1}$$

The value of σ determines the scale of features. Convolving an $N \times N$ LoG filter, Fig. 2.5(a), over an image patch at one pixel location takes N^2 computations. For feature correspondence, LoG filters of various scales are applied to all pixels in both images, and all interest points identified in one image are compared to those from the other. Multi-pass LoG filtering requires excessive computation. Instead, the LoG is approximated with a sufficiently large set of overlapping weighted boxes, Fig. 2.5(b)- 2.5(d), but with a set small enough to remain computationally efficient. Determining a sufficient number of boxes and their weights is fundamental to replicating the response of the truncated LoG filter.

3.1 Determining the number of boxes

To estimate the required number of summing boxes, the one dimensional LoG function was sampled and its frequency content examined[14]. The LoG function's Discrete Fourier transform is shown in Fig. 3.2. The majority of the frequency content occurs below 1

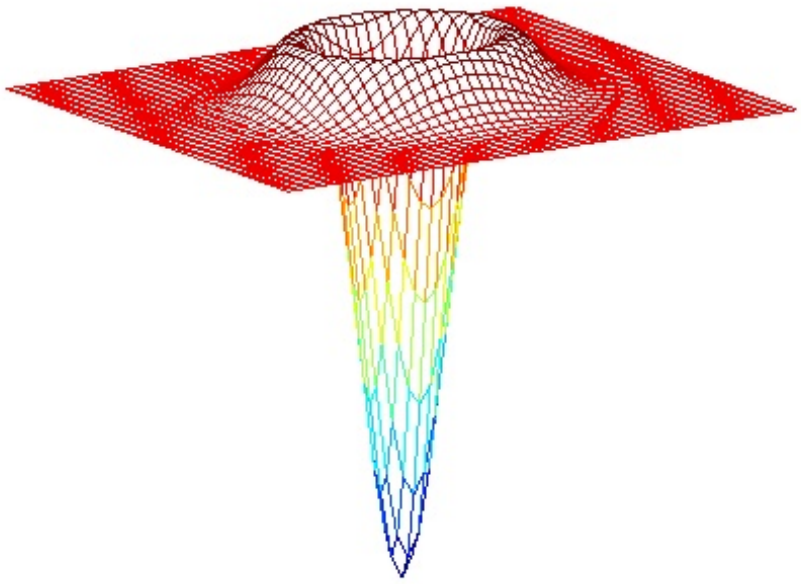
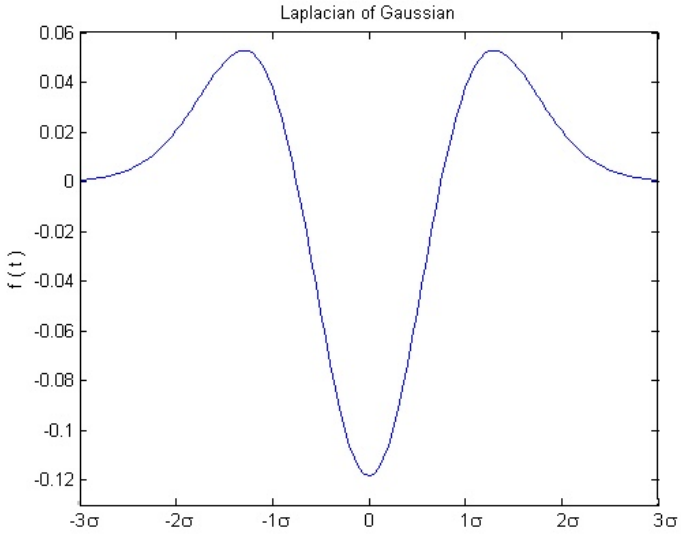


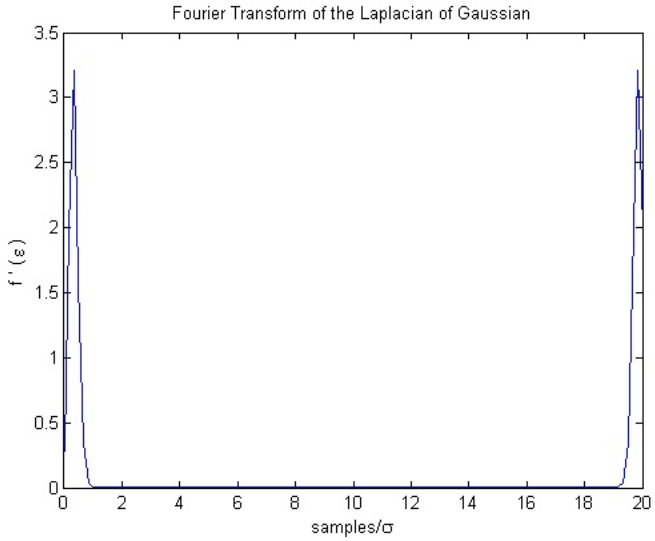
Figure 3.1: *2-D Laplacian of Gaussian*

sample per σ, indicating a sample rate of at least 2 samples per σ to accurately represent the frequency content of the curve(Fig. 3.2(b)). Due to symmetry, it is sufficient to use 6 boxes to approximate the LoG over 6σ, because this results in a sampling frequency of 2 samples per σ. Since very little frequency content is above one sample per σ, increasing the number of boxes beyond 6 only marginally improves accuracy. This analysis implies that CenSurE's DoB approach with only two boxes cannot accurately represent the frequency response of the LoG.

To confirm that 6 boxes suffices, an experiment was done with different numbers of boxes where in each case the box sizes and weights were optimized to best represent the LoG operation. Figure 3.3 plots the error as the number of boxes increases. The error is the sum of squared difference between the truncated discrete Laplacian of Gaussian and the box approximation. As expected, only marginal reduction in error occurs for N greater than 6. With 6 boxes being applied regardless of the scale and taking advantage of integral images, the OBALoG filter requires only 24 additions and 6 multiplications for each scale and pixel



(a)



(b)

Figure 3.2: *One dimensional LoG function (a) and its frequency plot (b) demonstrate that the majority of the frequency content lies below 1 sample per* σ*.*

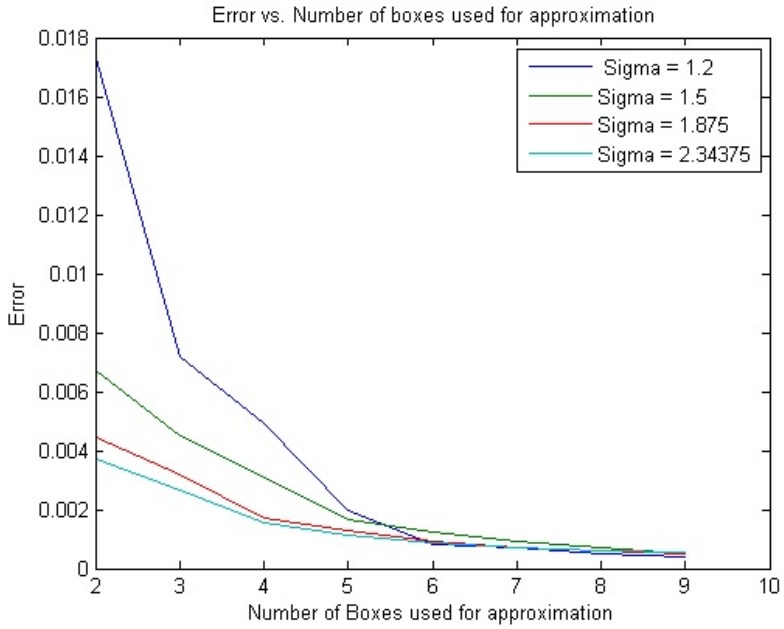


Figure 3.3: *This figure shows that there is a marginal improvement in error with more than 6 boxes.*

location. The true Log filter truncated to $N \times N$ requires N^2 computations.

3.2 Building the box filter approximations

For symmetry, the LoG filter size must be an odd number of pixels. Three pixel by three pixel filters only permit 4 possible box sizes, which are 3×3, 3×1, 1×3 and 1×1. Larger filters have many more combinations, however, filters with $\sigma < 1$ pixel produce features too small to be uniquely identified. To find the best set of box sizes and weights to approximate the LoG, for each filter size all possible combinations of box heights and widths were enumerated. The weights of each combination were then determined by minimizing the sum of squared differences with the LoG filter,

$$C = (\ OBALoG(w) - LoG(\sigma)\ )^2 \tag{3.2}$$

where OBALoG is the impulse response of the set of boxes parameterized by a vector of weights w. For a given filter scale, the combination of box sizes and weights resulting in

the smallest value of C is chosen. At larger scales, enumerating every possible box size becomes computationally intractable. For larger scales, the values are linearly extrapolated and round up to the nearest odd number to determine the optimal box dimensions. The weights are selected by minimizing C. The optimal box sizes extrapolate with the same ratio as the change in scale, as seen in Table 3.1. For example, as sigma scales from 1.2 to 11.175, the largest box size scales by the same factor from 11×11 to 103×103. To ensure optimality, the residual error values of the extrapolated scale and its nearest odd numbered scales can be compared.

Table 3.1: *Optimized box sizes and weights for various scales of OBALoG filters*

Sigma	Box Dimensions	Box Weights	Residual Error
$\sigma = 1.2$	[11,11][7,5] [5,7][3,1] [3,3][1,3]	0.000618,0.008284 0.008284,-0.05576 -0.035566,-0.05576	0.000818
$\sigma = 1.5$	[13,13][9,7] [7,9][5,1] [3,3][1,5]	0.000266,0.004890 0.004890,-0.02214 -0.048850,-0.02214	0.001237
$\sigma = 1.875$	[17,17][11,9] [9,11][5,1] [3,3][1,5]	0.000279,0.001994 0.001994,-0.016285 -0.034734,-0.016285	0.001522
$\sigma = 2.3437$	[21,21][13,11] [11,13][7,3] [7,7][3,7]	0.000219,0.002006 0.002006,-0.01499 -0.000824,-0.01499	0.002064
Extrapolating for larger scales			
$\sigma = 2.9296$	[27,27][17,13] [13,17][9,5] [9,9][5,9]	0.000103,0.001378 0.001378,-0.009607 0.002226,-0.009607	0.002398
$\sigma = 3.6621$	[33,33][21,17] [17,21][11,5] [11,11][5,11]	0.000072,0.000826 0.000826,-0.006072 -0.000001,-0.006072	0.002621
$\sigma = 4.5776$	[41,41][27,21] [21,27][13,7] [13,13][7,13]	0.000038,0.000525 0.000525,-0.003886 0.000282,-0.003886	0.002939
$\sigma = 5.7220$	[51,51][33,27] [27,33][17,7] [17,17][7,17]	0.000029,0.000329 0.000329,-0.002446 -0.000271,-0.002446	0.003311
$\sigma = 11.175$	[103,103][65,47] [47,65][27,9] [27,27][9,27]	0.000010,0.000084 0.000084,-0.00056 -0.000470,-0.00056	0.009576

Chapter 4

Implementation of the OBALoG interest point detector

The OBALoG interest point detection process is shown in Fig. 4.1. First, all interest points are found at a variety of scales using the OBALoG filters applied to every pixel. Then extrema are selected using thresholding and Non-Maximal Suppression[5–7] to localize features across scale-space, eliminating duplicate occurances of features at different scales. These extrema are filtered using a scale-adapted Harris measure selecting corners and blobs but rejecting poorly localized edge or line features. Finally, all interest points found are assigned an orientation and characterized with the descriptor from M-SURF for matching between images.

C++ has been chosen as the programming language to develop OBALoG. NASA Vision Workbench(VW) is a requirement for OBALoG. Vision Workbench is a library of C++ functions that provides functionality for many image processing needs like reading data from image files and filtering images using different filters. Each stage in the OBALoG interest point detection process is implemented as a different function which is called from OBALoG's main function(See Appendix B). More details about each stage are described in the following sections.

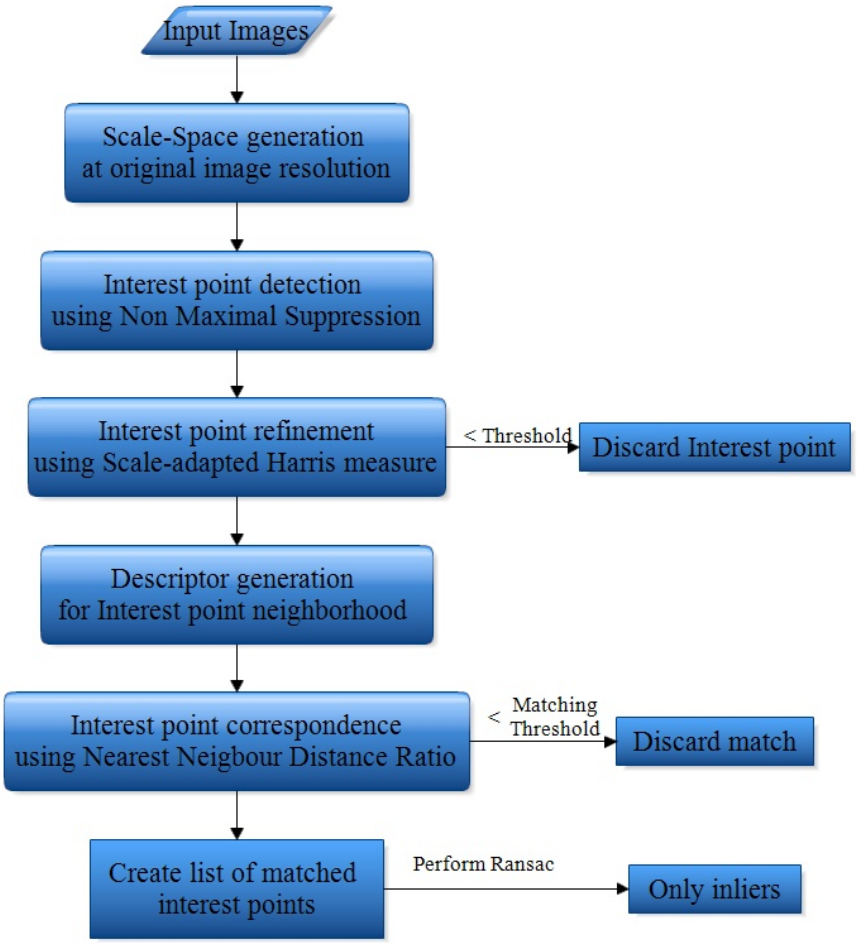


Figure 4.1: *Flowchart of OBALoG interest point detection and correspondence process*

4.1 Scale space setup

To find features at a variety of scales, OBALoG filters are constructed starting with $\sigma = 1.2$ and incremented through the first 8 scales shown in Table 3.1. Because optimal box sizes are determined through extrapolation and the box weight refinement is straight forward, any desired scale could be easily added. However, according to Bay et al., the number of features detected reduces rapidly at larger scales, as shown in Fig. 4.2. Based on that, the OBALoG scale-space was constructed with just 8 scales which find a sufficient number of large scale features.

The scale-space of OBALoG was implemented as an array of image structures having the same size of the original image. For each level, depending on the scale, the corresponding predefined box filters are loaded by calling the "DesignBoxes" function. Using an integral image, the "ProcessWithBoxes" function convolves the box filters and returns the filtered images. The implementation has an added option to view the normalized filter responses at each scale.

```
// SCALE SPACE SETUP
   // Iterate through each scale in the scale space
   for ( unsigned scale = 0; scale < 8; scale++ ) {
      // Load the predefined box filters corresponding to the scale
      BoxFilter Boxes = DesignBoxes(integral,scale);
      // Convolve the filters and save the returned filtered images
      interest_data.push_back( ProcessWithBoxes(integral,Boxes) );
    }
```

4.2 Thresholding and Non-Maximal Suppression

To construct the scale-space, an integral image is created for the input image, and then filtered at each scale as described in the previous section. Then, extrema (both maxima

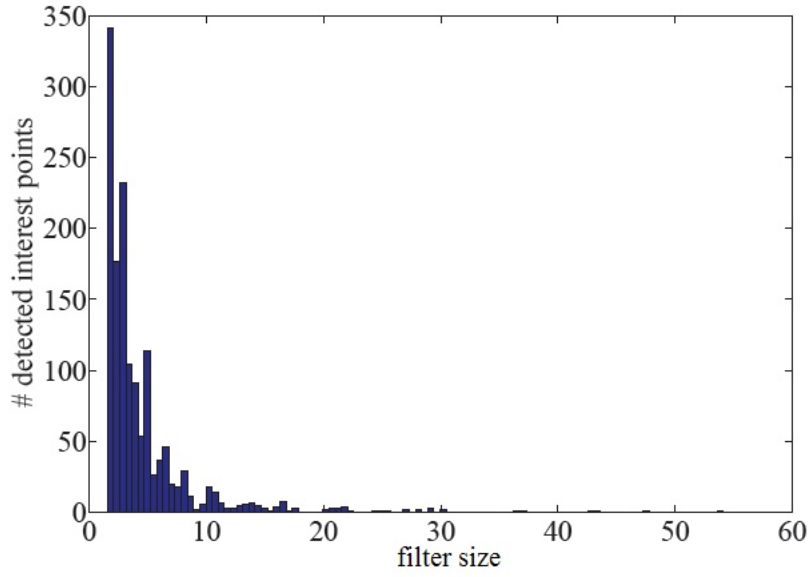
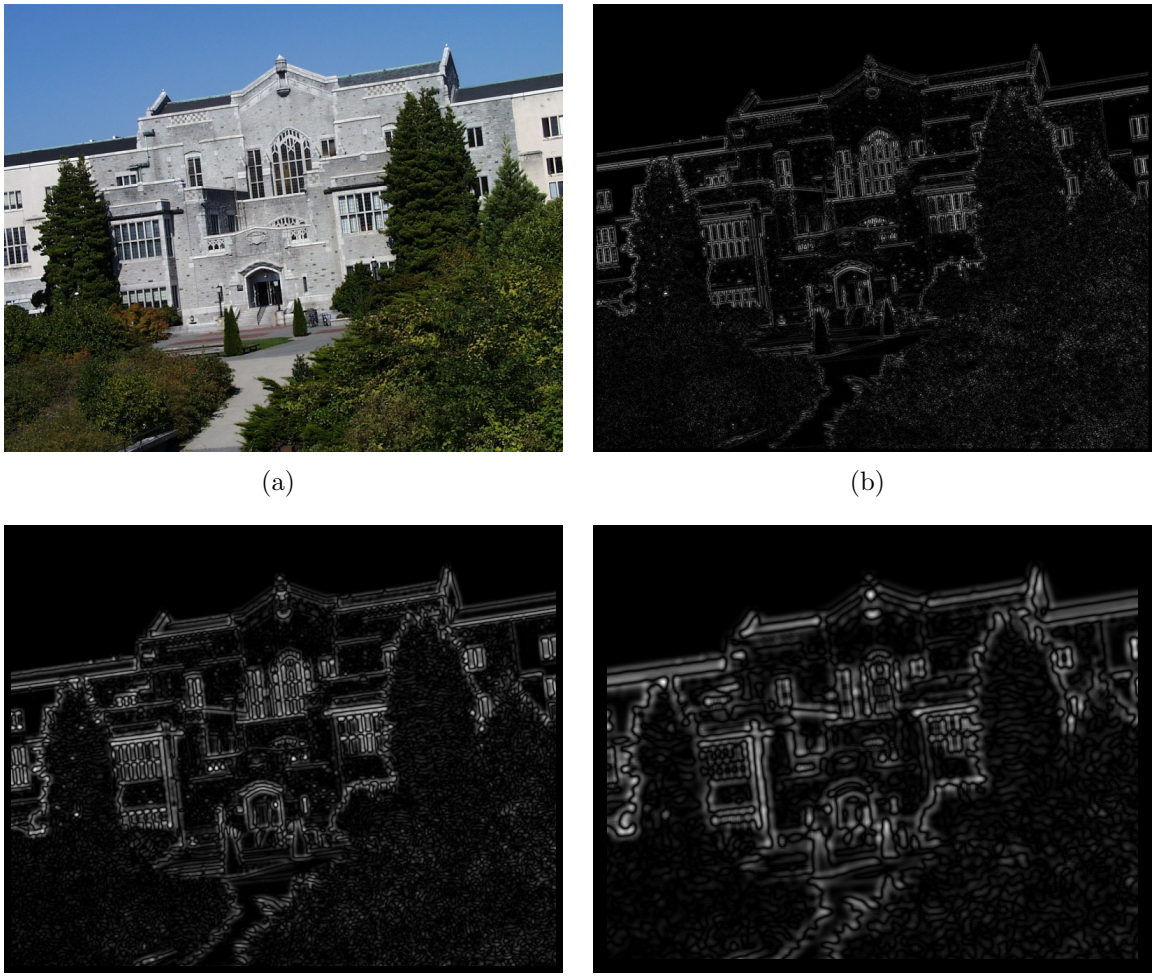


Figure 4.2: *This is a typical histogram from a single image which shows that the number of detected features decays very quickly with increase in filter size*[6]*. Thus, OBALoG scale-space stops at* $\sigma = 5.72$ *corresponding to a filter size of 51.*

and minima) of the filtered images are taken as interest point candidates (Fig. 4.3(b)) using a tunable threshold on the OBALoG response strength to eliminate weak features. This threshold can be adjusted to select the required number of features. A high threshold outputs fewer interest points compared to a lower threshold as seen in Fig. 4.4.

At higher scales, sub-sampling may be employed to further reduce computation, but it is not done in our implementation of OBALoG. For refining a feature's scale and for accurate localization of features found in sub-sampled images using non-maximal suppression, subpixel refinement would need to be applied[5,6].

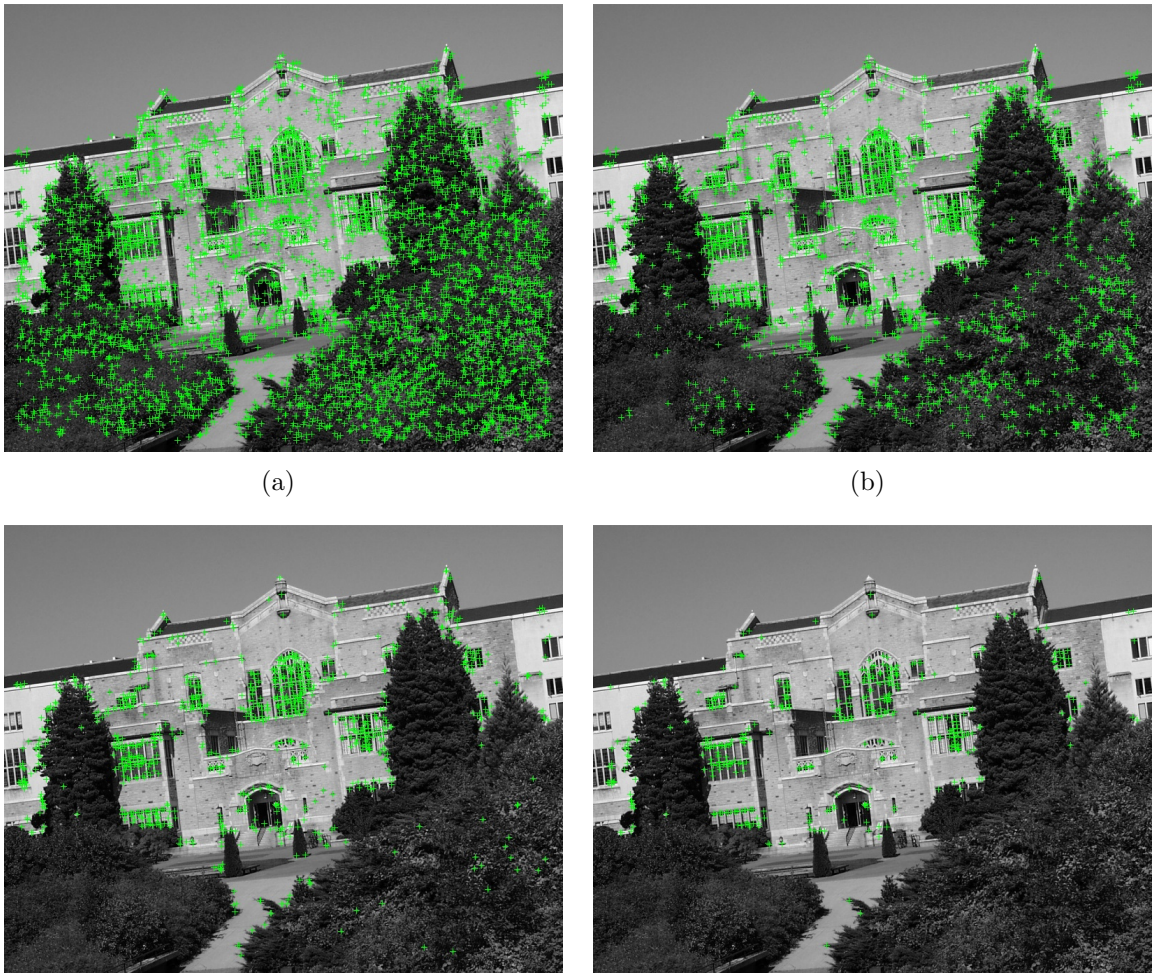
Non-maximal suppression is applied to the candidate features from thresholding to localize them as interest points. Figure 4.5 shows a graphical representation of non-maximal suppression, where extrema pixels are localized as interest points. The maxima and minima across scales are detected by comparing a pixel (marked as X) to its 26 neighbors in $3 \times 3 \times 3$ regions (marked with circles)[5–7]. A candidate point is kept if it is the maximum in this set. During implementation, Non-maximal suppression uses the absolute values of the



(a) (b)

(c) (d)

Figure 4.3: *Interest point detection using OBALoG filters. The input image (a), the normalized OBALoG filter responses corresponding to* $\sigma = 1.2$*(b),* $\sigma = 2.92$*(c) and* $\sigma = 5.72$*(d). Normalization is achieved by taking the absolute values of the pixels and scaling them to the range [0,255]*



(a) (b)

(c) (d)

Figure 4.4: *Interest points after non-maximal suppression using a low threshold (a) and after removing weak interest points using Harris method (b). Interest points after non-maximal suppression with a higher threshold (c) and after removing weak interest points using Harris method (d).*

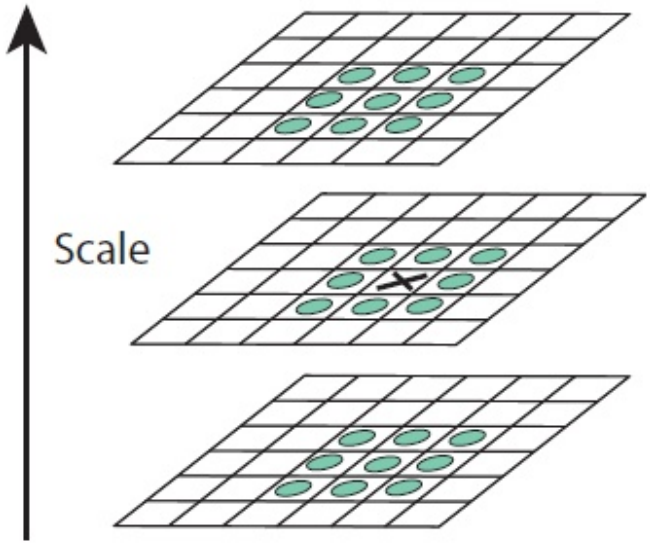


Figure 4.5: *Interest point localization using Non maximal suppression*[5].

pixels and checks only for maxima rather than maxima and minima in the filtered images. The function "is_3_3_3_max" checks if a particular pixel is a maximum or not.

```
// NON MAXIMAL SUPPRESSION
      // Iterate over the pixels in the filtered images
     for ( int ii = 1; ii < interest_data[0].cols() - 1; ii++ )
          for ( int j = 1; j < interest_data[0].rows() - 1; j++ )
          for ( int k = 1; k < interest_data.size() - 1; k++ )
              // Check for extrema in a 3x3x3 neighborhood
              // Also check the threshold condition
              if ( is_3_3_3_max( interest_data, ii, j, k, threshold ) )
            // If extrema is found:save it as interest point candidate
            interest_pts.push_back( InterestPoint(ii,j,truescale[k],
                                    interest_data[k](ii,j)));
```

4.3 Line Suppression

Interest points found by the LoG filter may be edges, corners, or blobs. Edges can be poorly localized features because a point on an edge looks similar to other points near it. The Scale adapted Harris[15] method is employed to remove these poorly localized edges (Fig. 4.4(b)). Stable interest points have significant curvatures in both directions. Edges have large principle curvature along their line but little curvature perpendicular to it. The trace and determinant of the second moment matrix, H, are used to compute the ratio of principal curvatures. A threshold of 12 sufficiently suppresses edge responses. This threshold value was experimentally chosen based on the edge suppression methods used by Lowe D. and Agarwal et al.

$$\frac{Tr(H)^2}{\det(H)} = \frac{(\Sigma L_x^2 + \Sigma L_y^2)^2}{\Sigma L_x^2 \Sigma L_y^2 - (\Sigma L_x L_y)^2} < 12 \tag{4.1}$$

Here L_x and L_y are defined in Eq. 2.4.

The "IsHarrisCorner" function applies Equation. 4.1 to the remaining set of potential interest points after non-maximal suppression to arrive at a final set of interest points. All potential interest points not satisfying Equation. 4.1 are removed from the set. The following code listing shows the implementation

```
// SCALE-ADAPTED HARRIS METHOD
    // Check if every potential interest point is a harris corner
    for (InterestPointList::iterator it = interest_pts.begin();
                            it != interest_pts.end(); it++ )
        if (!IsHarrisCorner(interest_data, (*it).scale,
                              (*it).ix, (*it).iy )) {
              // If not, erase that interest point
              it = interest_pts.erase(it);it--;
          }
```

Figure 4.6: *Haar wavelet filters to compute responses in x (left) and y direction(right). Black regions have weight of 1 and white regions have -1. Each wavelet takes just 6 computations when used with integral images*[6].

4.4 Orientation Assignment

To achieve rotation invariance, each detected interest point is assigned a reproducible orientation. The interest point descriptor is extracted relative to this orientation so that it may be found repeatably under varying image conditions. The SURF feature orientation step is used to assign a dominant direction to the OBALoG interest points. The details of the theory behind this are not presented here. However, the basic idea is to find the direction using two steps. The first step is to find a pie shaped region around the interest point where the gradients are large. The second step is to assign a direction using the x and y components of the gradients in this region.

To determine the x and y gradients, the responses of a Haar filter in both the x and y directions(Fig. 4.6) are found for each pixel in a set of pixels inside a circle of radius $6s$ around the interest point (where $s = \sigma$ refers to the scale at which the point was detected). This set of pixels is equally spaced by the distance s. The size of the Haar filter kernel used is $4s \times 4s$. The set of Haar responses are weighted by a Gaussian with standard deviation of $2.5s$, centered at the interest point. This process results in a set of weighted points, each having both an x and y component.

To find the region with large gradients, the circle is broken into overlapping pie shaped regions of angle $\pi/3$ using an increment of 0.5 radians. Then, the square of the sum of x

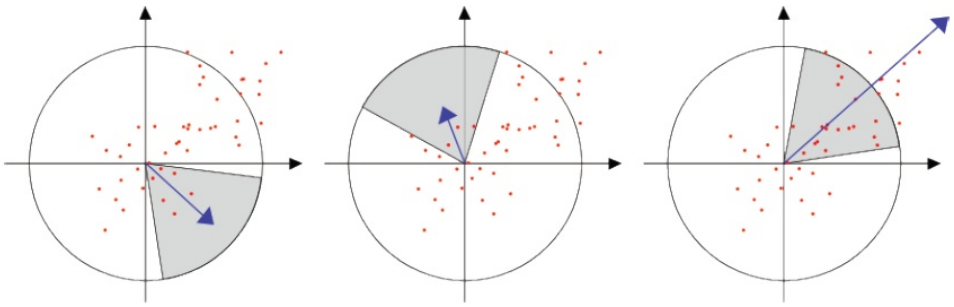


Figure 4.7: *The haar responses in the sliding window are summed to yield the vector shown in blue. The largest blue vector determines the region used to compute the dominant direction of the interest point*[16].

components and the square of the sum of y components are computed for each region and added together for a scalar representation of the magnitude of the gradients in each region. The region with the maximum scalar value is then used to compute the dominant direction of the interest point. This dominant direction is defined by a direction of an x-y vector whose components are the sum of the x components and the sum of the y components in this pie shaped region (Fig. 4.7).

In the implementation, the orientation step is split into two stages. The first stage calculates Haar wavelet responses around the interest point and the second stage iterates over the region to compute the responses for each pie shaped segment. In the first stage, calculate only those responses whose distance is less than 6σ from the center and perform weighting using a Gaussian function. In the second stage, step through a discrete set of angles and compute the sums of the responses to determine the pie shaped region defining the dominant direction. Then compute the dominant direction using this region. The simplified code listing below demonstrates the approach used in the implementation.

```
// ORIENTATION STEP
// Consider a square region of 12s
for ( int x = -6*s ; x <= 6*s ; x+=s ) {
   for ( int y = -6*s ; y <= 6*s ; y+=s ) {
```

```
    // Check whether current point is within circle 6s
    if (x*x + y*y < 36*s*s ) {
    gauss = gaussian(x,y,2*s);
        // Find x & y responses
    resX.push_back(gauss*HHaarWavelet(x,y,4*s));
    resY.push_back(gauss*VHaarWavelet(x,y,4*s));
 }
}
}
```

4.5 Descriptor and Matcher

An M-SURF (Modified SURF) feature descriptor is used to characterize the interest points. It is the MU-SURF descriptor in CenSurE[7](Fig. 4.8), but with orientation accounted for as described in Section 4.4.

The descriptor uses responses from Haar wavelet filters of size $2s$ (s=σ being the scale of the feature) in the horizontal and vertical directions of $24s \times 24s$ square region around the interest point. This region is rotated to align the vertical direction with the dominant direction from the orientation assignment. This square region is divided into 4×4 grid of $9s \times 9s$ sub-regions that have an overlap of $2s$. The Haar filter responses in each sub-region are weighted by a Gaussian with sigma of $2.5s$, centered on the sub-region center. Then a vector for the sub-region is calculated having the components $(\sum dx, \sum dy, \sum |dx|, \sum |dy|)$. Finally, these 16 sub-region vectors are weighted using another Gaussian with sigma of 1.5 sub-regions and combined to form the descriptor vector, as shown in the code listing below. The descriptor is an augmented vector containing all the 64 scalar values in the 16 vectors from the sub-regions. This modified descriptor, according to Agarwal et al., handles the interest point boundaries better that the original SURF descriptor, resulting in improved rotation invariance.

```
// Loop through the square region
for ( int i = 0 ; i < 24*s ; i+=9*s )
 for ( int j = 0 ; j < 24*s ; j+=9*s ) {
  // Inner loop  selects 81 sample points from subregion
  for ( int k = i ; k < i + 9*s ; k+=s ) {
  for ( int l = j ; l < j + 9*s ; l+=s ) {
  // Compute first Gaussian weights
  gauss1 = gaussian(k-18*s , l-18*s , 2.5f*s ) ;
  rx = gauss1*HHaarWavelet(k ,l , 2*s ) ;
  ry = gauss1*VHaarWavelet(k ,l , 2*s ) ;
  // Sum the x,y responses and the their absolute values
  dx += rx ;
  dy += ry ;
  mdx += fabs ( rx ) ;
  mdy += fabs ( ry ) ;
  }
  }
 }
// Weight the vectors with another gaussian of sigma 1.5
// to form the descriptor vector
```

For matching, the nearest neighbor distance ratio (NNDR) method is applied to compare feature descriptors in one image to those in the other image. The distance between a feature in one image and a feature in the other image is defined by the dot product of the two descriptors. This distance is calculated for every feature in the other image. A tunable threshold (usually set to 0.5) on the ratio of the distance to the closest neighbor to that of the second-closest neighbor determines matches (Fig. 4.9(a)). Matching can be sped up by only comparing features which have OBALoG responses of the same sign.

RANSAC is applied to remove outliers and determine the final correspondence list (Fig. 4.9(b)). RANSAC is an iterative algorithm that estimates parameters of a mathematical model from observed data containing outliers or false matches. It eliminates the outliers which do not fit the model[9]. We commonly use an affine transformation as the RANSAC model.

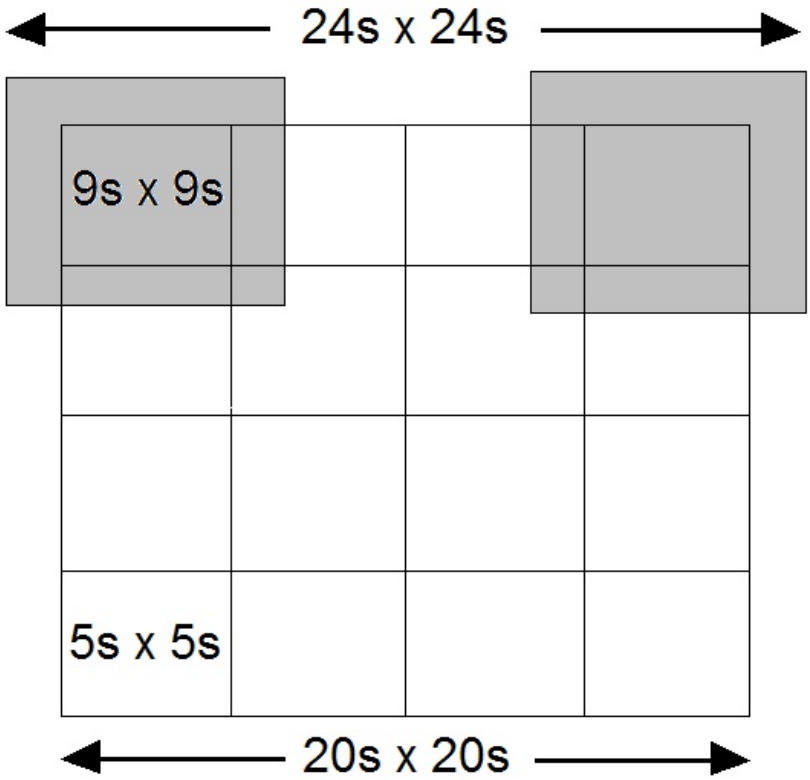
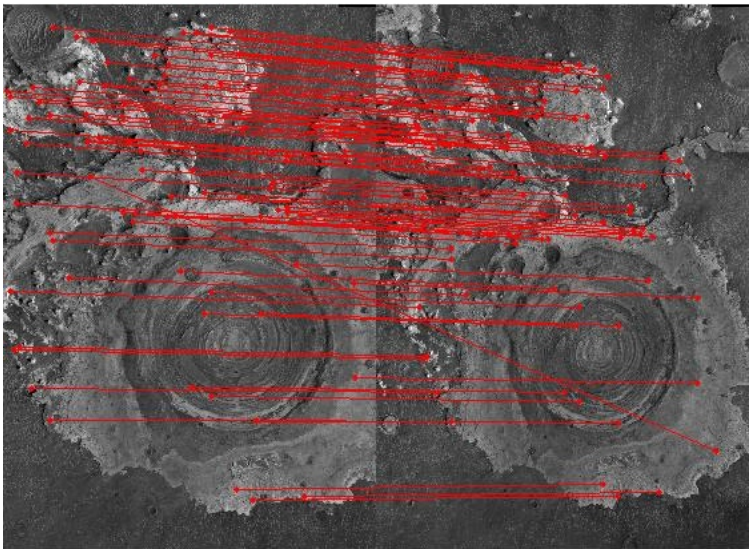
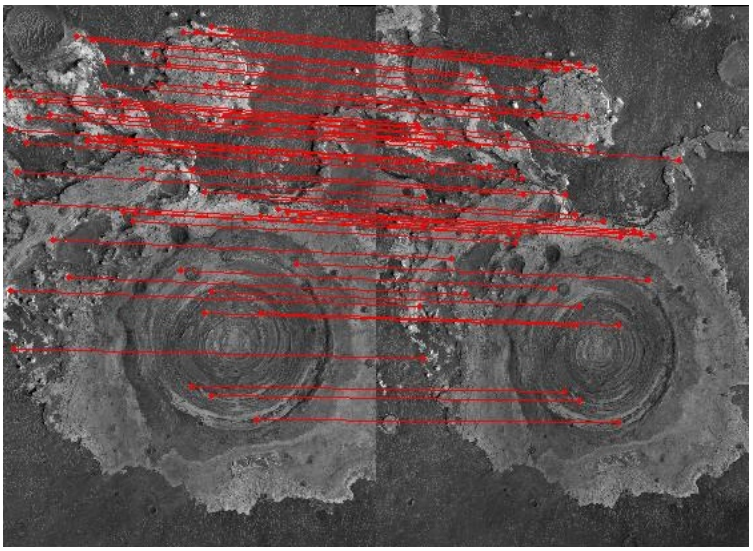


Figure 4.8: *Modified SURF (M-SURF) Descriptor regions and subregions*[7]*. 's' denotes the scale of the feature.*



(a)



(b)

Figure 4.9: *Feature Correspondence.Before RANSAC 4.9(a),After RANSAC 4.9(b)*

Chapter 5

Modified OBALoG for interest point detection in orbiter image pairs

Feature extraction is the precursory step for creating Digital Terrain Models (DTMs). A DTM is a digital representation of the ground surface topography[17]. Photogrammetric preparation of DTMs from overlapping imagery is a two step process. The first step is the refinement of camera's extrinsic parameters using Bundle Adjustment (BA) and the second step is terrain extraction using a dense, pixel by pixel correspondence. BA optimally refines both the 3D structures and camera motions for a given sequence of images[18]. Inputs to BA are the features that correspond in the given images. For DTM creation, it is essential to find features that are abundant, accurate and uniformly distributed throughout the image.

The modified OBALoG implementation subdivides the stereo images into smaller blocks, ensuring that adequate features are found throughout the entire scene. The algorithm also takes advantage of the fact that orbiter image pairs come with known scale and rotational differences, thus improving computational efficiency.

5.1 Problem Statement

Bundle adjustment is a photogrammetric technique to combine multiple images of the same scene into an accurate 3D reconstruction[19]. It does geometric parameter estimation, where the parameters are the 3D feature coordinates, camera poses and calibrations. To get

started, BA needs the initial estimates of camera parameters and the 2D feature locations that match in the imagery. Providing BA with uniformly distributed features improves the precision of the topographic information of the region and results in the creation of a higher quality DTM.

Finding uniformly distributed features in huge images is a challenge in feature detection. Present implementations of SIFT and SURF were not designed to process very large imagery. High processing power and large system memory are required to process such images, usually in the order of 20000 by 70000 pixels. The main issue is that every detected feature in one image is compared to every feature detected in the other. For satellite imagery, there exists a much smaller and measurable region where matches occur. Exhaustive searches are wasted efforts. A high threshold can be applied to select adequate number of features, but only the top features in the entire image are selected. These are usually clustered in the dense feature-rich regions and other sparse regions are neglected. This problem can be overcome by feature detection using block processing. Large imagery is divided into pair-wise blocks and an adaptive threshold is applied that selects features uniformly from both the sparse and feature-rich regions of the image. Large computational power is not needed to process these small blocks of imagery. Also, the pair-wise blocks can be parallel processed to improve the algorithm efficiency.

The original OBALoG feature detection algorithm is optimized so that it is best suited to find correspondence in Orbiter stereo pairs while also overcoming the problem of clustered feature matches and the need for high processing power.

5.2 Optimized feature detection and correspondence using Modified OBALoG

The pre-processing stage in Modified OBALoG algorithm first converts the given imagery into intensity images (gray scale images). Image normalization is performed on these gray scale images. Normalization is the process that changes the range of the pixel intensity

values and its purpose is to bring both the images into a similar range of intensities. It eliminates the variations like noise and illumination which are related to the conditions of image acquisition and are irrelevant to image features. After normalization, the images are subjected to block processing. Block processing allows the big images to be processed as pair-wise blocks. Upon adjusting the appropriate thresholds and selecting the top N feature points in each block pair, uniform detection of interest points throughout the image is ensured.

The scale space of the modified OBALoG algorithm is reduced to take advantage of known scale difference between orbiter images. Instead of comparing features over a wide range of scales as described in Section 4.1, only the required scales are compared to save computation time. Also, custom scales can be added (as shown in Table 3.1) to find specific artifacts of known size. The orientation step is ignored in the modified OBALoG algorithm because orbiter imagery comes with little or no rotation difference.

The results of Modified OBALoG on several Mars orbiter images comparing to results from SIFT and SURF are presented in Section 6.3.

Chapter 6

Results & Discussion

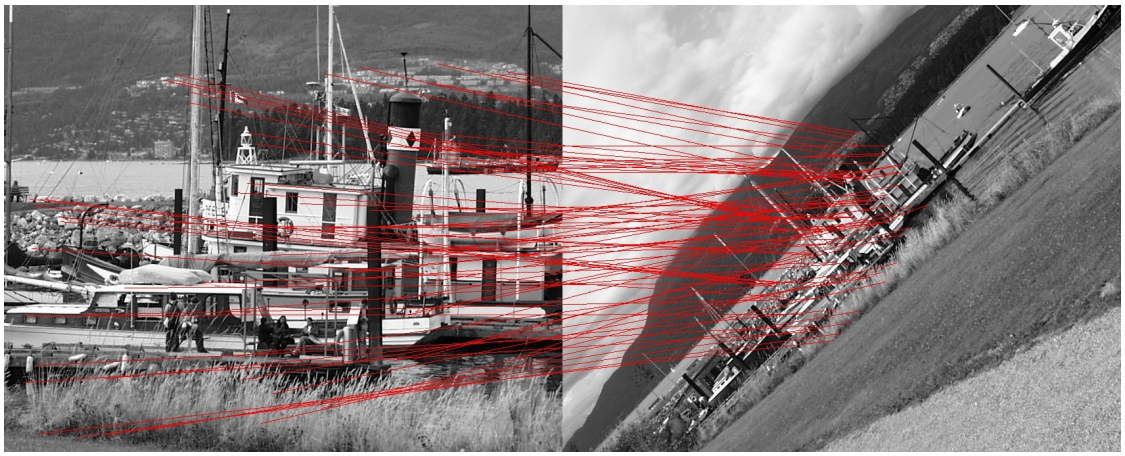
The OBALoG detector is compared with SIFT, SURF, FAST[20], SUSAN and CenSurE detectors for both repeatability and computational speed. To evaluate repeatability, testing was done on the boat, graffiti and wall sequences used in the Mikolajczyk framework[15,21]. There are six images in each sequence. Every sequence depicts an increasing transformation for different conditions like rotation, scale change and viewpoint change. The number of interest points detected by Lowe's original SIFT implementation (executable binary file) was used as the benchmark and appropriate thresholds were adjusted for all other detectors to find the same number of interest points in the common overlapping regions. When evaluating computational efficiency, the execution time of the C++ implementation of OBALoG without compiler optimization was compared to the other five detectors (binary executable files of SIFT,SURF,FAST and C++ implementations of CenSurE,SUSAN were used).

6.1 Repeatability

Repeatability is given by the ratio of repeated features to the detected features. A feature is said to be detected if it is extracted in one and appears in the other image. It is repeated if it is detected nearby in the second image[20]. More details about repeatability are given by Mikolajczyk et al[15,21], and Rosten et al[20]. Figures 6.2, 6.4 and 6.6 demonstrate that the OBALoG detector performs better than the other detectors in most of the cases. However, at very large viewpoint changes, the differences in repeatability are marginal. Figure 6.7

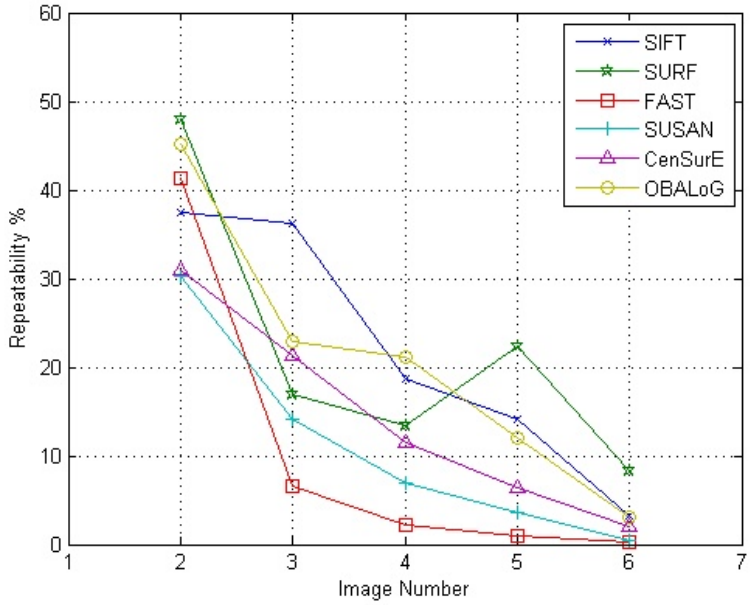
(a)

(b)

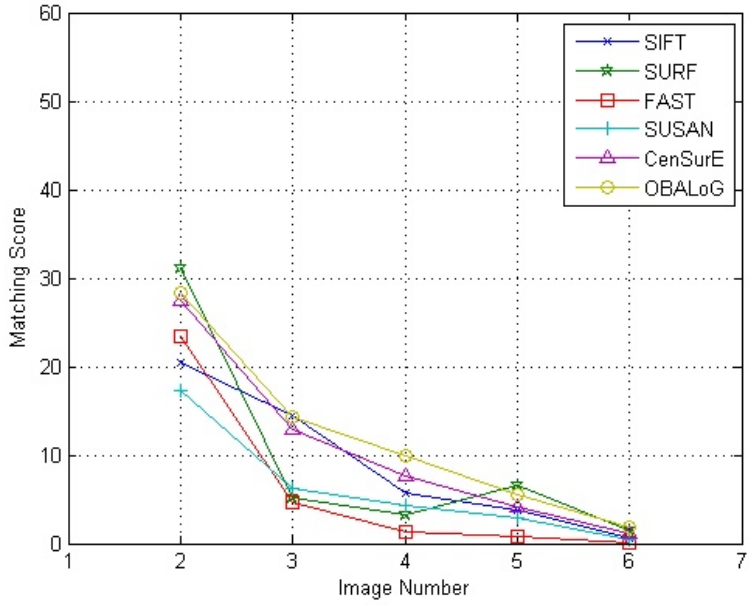


(c)

Figure 6.1: *Feature Correspondence on the boat sequence*



(a)



(b)

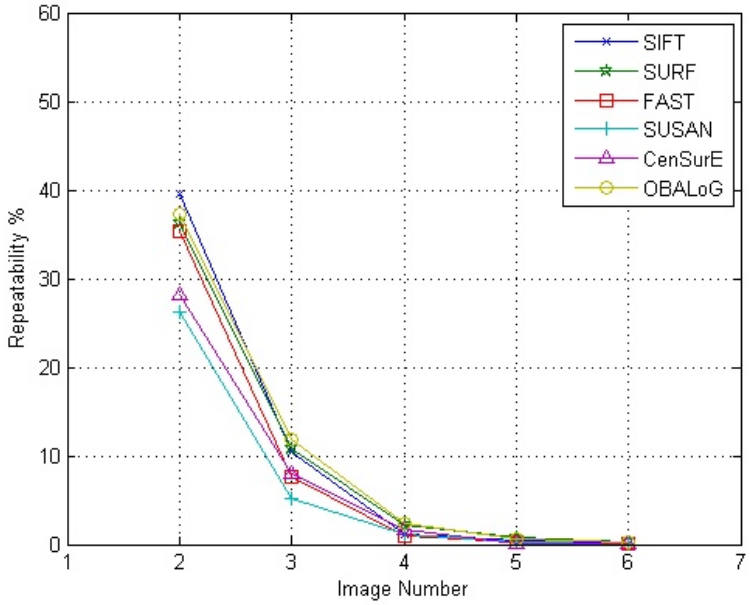
Figure 6.2: *Repeatability results(a) and matching scores(b) for different detectors on the boat sequence*

(a)

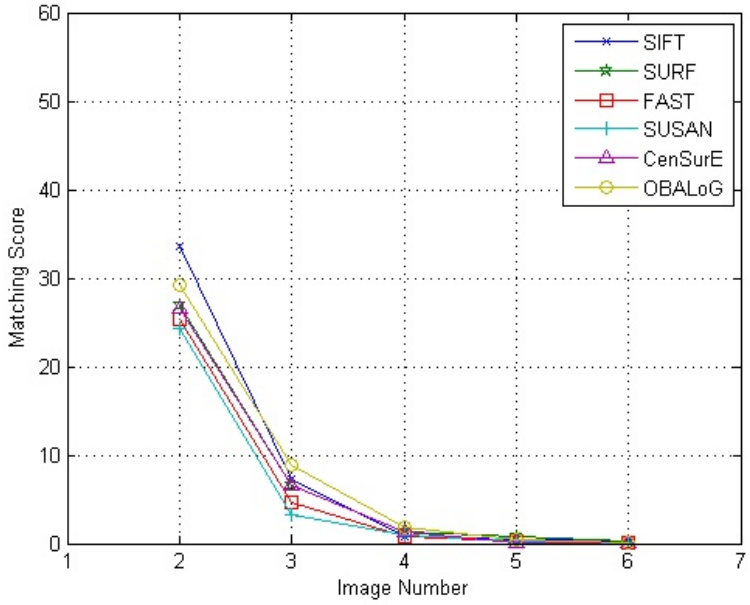
(b)

(c)

Figure 6.3: *Feature Correspondence on the graffiti sequence*

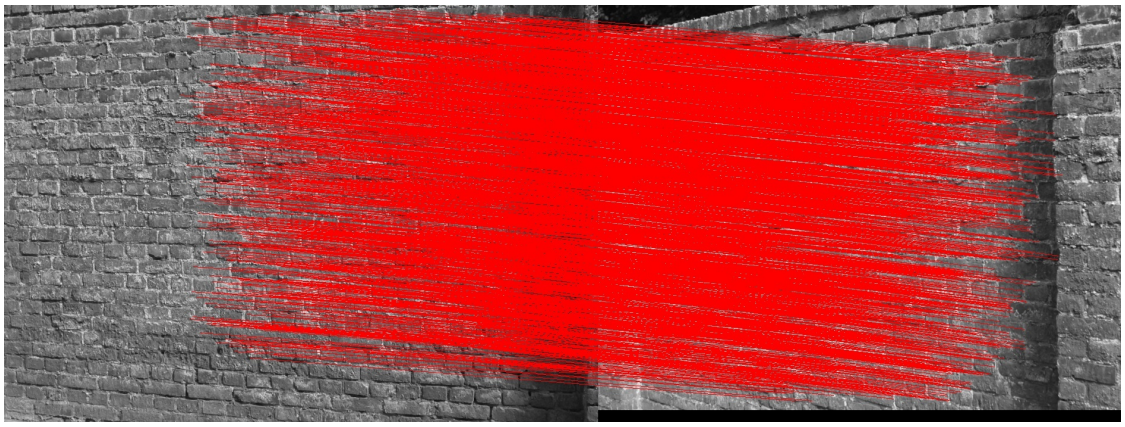


(a)

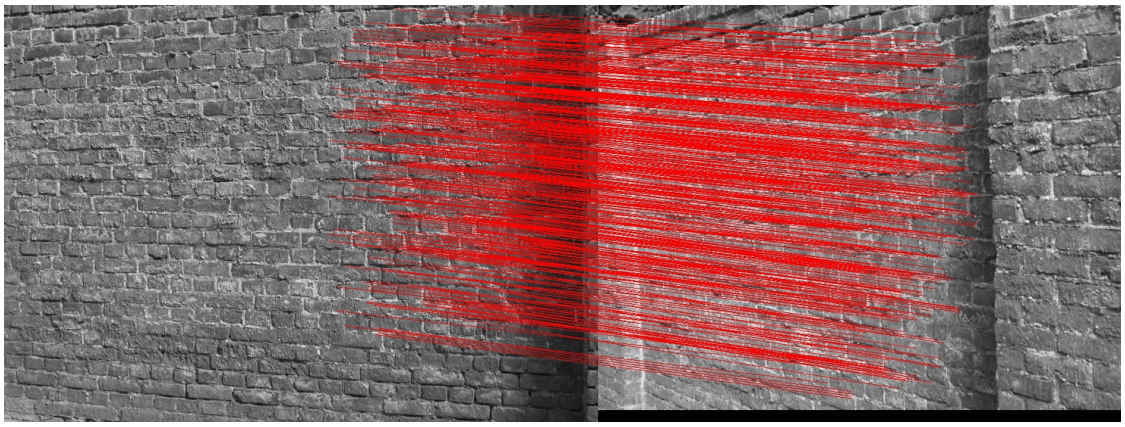


(b)

Figure 6.4: *Repeatability results(a) and matching scores(b) for different detectors on the graffiti sequence*



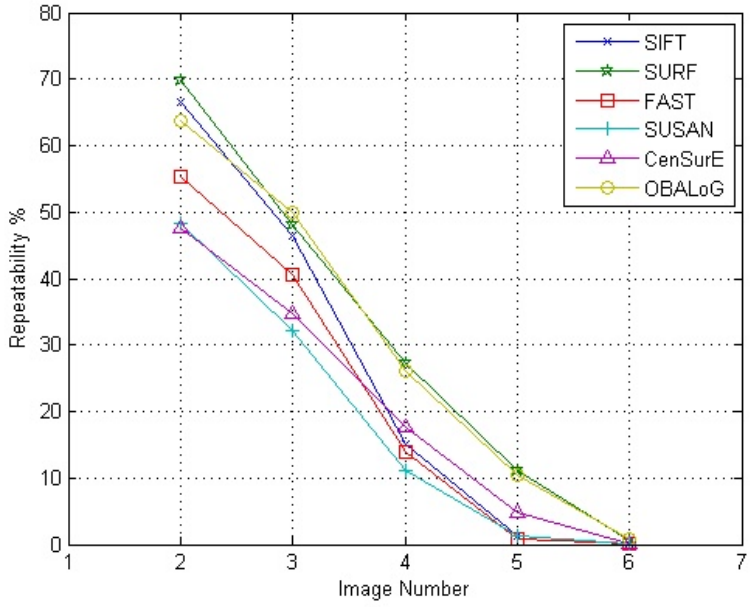
(a)



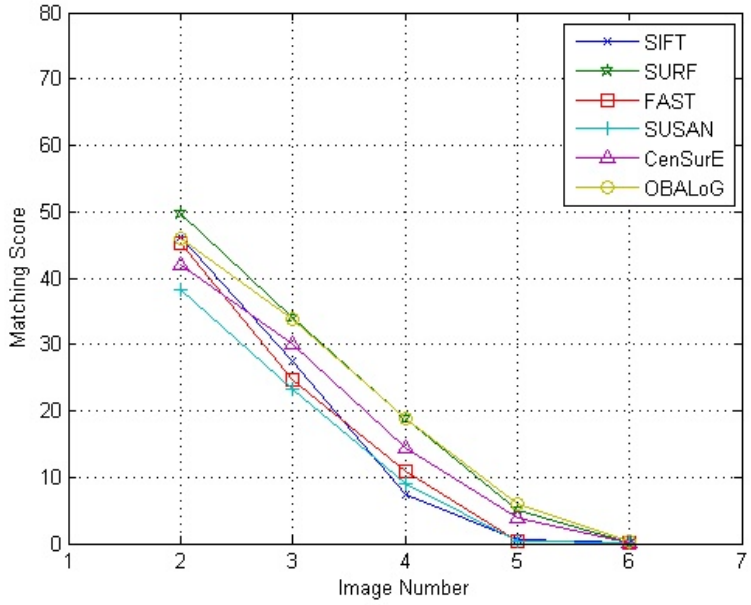
(b)

(c)

Figure 6.5: *Feature Correspondence on the wall sequence*



(a)



(b)

Figure 6.6: *Repeatability results(a) and matching scores(b) for different detectors on the wall sequence*

Table 6.1: *Timing Results*

SIFT	1683ms
SURF	995ms
FAST	1108ms
SUSAN	1946ms
CenSurE	512ms
OBALoG	721ms

shows the average accuracy of matching for detectors on different image sequences.

Direct examination of the Eigen values of the Hessian matrix is the most accurate and repeatable measure of interest points. Local extrema from the LoG filter should include all these same features plus less repeatable edges. Regardless, edges are efficiently rejected. With this in mind, the best the approximation of either the Hessian or the LoG filter should result in the most accurate feature set. SIFT's accuracy measures indicate that the DoG approximation of the LoG filter is very precise. SURF approximates the Hessian directly using boxes. Even though their method is geometrically constrained to certain ratios of box sizes, their approximation is also quite accurate. FAST traded its accuracy for speed, but did not have the advantage of integral images, and thus isn't as efficient nor as accurate as other detectors. Similarly, CenSurE's two box approach isn't as accurate as SURF, SIFT or OBALoG. The main disadvantage of the CenSurE algorithm lies in the detection of limited number of interest points. This is due to inaccurate representation of the LoG filter by using just 2 box filters. Although it manages to maintain a decent repeatability ratio for the features it detected, CenSurE's use is mostly restricted to applications requiring only a small set of interest points. OBALoG improves on CenSurE as it approximates the LoG efficiently using six box filters.

6.2 Timing Results

Table 6.1 shows the timing results on a 500 x 320 image for these detectors running on the same Intel Core2 Q6600 (2.4GHz) processor. SURF, CenSurE and OBALoG are faster than

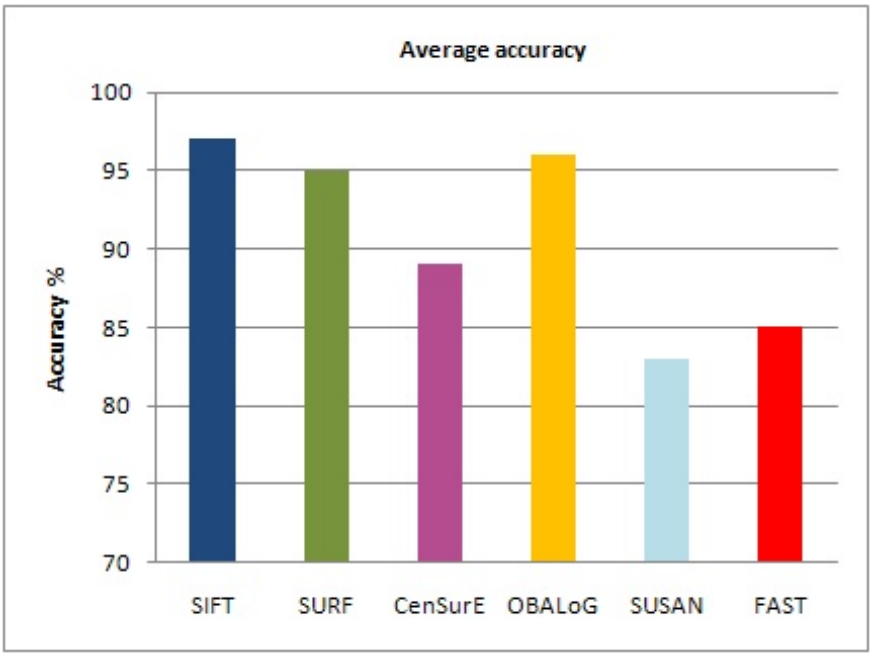


Figure 6.7: *Overall accuracy of correspondence for different detectors*

SIFT, FAST and SUSAN due to the use of integral images for fast convolution. Between SURF, CenSurE and OBALoG, the latter two are faster than SURF since SURF employs 10 box filters to approximate the Hessian while CenSurE and OBALoG only have 2 and 6 boxes respectively for their LoG approximation. Although OBALoG and CenSurE involve the additional step of using the scale-adapted Harris measure to eliminate weak extrema, these computations only occur at detected interest points, not the whole image.

6.3 Improved feature detection in orbiter imagery using Modified OBALoG

Testing was done on several regular HiRISE image stereo pairs from the Mars Reconnaissance Orbiter[22]. These 20000 x 50000 pixel images were split into smaller 5000 x 5000 pixel blocks. SIFT, SURF and the Modified OBALoG algorithms were applied on these blocks and their timing results are presented in Table 6.2 (All the detectors ran on the same Intel Core2 Q6600 (2.4GHz) processor). Threshold was applied on these algorithms to ensure the

Table 6.2: *Timing Results for a HiRISE Image*

SIFT	10.208s
SURF	6.984s
Modified OBALoG	2.969s

detection of same number of interest points.

The Modified OBALoG algorithm was found to be around four times faster than the SIFT algorithm in runtime performance. It picked out uniformly distributed features throughout the images. In addition, the feature detector in Modified OBALoG algorithm considered the scale change factor between the images (shown in Fig. 6.8 and 6.9). Since the scale difference between the images in the stereo pair was known to be less than 10%, the scale space setup used for feature detection was constructed using this known scale step. This also helped reduce the number of false matches that usually arise when there is a large difference in scale between the two images.

Figure 6.8 represents the clustered feature correspondence found when SIFT algorithm was applied on a large block of the HiRISE image East Martian Tholus[22]. Since a large threshold was applied to select adequate numbers of features, only the top quality features on the feature-rich top-left portion of the image were matched. There are very few or no features in other areas of the image. Figure 6.9 shows the performance of Modified OBALoG algorithm on the same HiRISE image. Here, the large image was divided into blocks and an adaptive threshold was applied. A higher threshold was used on the top-left region and a lower threshold was applied for other regions to ensure detection of adequate number of features. This resulted in feature matches that were uniformly spread across the image.

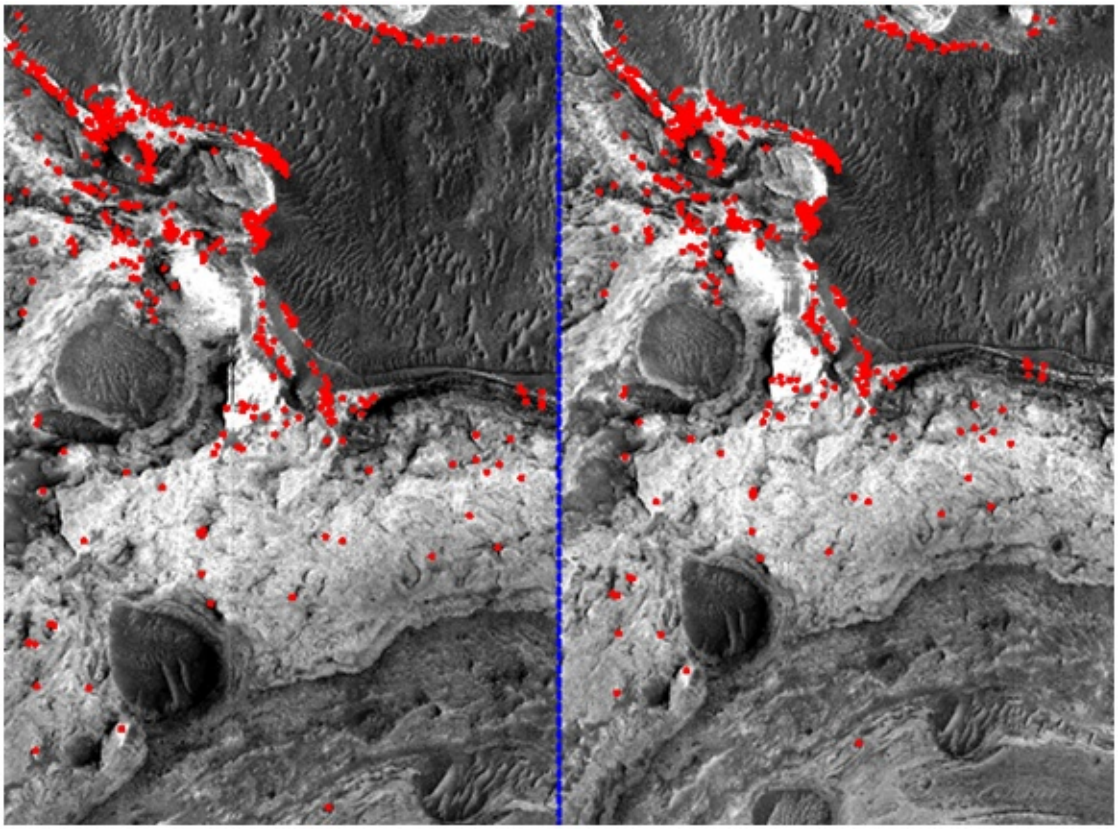


Figure 6.8: *This figure shows clustered feature correspondence when a high threshold SIFT algorithm was applied on orbiter imagery*[22]*. Many features matches were clustered in the top-left portion of the image, ignoring other regions.*

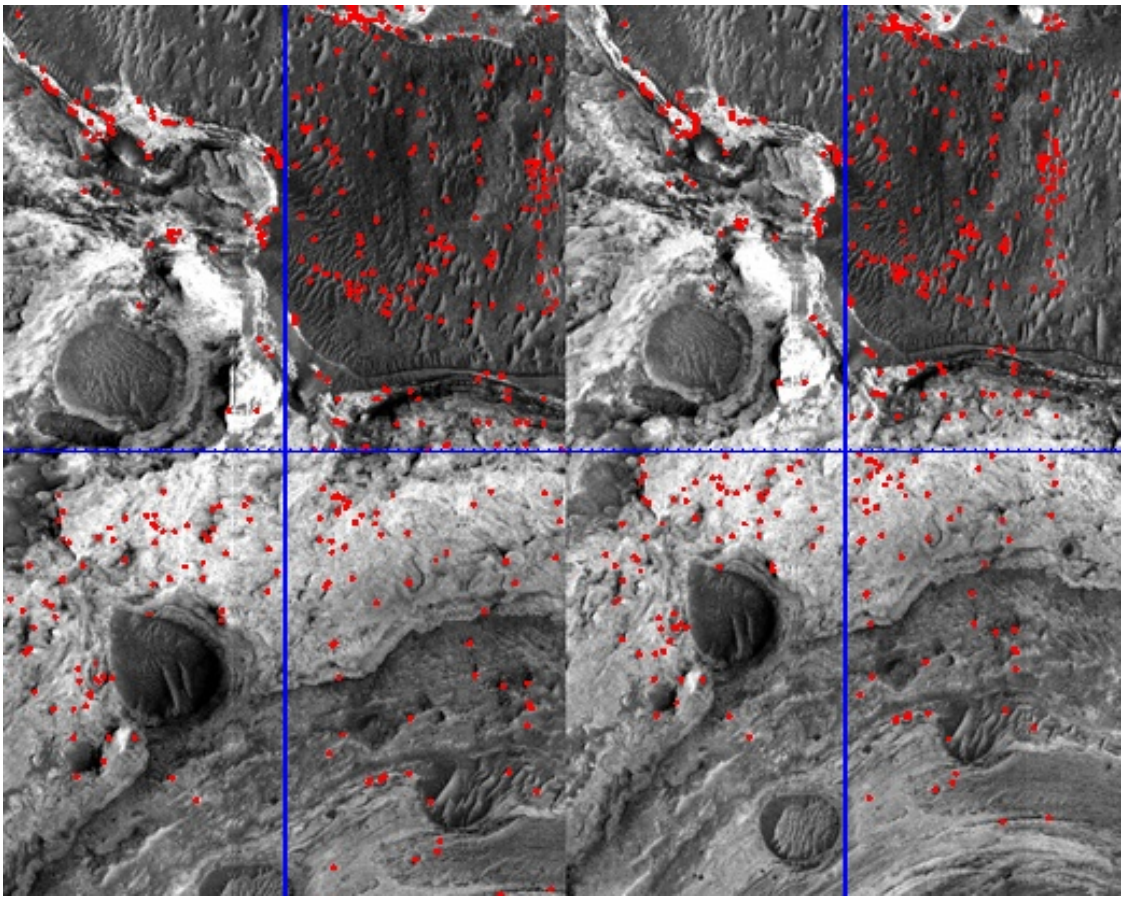


Figure 6.9: *This figure shows uniformly distributed feature correspondence using the Modified OBALoG algorithm. It involves block processing of the stereo imagery with adaptive thresholds*

Chapter 7

Conclusions

This thesis presented a novel approach for feature detection by using weighted box filters to approximate the Laplacian of Gaussian. In this approach, interest points are computed as the extrema of the filter responses over a variety of scales, using the original image resolution for each scale. These interest points are stable, repeatable and efficient even in the presence of image deformations. A modified SURF descriptor was used to describe the neighborhood of interest points. Experimental comparison of the performance of this algorithm with other standard feature detectors was done in terms of repeatability and speed. It was found that the OBALoG detector is faster than other leading methods while detecting accurate and repeatable interest points. The future intent of this work is to release an optimized algorithm for unrestricted use.

Bibliography

[1] M. Brown and D. Lowe. Invariant features from interest point groups. In *Electronic Proceedings of the 13th British Machine Vision Conference*, pages 253 – 262, 2002.

[2] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference*, pages 147 – 151, 1988.

[3] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45 – 78, 1997.

[4] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, November 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008.

[7] M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: Center surround extremas for real-time feature detection and matching. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5305 LNCS Part 4, pages 102 – 115, 2008.

[8] M. Grabner, H. Grabner, and H. Bischof. Fast approximated SIFT. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3851 LNCS, pages 918 – 927, 2006.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381 – 395, June 1981.

[10] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 207(1167):187–217, February 1980.

[11] Moravec H. P. Obstacle avoidance and navigation in the realworld by a seeing robot rover. Technical report, Robotics Institute, Carnegie-Mellon University, September 1980.

[12] P. Wyatt and H. Nakai. Fast feature extraction using approximations to derivatives with summed-area images. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3851 LNCS, pages 776 – 786, 2006.

[13] S. R. Gunn. On the discrete representation of the laplacian of gaussian. *Pattern Recognition*, 32(8):1463 – 1472, August 1999.

[14] S.-C. Pei and J.-H. Horng. Design of FIR bilevel laplacian-of-gaussian filter. *Signal Processing*, 82(4):677 – 691, 2002.

[15] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63 – 86, 2004.

[16] Christopher Evans. Notes on the opensurf library. Technical report, January 2009.

[17] USGS. USGS digital elevation models (DEM's) resources, 2009. http://data.geocomm.com/dem/demdownload.html.

[18] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern

synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[20] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pages 430 – 443, 2006.

[21] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615 – 1630, October 2005.

[22] NASA/JPL/University of Arizona. Hirise planetary data system. http://hirise-pds.lpl.arizona.edu/PDS/.

Appendix A

GNU Free Documentation License

Version 1.3, 3 November 2008

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a

printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this

License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given

on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections

Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been

terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix B

Functions used in OBALoG Implementation

1.Integral Image

Overview: Creates integral images from the input image

Inputs: An Image

Processes: Creates the integral image representation by summing the pixels to the left and above of the pixel in the input image

Outputs: The integral image representation

```
// Find the integral image
integral = IntegralImage(image);
```

2.Scale-space setup

Overview: Constructs scale-space using the box filtered images at different scales

Inputs: Integral image and the scales

Processes: Loads the predefined box dimensions and weights based on scale. Convolves the integral image with OBALoG boxes

Outputs: An array of image structures that form the scale-space

```
// Scale-space
 for ( unsigned scale = 0; scale < 8; scale++ ) {
   BoxFilter Boxes = DesignBoxes(integral,scale);
```

```
    interest_data.push_back( ProcessWithBoxes(integral,Boxes) );
 }
```

3.Non-maximal suppression and thresholding

Overview: Finds potential interest points from the scale-space images

Inputs: Scale-space images and threshold value

Processes: Checks if that pixel satisfies threshold condition. Performs non maximal suppresion

Outputs: A list of potential interest points

```
if ( is_3_3_3_max( interest_data, ii, j, k, threshold ) )
// If threshold condition is satisfied,extrema is found: save it
interest_pts.push_back(InterestPoint(ii,j,scale,interest_data[k](ii,j)));
```

4.Scale-adapted Harris Method

Overview: Checks if the interest points are corners

Inputs: The scale-space images and each interest point data

Processes: Checks if the ratio of principal curvatures satisfies a threshold condition

Outputs: List of final interest point candidates

```
if (!IsHarrisCorner(interest_data, (*it).scale, (*it).ix, (*it).iy ))
it = interest_pts.erase(it);
```

5.Orientation

Overview: Assigns the dominant direction of each interest point as its orientation

Inputs: Integral image and the list of interest points

Processes: Calculates haar wavelet responses in a circular region around each interest point. Sums the responses and finds the dominant direction using a sliding window

Outputs: List of interest points with their orientations assigned

```
interest_pts=OBALoGOrientation(integral,interest_pts);
```

6.Descriptor

Overview: Extracts descriptor components for a given set of detected interest points

Inputs: Integral image and the list of interest points

Processes: Calculates Haar wavelet responses in a bounding square around interest point.

Extracts 64-dimensional descriptor vector based on sums of haar wavelet responses

Outputs: List of interest points with their descriptors

```
interest_pts= OBALoGDescriptor(integral,interest_pts);
```

7.Matching

Overview: Matches two lists of interest points

Inputs: Interest point lists of two images

Processes: Performs nearest neighbor distance ratio to determine matches

Outputs: List of matches