Chapter 7
A Novel Feature Descriptor based on Elliptical Sampling

7.1 Introduction

Summary of the enhancements/improvements done for feature description methods till 2017: Computer Vision defines image matching as a fundamental task to establish correspondences between similar objects of the same scene in different images that undergo certain geometric or photometric deformations. Many applications such as, Simultaneous Localization and Mapping (SLAM) system [Zou et al. 2013, Bresson et al. 2015], image retrieval [Jain at al. 2015, He 2010], camera calibration [Wang et al. 2015, Liu et al. 2014], 3-D reconstruction [Michailidis et al. 2014, Xue et al. 2014], object recognition and tracking [Liu et al. 2012, Belongie et al. 2002] etc. rely on adequate image matching algorithms. These image matching algorithms follow a two-stage procedure for defining a list of stable and useful features in images. The two-stage procedure starts with extracting features of interest using a suitable detector, for example: Scale Invariant Feature Transform (SIFT) [Lowe 2004], Harris-Laplace [Mikolajczyk and Schmid 2002, Mikolajczyk and Schmid 2004], Hessian Laplace [Mikolajczyk and Schmid 2002, Mikolajczyk and Schmid 2004], Maximally Stable Extremal Regions (MSER) [Matas et al. 2004] etc. In the second stage, each of these detected features are uniquely defined using their respective local neighborhood in the form of a distinctive descriptor vector. The descriptor evaluation for an extracted feature makes it easy to identify the same feature in a different image that captures the same scene under certain transformations. Therefore, the task of establishing a discriminating and efficient local descriptor that performs well even for affine transformations becomes difficult.

In recent decades, many studies have proposed new techniques for describing affine invariant descriptors for local features. For example: SIFT, proposed by Lowe [Lowe 2004], describes each interest point with its local neighborhood by a gradient orientation histogram of 128 dimension vector. As an extension to SIFT, PCA-SIFT descriptor decrease the high dimensionality of original SIFT descriptor by applying the standard Principal Components Analysis (PCA) technique to a 41 × 41 image patch extracted around the keypoint [Ke and Sukthankar 2004]. PCA-SIFT, using the image patch, computes image gradients in the vertical and horizontal directions to formulate a feature vector. Therefore, the feature vector of length 2 × 39 × 39 = 3042 dimensions is computed and is normalized to unit magnitude to lessen the repercussions of illumination variations in a scene. Yu and Morel [Yu and Morel 2011], in their study proved that SIFT is only invariant to zoom, rotation and translation parameters, among the six parameters of affine transformations. Therefore, Yu and Morel proposed an affine invariant method (Affine-SIFT (ASIFT)), and based their theory on the fact that camera captured images of a physical object with a smooth boundary in changing positions undergo smooth apparent deformations and affine transforms of the image plane can very well approximate these local deformations. Therefore, normalization methods are used by the authors to extract affine invariant features for performing tasks like solid object recognition and the performance of affine recognition is evaluated using two parameters: absolute tilt: degree of tilt between the frontal view (f_1) and slanted view (f_2) of the image scene, and transition tilt: real time captured images are usually slanted views, therefore, transition tilt is used to measure the degree of tilt between two such slanted views (f_2) and (f_3). The authors simulated all image views by varying three parameters: the latitude angle, the longitude angle and scale. However, since all viewpoints and scales are simulated, ASIFT's computational complexity is inefficient to match local descriptors. Gradient Location and Orientation Histogram (GLOH) descriptor [Mikolajczyk and Schmid 2004], proposed by Mikolajczyk and Schmid, enhances the robustness of SIFT descriptor by adopting a log-polar location grid and uses PCA to reduce the dimensionality of the descriptor. The GLOH descriptor is more distinctive than SIFT but is computationally more expensive [Mikolajczyk and Schmid 2004]. Speeded-Up Robust Features Descriptor (SURF) descriptor, proposed by Bay et al. [Bay et al. 2008] is designed as an efficient alternative to SIFT. The main advantage of the SURF descriptor over SIFT is the processing speed as SURF only uses a 64 dimensional feature vector to describe a local feature. Although SURF is proven to be efficient for a wide range of computer vision applications, it also has some shortcomings like: SURF descriptor is not fully affine invariant [Pang et al. 2012] and it fails under extreme changing imaging conditions.

Few more efficient descriptors have been proposed till date. For instance, Local Binary Pattern (LBP) descriptor [Heikkiläa et al. 2009], analyzes the spatial structure of a texture by computing the order based feature for each pixel by comparing each pixel's intensity value with that of its neighboring pixels. In recent times, several variations of LBP have been proposed, example, Center-Symmetric Local Binary Pattern (CS-LBP) [Hong et al. 2014], Local Ternary Pattern (LTP), Center-Symmetric Local Ternary Pattern (CS-LTP) and Orthogonal-Symmetric Local Ternary Pattern (OS-LTP) [Huang et al. 2015]. Unfortunately, despite many advantages of these texture based binary descriptors, they produce higher dimensional features and are insensitive to Gaussian noise on flat regions. Another widely used binary descriptor is the Binary Robust Independent Elementary Features (BRIEF) [Calonder et al. 2010] which operates on simple binary comparison test and uses Hamming distance instead of Euclidean or Mahalanobis distance. However, BRIEF descriptor is not rotation invariant and fails to operate for rotation transformation for more than 35° (approx.) angle. Features from Accelerated Segment Test (FAST) [Roster et al. 2010], Binary Robust Invariant Scalable Keypoints (BRISK) [Leutenegger et al. 2011] and Oriented FAST and rotated BRIEF (ORB) [Rublee et al. 2011]) adds to the list of other widely used binary feature detectors. However, a comparative study of binary detectors and descriptors against conventional feature detection methods such as SIFT and SURF conducted by Heinly et al. [Heinly et al. 2012] shows that, except for non-geometric transformations, SIFT outperforms all binary feature detectors and descriptors.

Need of the proposed improvement: In spite of many advancements made till date to develop a fully affine invariant feature descriptor that satisfies characteristics like low computational complexity and robustness, there still exists much scope of improvement. Proposed feature descriptor focuses on improving feature description efficiency in terms of execution time while finding sufficient number of correct matches between an image pair that describes a scene under varied imaging conditions.

7.2 Proposed Improvement in Feature Description

A novel feature descriptor based on local elliptical sampling for image matching is proposed. The design of the proposed feature descriptor is based on the fact that if a plane is transformed using affine transformation of linear form using homogeneous coordinates, and a circle in that plane is perfectly confined by a square, then after transformation, the square transforms to a parallelogram and the circle transforms to an ellipse [Foley et al. 2015]. Therefore in present work, circular sampling is done in the reference image and elliptical sampling is done in the matched image respectively as the scene described in the matched image is expected to be transformed in some way from the one described in the reference image. Finally, orientation histograms over the gradient magnitudes of these sampling points are used to generate the feature descriptor vector. Also, circular and elliptical sampling is done in two ways: 1) using parametric equations and 2) using elliptical curve tracking (i.e. scan conversion at subpixel levels). The purpose of the later is to further reduce the computational complexity of the descriptor while maintaining efficiency in terms of correct number of matches.

7.3 Local Sampling

This section provide details of the sampling procedure for the proposed feature descriptor. Let $k_i =$

(x_n, y_n, σ_n) denotes the extracted keypoint from the input image I, where n describes the index of the keypoint k_i located at (x_n, y_n) position in the input image I and is extracted at scale σ_n .

For computing the local descriptor for the extracted keypoint k_i, neighboring pixels of k_i are sampled from the local image patches. Thereby, two types of sampling evaluations are considered: elliptical sampling and circular sampling which are implemented using two approaches as detailed below (Figure 7.1, Subsection 7.3.1 and Figure 7.4, Subsection 7.3.2 gives the illustration for elliptical and circular sampling respectively).

7.3.1 Elliptical Sampling

In elliptical sampling, three parameters are used: the major axis j_k, minor axis n_k (where *k* is the order of elliptical sampling as shown in Figure 7.1(a)) and the rotation angle θ_0 (Figure 7.1 (a)). The major and minor axes of the ellipse at the order *k* are defined as:

$$j_k = kw/m, k = 1, \ldots., m \text{ and } \; n_k = j_k/q \tag{7.1}$$

where, $m = 10$, $q = 2$ and w is the window size of local region defined as follows [Vedaldi and Fulkerson 2008]:

$$w = (3\sigma_n \times \sqrt{2} \times 5 + 1)/2 \tag{7.2}$$

where, σ_n represents the scale of the extracted keypoint. In equation (7.1), $n_k = j_k/q,\; q$ defines the relation between major and minor axis and for $q = 1$, circular sampling is processed on similar configurations as the elliptical sampling. Also, equations (7.1) and (7.2) defines the relationship between the scale of the extracted keypoint σ_n and major axis j_k.

The two approaches adopted for elliptical sampling implementation are as follows:

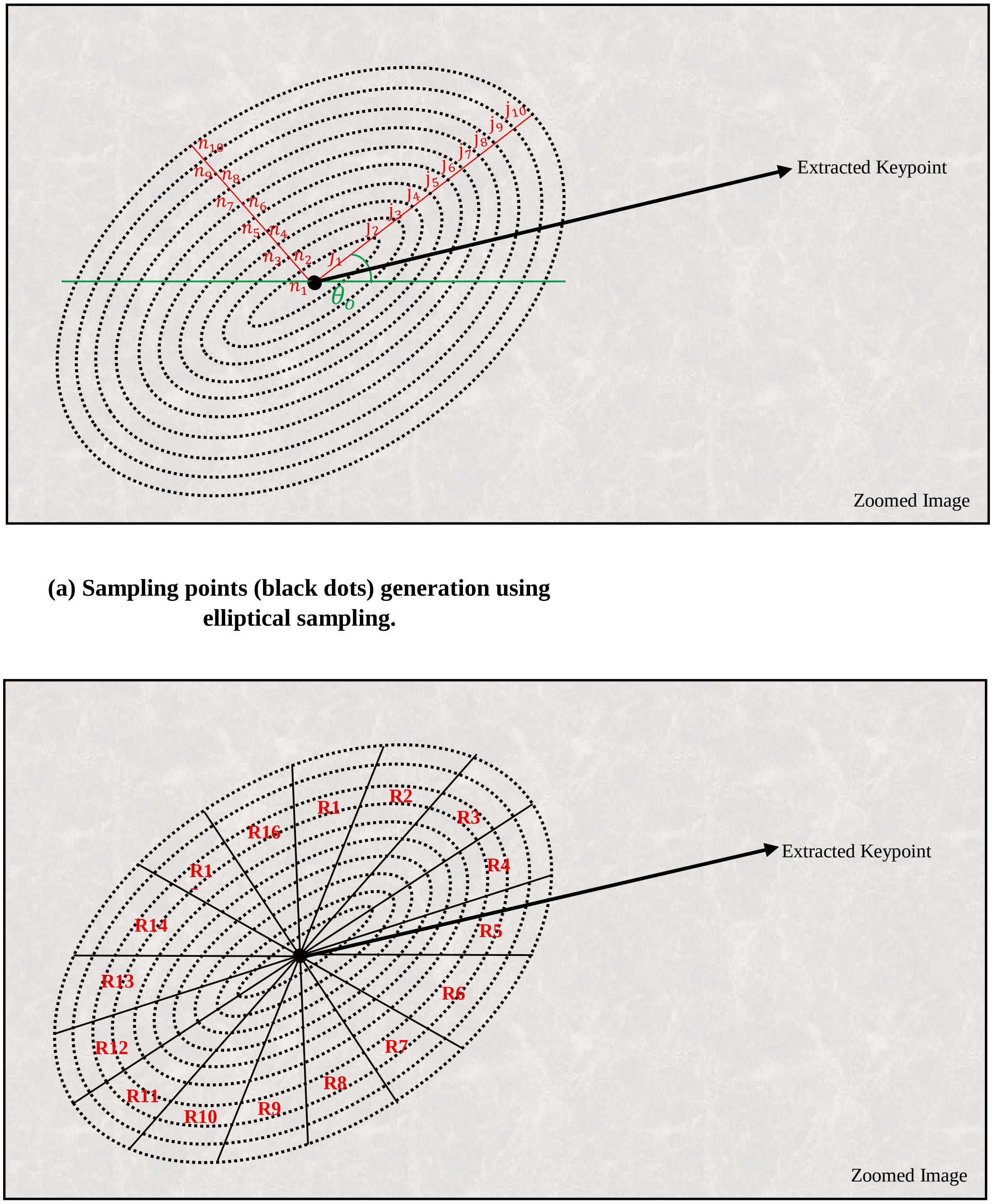
1. Using Parametric Equations

The elliptical sampling is done as follows:

$$x_n^{kp} = j_k \cos(\theta_p)\cos(\theta_0) - n_k \sin(\theta_p)\sin(\theta_0) + x_n$$

$$y_n^{kp} = n_k \sin(\theta_p)\cos(\theta_0) + j_k \cos(\theta_p)\sin(\theta_0) + y_n \; , \tag{7.3}$$

where, (x_n, y_n) represents the center of ellipse, $(x_n^{k\rho}, y_n^{k\rho})$ denotes *n*th pixel coordinate obtained at *k*th concentric ellipse at delta *p* which corresponds to θ_p in equation (7.3) and is the angular step size chosen for generating a more continuous ellipse boundary. For kth concentric ellipse, $\theta_p = 1/j_k$. The rotational angle $\theta_0 \in [0^\circ, 90^\circ]$, as for an ellipse with rotational angle θ_0, the major axis $j_k = j$ and minor axis $n_k = n$ (major and minor axes equivalence to some j and n value is just taken as an assumption to explain the concept). Now, if the rotation angle of the ellipse is changed to $\theta_0 + 90^\circ$ with respect to positive x axis, the major and minor axes of the ellipse remains same, i.e., $j_k = j$ and $n_k = n$.



(a) Sampling points (black dots) generation using elliptical sampling.

(b) Descriptor Presentation

Fig. 7.1. Elliptical Sampling

2. Using Elliptical Curve Tracking

Sampling is done by scan converting the ellipse (i.e. calculating the points $(x_n^{k\rho}, y_n^{k\rho})$) using a variant of the algorithm described in [Foley et al. 2005]. The algorithm is based on Pitteway curve tracking algorithm [Pitteway 1967] and is designed for general conics like ellipses with tilted axes, hyperbolas, circles and parabolas. The algorithm works by first defining the conic using the six coefficients G, H, I, J, K and L of the equation:

$$C(x,y) = Gx^2 + Hxy + Iy^2 + Jx + Ky + L = 0 \tag{7.4}$$

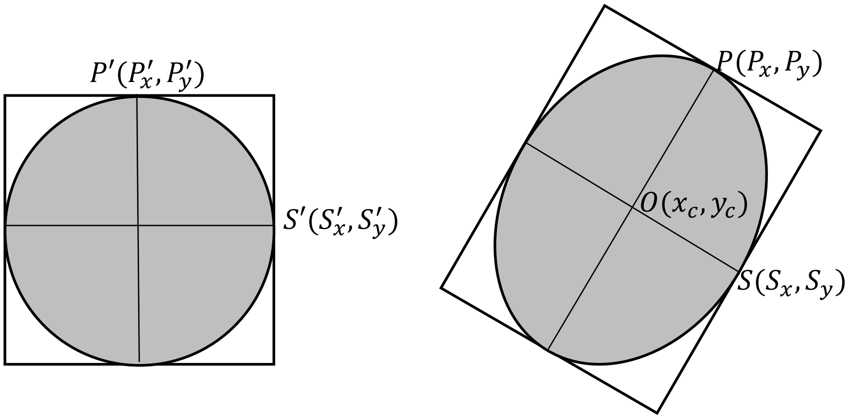
and then performing scan conversion.

Considering a circle on a plane confined by a unit square that undergoes linear transformation, the square tends to transform to a parallelogram and the circle transforms to an ellipse. So, the midpoints of the sides of the parallelogram (S and P transformed from S' and P' (midpoints of the sides of the square) as shown in Figure 7.2) are points on the ellipse and hence, determining these points along with the center of the parallelogram ($O(x_c, y_c)$) would also determine the points of the ellipse. Therefore, for points $S(S_x, S_y)$ and $P(P_x, P_y)$, the six coefficients from equation (7.4) are given as:

$$G = \left(S_y^2 + P_y^2\right);\ H = -2\left(S_x S_y + P_x P_y\right);\ \ I = \left(S_x^2 + P_y^2\right);\ \ J = 0;\ \ K = 0;\ L = -\left(S_x P_y - S_y P_x\right)^2 \tag{7.5}$$

Now if the resulting ellipse is translated to a new coordinate system centered $-S(S_x, S_y)$, the equation of the ellipse becomes:

$$C(x,y) = G(x + S_x)^2 + H(x + S_x)\left(y + S_y\right) + I\left(y + S_y\right)^2 + J(x + S_x) + K(y + S_y) + L = G'x^2 + H'xy + I'y^2 + J'x + K'y + L' = 0 \tag{7.6}$$



(a) Circle in a plane perfectly confined by a square

(b) Transformed Square and Circle to a parallelogram and an ellipse respectively

Fig. 7.2. Circle to Ellipse Transformation

and comparing the coefficients of similar terms and solving resulting algebraic equations, the six coefficients for the new coordinate system centered $-S(S_x, S_y)$ are given as:

$$G' = G;\ H' = H;\ \ I' = I;\ \ J' = 2P_y\left(S_xP_y - S_yP_x\right);\ \ K' = -2P_x(S_xP_y - S_yP_x);\ L = 0 \qquad (7.7)$$

Scan converting this conic would yield its points and origin would lie on this particular conic. But if point $\left(S_x, S_y\right)$ is added to each point of this conic, then points of the original conic which is centered at the origin could be obtained. So, for efficient scan conversion and to take the advantage of symmetry of ellipse curve, the process is divided into eight octants. The current octant indicates the tracking direction of the algorithm i.e. to go to the next octant from the current octant, the choice is made between making a square move or a diagonal move depending on whether one coordinate changes or both does. Depending on the current octant, from the current pixel, square move is defined as: just above, just below, just right or just left. Similarly, diagonal move is defined as: move right then above, move right then below, move left then above or move left then below (Figure 7.3, where C indicates current pixel and, S and D indicates pixels after making square and diagonal move respectively). Also, the algorithm keeps a track of the odd and even numbered octants which are terminated by reaching diagonal and square octant boundary respectively. To determine the starting octant, it is noted that at any point of conic that satisfies $C(x, y) = 0$, the gradient of C i.e., $(2Gx + Hy + J, Hx + 2Iy + K)$, points perpendicular to the conic. The gradient coordinates are used for determining the direction of motion and hence drawing the octant. Also, for each diagonal and square move, the decision variable D_v is updated by adding increments I_d and I_s respectively. If $C(x, y) = 0$ is the equation of the ellipse, then D_v is defined by evaluating C at the midpoint of the segment between the next two pixel choices. As it is known that $C(x, y) < 0$ and $C(x, y) > 0$ if the point (x, y) is inside and outside the ellipse respectively, therefore, $(-D_v)$ indicates that the ellipse passes outside the midpoint and outer pixel should be selected. Similarly, $(+D_v)$ indicates the inner pixel selection. In case when $D_v = 0$, for odd numbered octants: square move is taken for $(-D_v)$ and for even numbered octants: diagonal move is taken for $(-D_v)$. Taking an example: In octant 1, if (x_c, y_c) represents the current drawn pixel, D_v is the decision variable to decide the selection of next pixel between $(x_c + 1, y_c)$ and $(x_c, y_c + 1)$ and the representations I_{d+1} and I_{s+1} are used for the

D	***S***	***D***
S	***C***	***S***
D	***S***	***D***

C: Current Pixel
S: Pixel accessed after making Square move
D: Pixel accessed after making Diagonal move

Fig. 7.3. Square and Diagonal move in an octant

added increment to D_v for evaluating D_{v+1} , then at each pixel following tasks are performed: 1) Current pixel is stored 2) next pixel is selected based on the value of D_v . 3) I_d and I_s are updated to I_{d+1} and I_{s+1} based on the choice made. 4) D_v is updated to D_{v+1} by adding either I_{d+1} and I_{s+1} . 5) Checking for octant change.

The D_{v+1} value is evaluated from D_v by a differencing technique, i.e., if in the current octant (Octant 1), pixel to be chosen beyond (x_c, y_c) is decided using $D_v = C\left(x_c + 1, y_c + \frac{1}{2}\right)$, then for a square move $x_{c+1} = x_c + 1$ and $y_{c+1} = y_c$, new decision variable $D_{v+1} = C(x_c + 2, y_c + \frac{1}{2})$ and I_{s+1} is given as:

$$I_{s+1} = D_{v+1} - D_v = G(2x_c + 1) + H\left(y_c + \frac{1}{2}\right) + J + 2G \tag{7.8}$$

And for a diagonal move $D_{v+1} = C(x_c + 2, y_c + \frac{3}{2})$ and I_{d+1} is given as:

$$I_{d+1} = D_{v+1} - D_v = (2G + H)x_c + (H + 2I)y_c + G + \frac{H}{2} + J + K + (2G + 2H + 2I) \tag{7.9}$$

Now if $I_s = G(2x_c + 1) + H\left(y_c + \frac{1}{2}\right) + J$, then $I_{s+1} = I_s + 2G$. Similarly, if $I_d = (2G + H)x_c + (H + 2I)y_c + G + \frac{H}{2} + J + K$, then $I_{d+1} = I_d + (2G + 2H + 2I)$. Also, even if not used, both values of I_{d+1} and I_{s+1} are calculated for square and diagonal moves and are given as:

Square move: $I_{s+1} = I_s + 2G$ and $I_{d+1} = I_d + (2G + H)$ (7.10)

Diagonal move: $I_{s+1} = I_s + (2G + H)$ and $I_{d+1} = I_d + (2G + 2H + 2I)$ (7.11)

Transition from Octant 1 to next octant is done when the two components of the gradient vector $\left(\frac{\partial C}{x}, \frac{\partial C}{y}\right)$ goes from being negative to being zero. As $\left(\frac{\partial C}{x}, \frac{\partial C}{y}\right) = (2Gx + Hy + J, Hx + 2Iy + K)$, so in terms of I_d and I_s , $\left(\frac{\partial C}{x}\right) = I_s - \frac{2G+H}{2}$ and $\left(\frac{\partial C}{x} + \frac{\partial C}{y}\right) = I_d - \frac{2G+H}{2}$. Therefore, sign of $\left(\frac{\partial C}{x} + \frac{\partial C}{y}\right)$ determines the transition from 1st octant to next octant and the sign of $\left(\frac{\partial C}{x}\right)$ serves as the corresponding check for transition of 2nd octant to next octant. Also, when transition is made from octant 1 to octant 2, the square moves become vertical rather than horizontal and the decision variable being updated becomes $D'_v = C\left(x_c + \frac{1}{2}, y_c + 1\right)$ and values for D'_v , I'_s and I'_d in terms of D_v , I_s and I_d is given as:

$$D'_v - D_v = C\left(x_c + \frac{1}{2}, y_c + 1\right) - C\left(x_c + 1, y_c + \frac{1}{2}\right) = \frac{I_d}{2} - I_s + \frac{3}{8}(2G + 2H + 2I) - \frac{1}{2}(2G + H) \tag{7.12}$$

$$I'_s - I_s = I_d - I_s - \frac{2G+H}{2} + \frac{2G+2H+2I}{2} \tag{7.13}$$

$$I_d' - I_d = -G + I \tag{7.14}$$

It is also observed that sometimes a diagonal move may result in incorrect evaluation of the ellipse's points and several octant changes are made in a single move, resulting in the breakdown of the ellipse. A solution to this problem is proposed was by Pratt [Pratt 1985] where while tracking pixels in octant 1, a radical direction change is observed in the gradient vector of $C(x, y)$ if an unusual jump across the ellipse is made. For example, in octant1, *x* and *y* component of the gradient vector is always positive and negative respectively, therefore if a jump is made across the ellipse, next ellipse point may lie in octant 3, 4, 5 or 6. So, a division line is determined between these and the opposite octants, i.e., octant 1, 2, 7 and 8, by setting the x component of the gradient vector, i.e., $2Gx + Hy + J = 0$. Also, as $I_s = G(2x_c + 1) + H\left(y_c + \frac{1}{2}\right) + J = 2Gx_c + Hy_c + J + G + H/2$, therefore, I_s can also be used to detect such crossings and similar monitoring can be done for each octant.

7.3.2 Circular Sampling

The circular sampling is done by treating the radius $r_k(k = 1 \dots . m)$ as a single supervised parameter, where *k* is the order of circular sampling as shown in Figure 7.4(a), and is defined as:

$$r_k = kw/m, k = 1, \dots., m \tag{7.15}$$

where, $m = 10$ and *w* is evaluated using equation (7.2). The above equations (7.2) and (7.15) defines the relationship between the scale of the extracted keypoint σ_n and radius r_k. The two approaches adopted for circular sampling implementations are as follows:

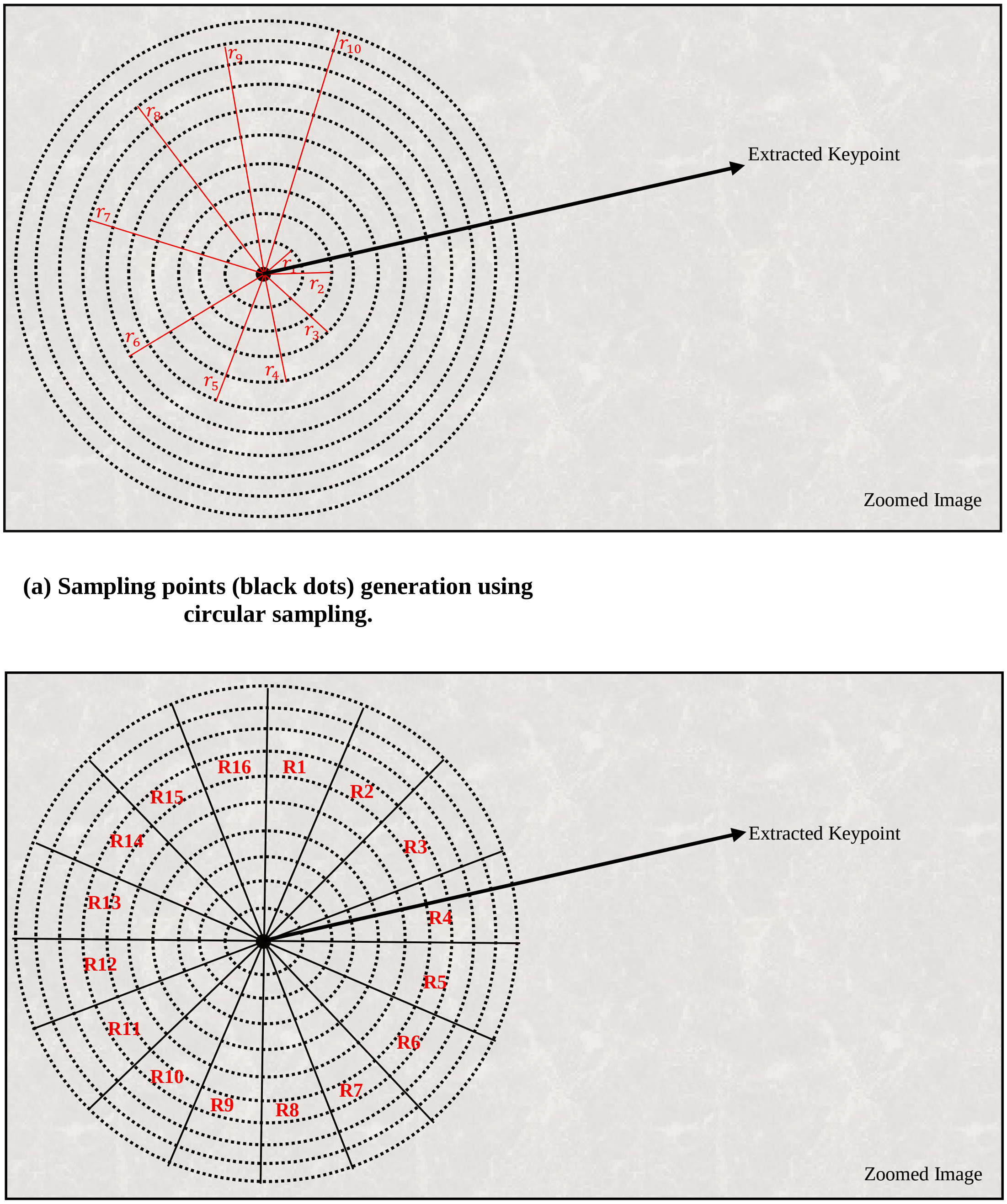
1. Using Parametric Equations

The circular sampling is done as follows:

$$x_n^{kp} = r_k \cos(\theta_p) + x_n$$

$$y_n^{kp} = r_k \sin(\theta_p) + y_n \text{ , where } k = 1, \dots., m \tag{7.16}$$

In the above equation (7.16), x_n^{kp} and y_n^{kp} denotes the horizontal and vertical coordinates of *n*th sampling point obtained at *k*th concentric ellipse at delta p which corresponds to θ_p in equation (7.16) and for kth concentric circle, $\theta_p = 1/r_k$. Also, equation (7.16) and (7.3) computes floating point values for $\cos(\theta_p)$ and $\sin(\theta_p)$ functions and thereby, for evaluating accurate image intensity at point $(x_n^{k\rho}, y_n^{k\rho})$, bilinear interpolation is used. For notation clarity, $Ɨ(x_n^{k\rho}, y_n^{k\rho})$ is used to denote image intensity at point $(x_n^{k\rho}, y_n^{k\rho})$ after bilinear interpolation.



(a) Sampling points (black dots) generation using circular sampling.

(b) Descriptor Presentation

Fig. 7.4. Circular Sampling

2. Using Circular (Elliptical) Curve Tracking

Sampling is done by scan converting the circle (i.e. calculating the points $(x_n^{k\rho}, y_n^{k\rho})$) using the same procedure followed for scan converting the ellipse [Section 7.3.1]. However, as circle is treated as a special case of ellipse where the major and minor axes (j_k and n_k respectively) are equal, i.e., $j_k = n_k =$

r_k , therefore while performing scan conversion for circle, the two axes are kept same. The rest of the implementation for curve tracking is done by following similar steps as detailed for ellipse scan conversion.

7.4 Feature Descriptor

The local descriptor is formulated using the sampling points generated using circular and elliptical sampling. The descriptor formulation is done by first computing the gradients of the sampling points, followed by orientation computation of local gradients. So, the formulation involved is as follows:

$$D_p\left(x_n^{kp}, y_n^{kp}\right) = \text{Ɨ}\left(x_n^{k(p+1)}, y_n^{k(p+1)}\right) - \text{Ɨ}\left(x_n^{k(p-1)}, y_n^{k(p-1)}\right) \quad (7.17)$$

$$D_k\left(x_n^{kp}, y_n^{kp}\right) = \text{Ɨ}\left(x_n^{(k+1)p}, y_n^{(k+1)p}\right) - \text{Ɨ}\left(x_n^{(k-1)p}, y_n^{(k-1)p}\right) \quad (7.18)$$

$$m\left(x_n^{kp}, y_n^{kp}\right) = \sqrt{(D_p\left(x_n^{kp}, y_n^{kp}\right))\text{^}2 + (D_k\left(x_n^{kp}, y_n^{kp}\right))\text{^}2} \quad (7.19)$$

$$angle\left(x_n^{kp}, y_n^{kp}\right) = atan2(D_p\left(x_n^{kp}, y_n^{kp}\right), D_k\left(x_n^{kp}, y_n^{kp}\right)) \quad (7.20)$$

In the above equations, $D_p\left(x_n^{kp}, y_n^{kp}\right)$ and $D_k\left(x_n^{kp}, y_n^{kp}\right)$ represents the gradient of image pixels that differ by delta p and *k*th concentric circle or ellipse respectively, $m\left(x_n^{kp}, y_n^{kp}\right)$ denotes the gradients magnitude and $angle\left(x_n^{kp}, y_n^{kp}\right)$ denotes the local gradient orientation.

Illustrating the difference between gradient computation in the proposed method from the original SIFT descriptor which uses nearby pixels for gradient computation such as $\left(x_n^{kp} - 1, y_n^{kp}\right)$, $\left(x_n^{kp} + 1, y_n^{kp}\right)$, $\left(x_n^{kp}, y_n^{kp} - 1\right)$ and $\left(x_n^{kp}, y_n^{kp} + 1\right)$, equation (7.17) and (7.18) determines the image pixel selection for evaluating the gradient magnitude and the local gradient orientation in equation (7.19) and equation (7.20) respectively. The main idea of following this strategy to perform optimal pixel selection to correctly reflect the variation in viewpoint between the two images that are being matched.

For final descriptor value, an orientation histogram is evaluated using the gradient orientation of sampling points computed using equation (7.20). The histogram is divided into 8 bins which covers the 360°range of orientations, i.e. $0^{\circ} - 45^{\circ}$is labelled as Bin1, $46^{\circ} - 90^{\circ}$ as Bin2 and so on. Also, following the similar convention as in [Lowe 2004], each sample added to the histogram is weighted by its gradient magnitude and a Gaussian weighting function with scale equal to half the radius of the outermost circle or ellipse of the descriptor window. Figure 7.4 illustrates the keypoint descriptor computation based on circular sampling, where Figure 7.4(a) gives a pictorial representation of sampling point selection and

Figure 7.4(b) represents the descriptor formulation where all the sampling points are summarized over 4 × 4 subdivisions with 8 orientation bins in each subdivision. Similarly, for elliptical sampling, Figure 7.1(a) and 7.1(b) shows sampling point selection and descriptor formulation respectively. Hence, as a result of both circular and elliptical sampling, a $4 \times 4 \times 8 = 128$ element feature vector is produced for each detected keypoint. Also, the effects of illumination changes are reduced by normalizing the feature vector to unit length. Moreover, the non-linear illumination changes that could potentially effect gradient magnitudes are handled by thresholding the values in the unit feature vector to be no larger than 0.4 and then again normalizing the feature vector to unit length. As in notation, the final descriptor is denoted by FD_i for the i^{th} keypoint.

7.5 Image Matching

In an attempt to match two images with diverse viewpoint variations, circular and elliptical sampling is done in the reference and the matched image respectively, where the circles in the reference image I_o are transformed into ellipses in the matched image I_m to pertain viewpoint and scale variations in the two images.

So, initially given the reference image I_o and the matched image I_m, SIFT detector is used for extracting keypoints from I_o and I_m. Each detected keypoint i is specified by its (x_i, y_i) coordinates, scale σ_i and orientation o_i. For M keypoints in the reference image I_o , feature descriptor obtained using circular sampling is denoted by $FD = \{FD_i\}_{i=1}^{M}$ and for N keypoints in the matched image I_m , feature descriptor obtained using elliptical sampling is denoted by $FD' = \{FD'_i\}_{i=1}^{N}$. The best match for each keypoint in the reference image is found by identifying its nearest neighbor from the collection of keypoints in the matched image. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. However, many factors like noise, object occlusion etc. may cause mismatching of keypoints in the matched image keypoint collection. Therefore, for discarding false matches, keypoint matching specifications from SIFT [Lowe 2004] are used where an efficient measure is applied to identify the closest and the second-closest neighbor to a particular keypoint in the matched image. The distance of these two neighbors from the keypoint is evaluated and compared as the correct match needs to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. Also, there is a possibility of finding a number of incorrect matches within similar distances due to the high dimensionality of the feature space. Therefore, second-closest match is treated as a measure to determine the density of incorrect matches within a particular portion of the feature space. The Probability Density Function (PDF) for correct and incorrect matches are thereby computed as the ratio of closest to second-closest neighbors of each keypoint. The PDF value for the nearest neighbor correct match is justified at a much lower ratio than that for incorrect matches i.e., matches with distance ratio greater than 0.8 are rejected eliminating 90% of the incorrect matches while discarding less than 5% of the correct matches [Lowe 2004].

7.6 Methodology & Experimental Setup

7.6.1 Methodology

Methodology and how to compare performance: The proposed feature descriptor is implemented using two variations in circular and elliptical sampling: 1) using parametric equations and 2) using curve tracking algorithm. Circular and elliptical sampling in both cases is done in the reference image and the matched image respectively. The two implementations are thereby compared in terms of correct number of correspondences between an image pair and time taken. Performance comparison of the proposed descriptor is done with one of the most conventional and widely used SIFT feature descriptor for determining and proving the efficiency of the proposed method.

Which statistics/metrics to use and how: Table 7.1 describes the corresponding tables and figures listing in the chapter with respect to comparative evaluation of the proposed descriptor (implemented by performing circular and elliptical sampling using parametric equations and curve tracking algorithm).

Table 7.1. Tables & Figures Representing Respective Performance Evaluation

	Table / Figure	**Comments**
Elliptical Sampling	Figure 7.2	Represents the sampling points selection and descriptor formation for elliptical sampling
Circular Sampling	Figure 7.4	Represents the sampling points selection and descriptor formation for circular sampling
Results for the proposed feature descriptor implemeted using parametric equations	Table 7.2	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset.
Results for the proposed feature descriptor implemeted using curve tracking algorithm	Table 7.3	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset.
Performance comparison of SIFT and the proposed descriptor	Table 7.4	The table also represents the speed-up obtained by the proposed feature descriptor for every pair of image in each image-set.

	Table / Figure	Comments
Average Speed-Up	Table 7.5	Represents the average speed-up for every image-set and the overall speed-up obtained by the proposed method.

7.6.2 Experimental Setup

Language, Software and Tools used for implementation and system specification: The experiments are carried out using single threaded code on a computer with 32GB RAM and Intel® Core ™ i7-6700 CPU@3.40ghz × 8 processor.

Implementation details: The proposed feature descriptor is implemented using two variations in circular and elliptical sampling, i.e., using parametric equations and using curve tracking. Implementation details for both the methods is given in Appendix E.1. For the case where circular and elliptical sampling is done using curve tracking, partial implementation from [Foley et al. 2005] is used. The code is completed and revised before using it in the work. Moreover, the elliptical sampling in both the cases is done at five different orientations of the ellipse, i.e., 0°, 30°, 45°, 60° and 90°, as rotational angle $\theta_0 \in [0^{\circ}, 90^{\circ}]$ for an ellipse (Reason briefed in Section 7.3.1). The results for the two cases for all five orientations are listed in Appendix E.2.

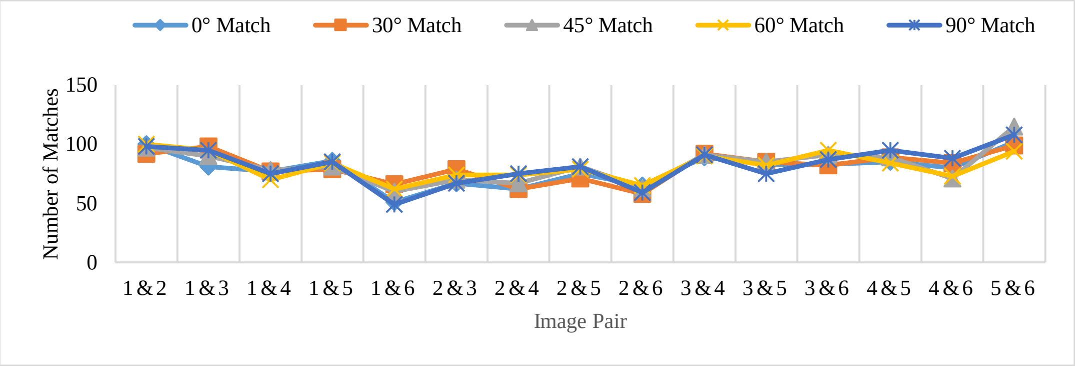
7.7 Data Reporting

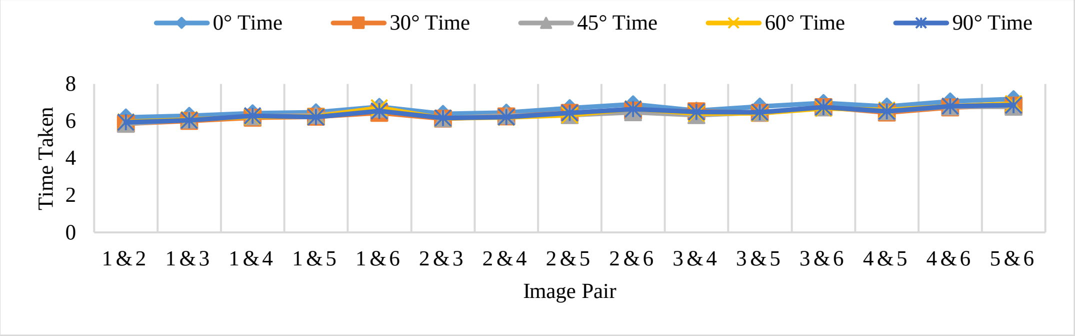
Dataset: The experiments are performed on Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1] as described in Chapter 4 [Section 4.6, Figure 4.2], containing eight image-sets with six images in each set (total 48 images). The images in the dataset defines image scene for five different imaging conditions: 1) viewpoint change, 2) scale change, 3) image blur, 4) illumination change and 5) JPEG compression.

The detailed results are depicted in Table 7.2 and Table 7.3 where corresponding graphs are represented for number of correct matches between every pair of image in each image-set and time taken (in seconds) by the two approaches used for performing elliptical and circular sampling for 0°, 30°, 45°, 60° and 90° ellipse rotation with respect to positive *x* axis (Please refer to Appendix E.2 for the numeric observations). From Table 7.2 and Table 7.3 it is clear that number of matches between image pairs and time taken for descriptor processing does not vary much in all image-sets for all five ellipse orientations. Table 7.4 represents the results for correct number of correspondences between an image pair and time taken (in seconds) for SIFT and the proposed descriptor. For the proposed descriptor, the values for correct number of matches and time taken for a particular image pair is the average of the output for that particular image pair for 0°, 30°, 45°, 60° and 90° ellipse rotation in both implementations.

From the graphs in Table 7.4 it is observed that the number of matches in case of SIFT are higher than the proposed method only in the case when two images describe an image scene that differs with very low degree of change in imaging condition and whenever the imaging condition is differing much, the proposed method always exhibits more number of matches in an image pair as compared to SIFT. For example, in case of Trees image-set, where degree of blur is sequentially increased in images from Image1 to Image6, the number of matches between an image pair by SIFT decreases drastically when Image1 is matched with Image6, but the proposed descriptor is able to exhibit a significant higher number of matches between the image pair. Figure 7.5 shows the results for number of matches by SIFT and the proposed method for six pairs of image of Trees image-set i.e. 1&1, 1&2, 1&3, 1&4, 1&5 and 1&6, where white lines represent the correspondences between an image pair. Similarly for other image-sets identical results are achieved that depicts stable performance of the proposed descriptor and its invariance to various parameters like viewpoint change, scale change, variations in image blur and illumination etc.

Table 7.2. Detailed Observations for 0°, 30°, 45°, 60° and 90° Ellipse Rotations for Circular and Elliptical Sampling using Parametric Equations

<table>
<tr><th>Image-set</th><th>Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]</th></tr>
<tr><td>Graffiti</td><td>




</td></tr>
</table>

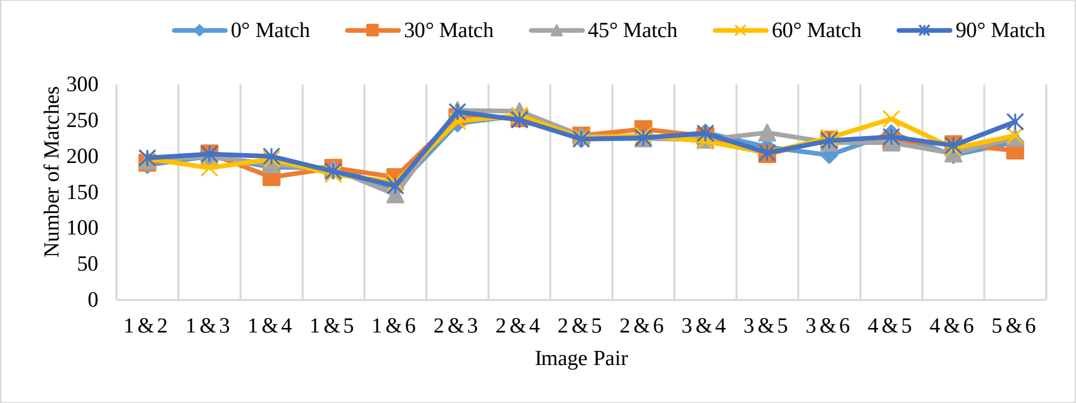
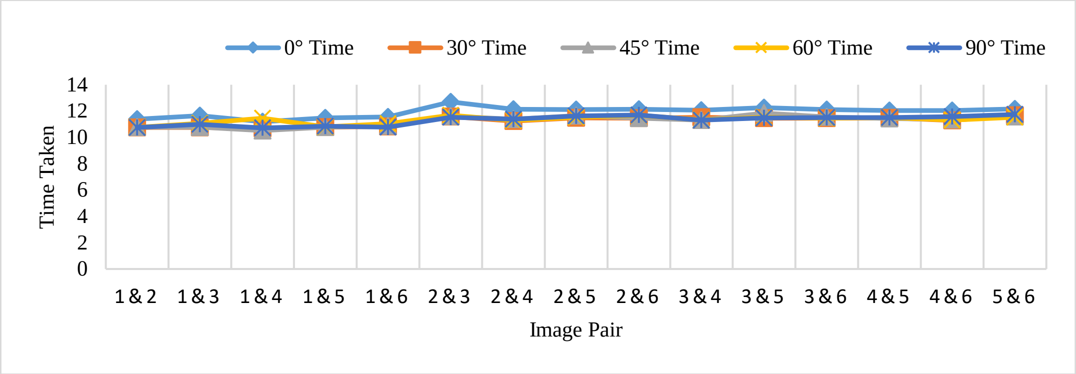
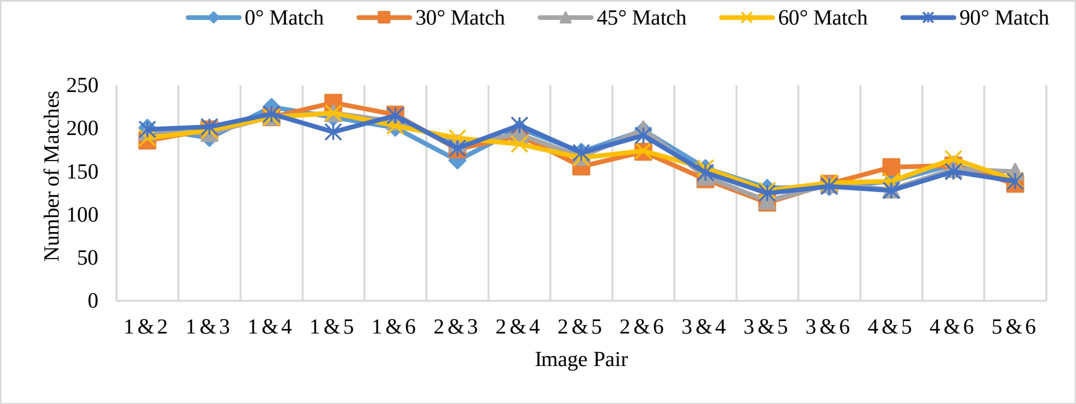
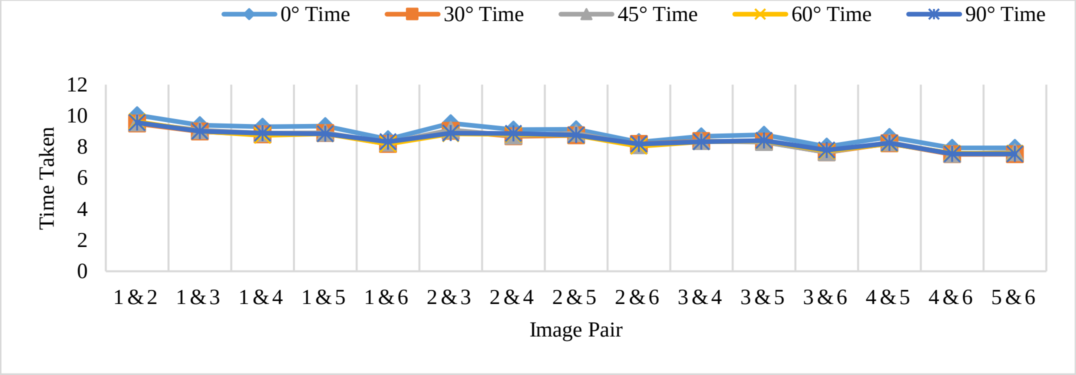
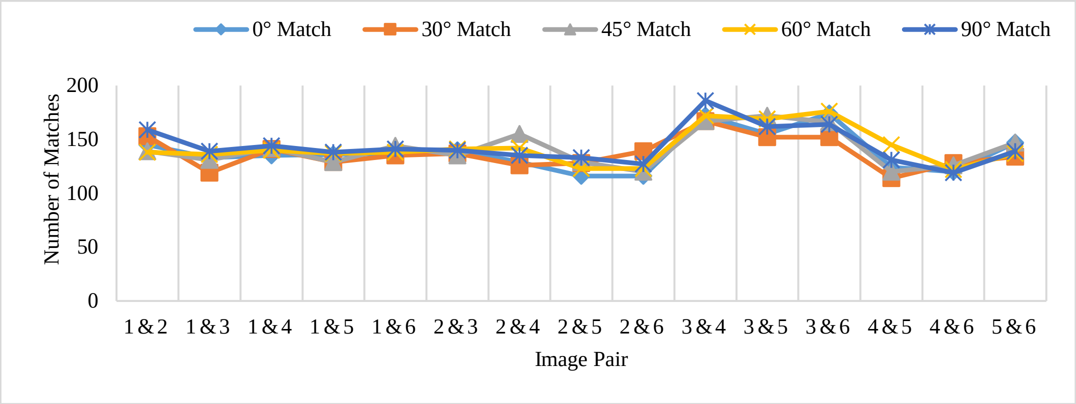
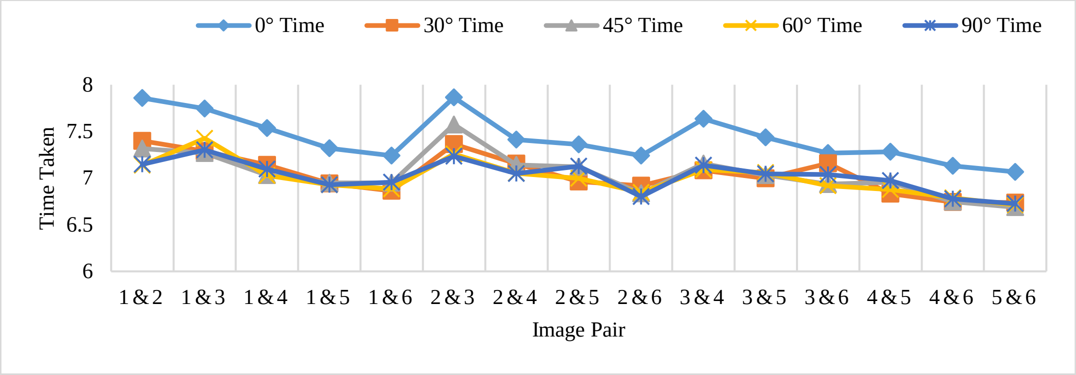
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Wall	
	
Boat	
	

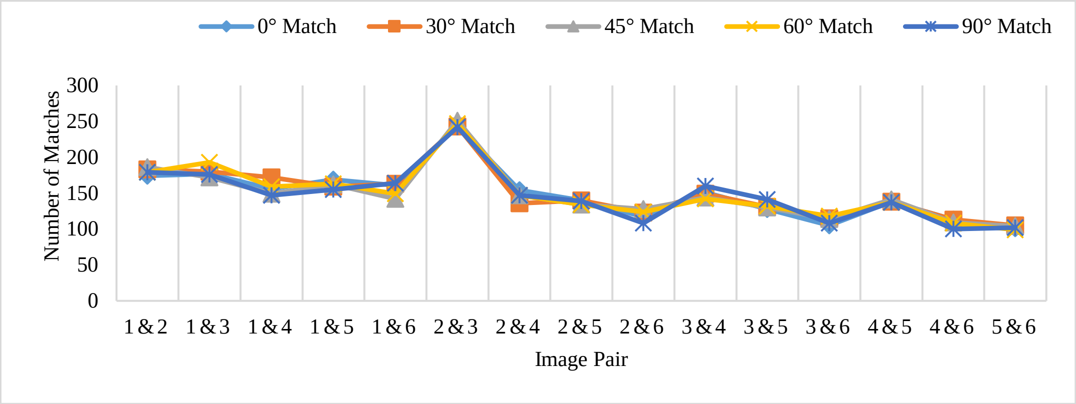
Image-set | **Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]**

Bark





Bikes



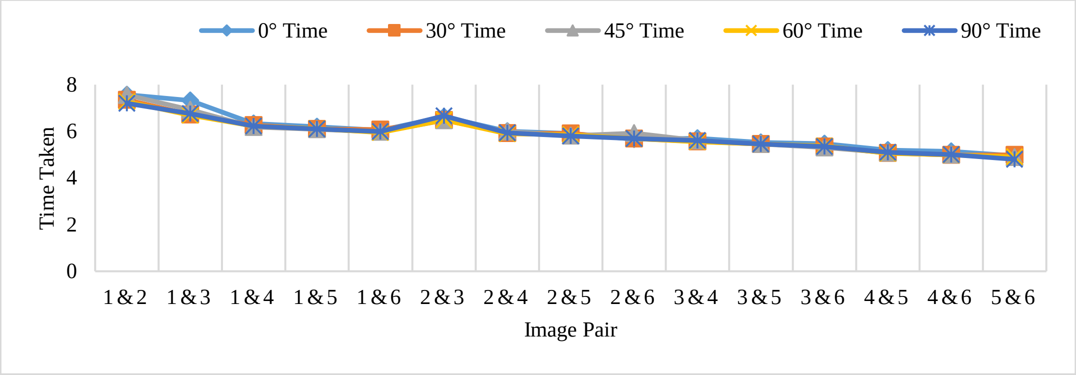


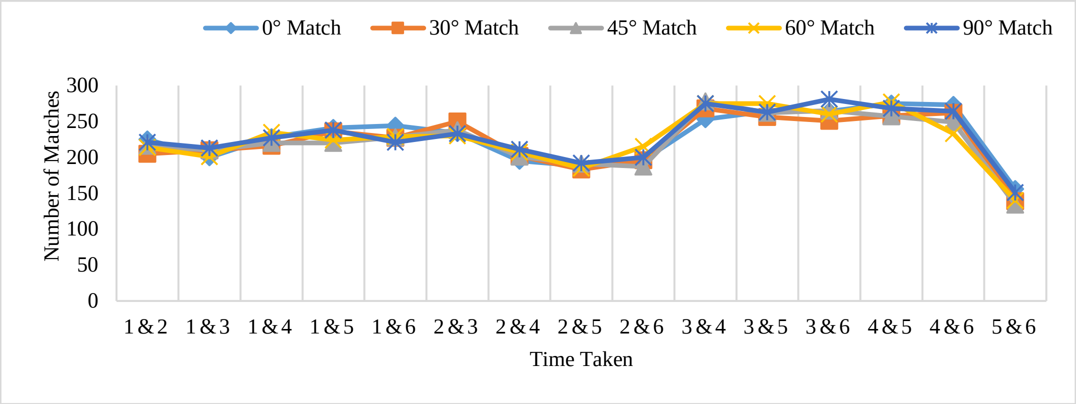
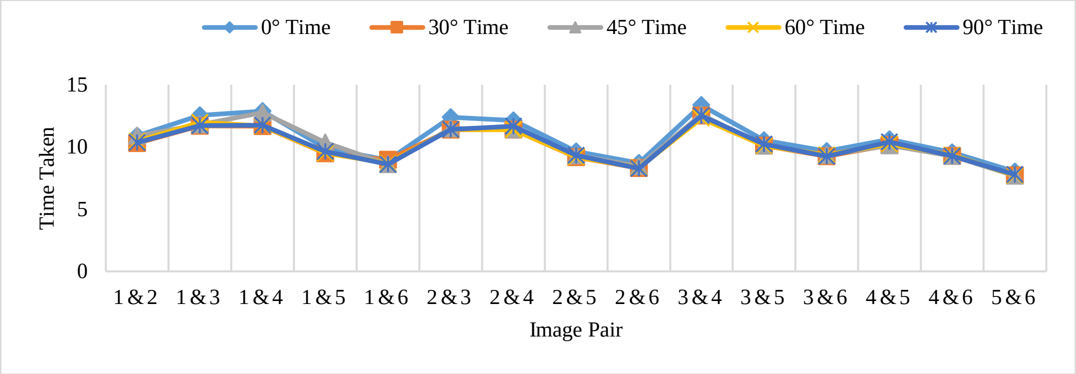
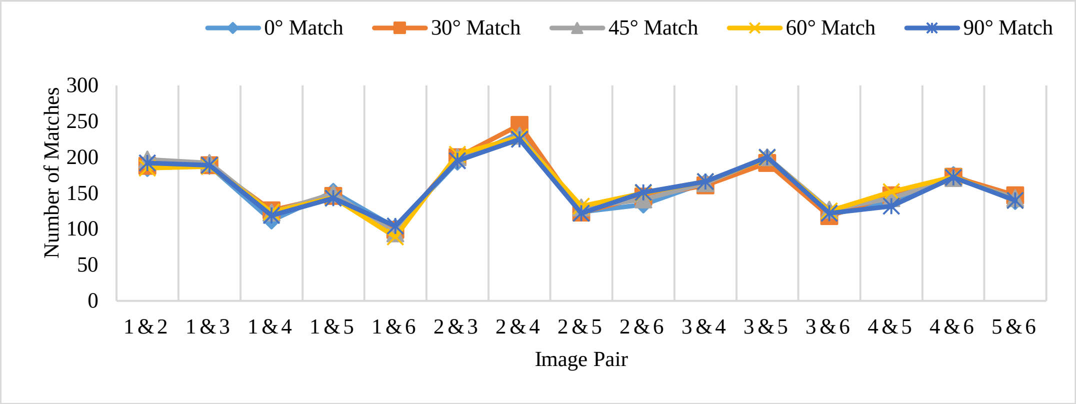
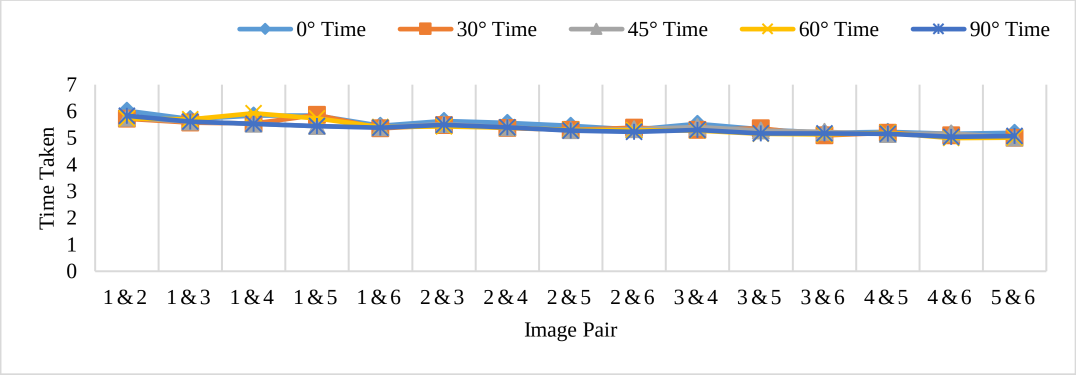
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Trees	 
Leuven	 

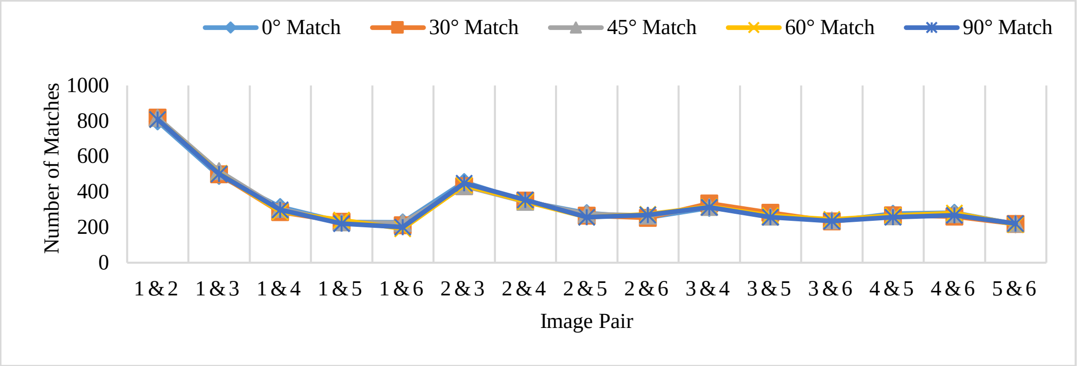
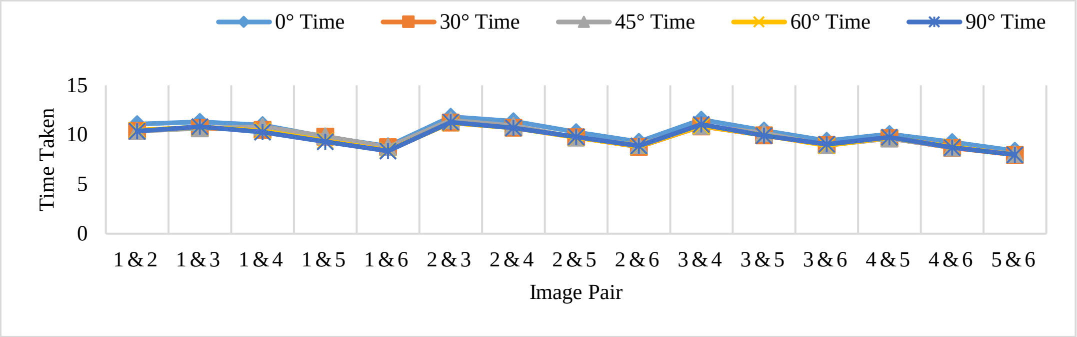
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Ubc	 

Table 7.3. Detailed Observations for 0°, 30°, 45°, 60° and 90° Ellipse Rotations for Circular and Elliptical Sampling using Curve Tracking

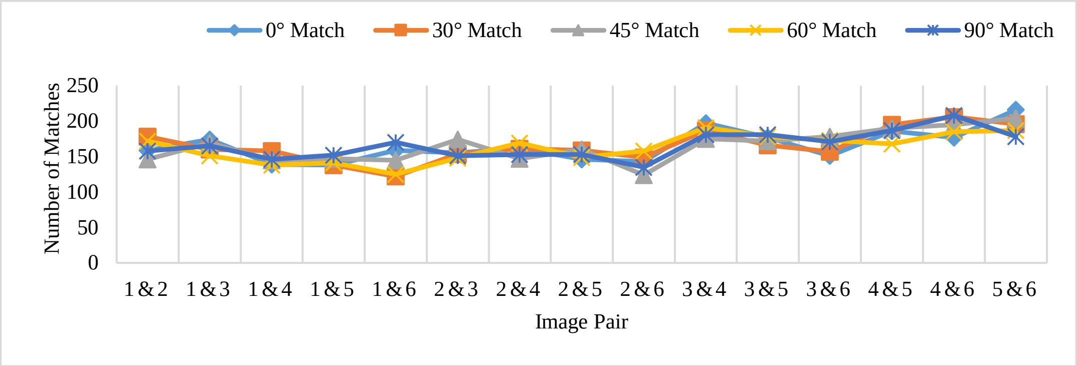
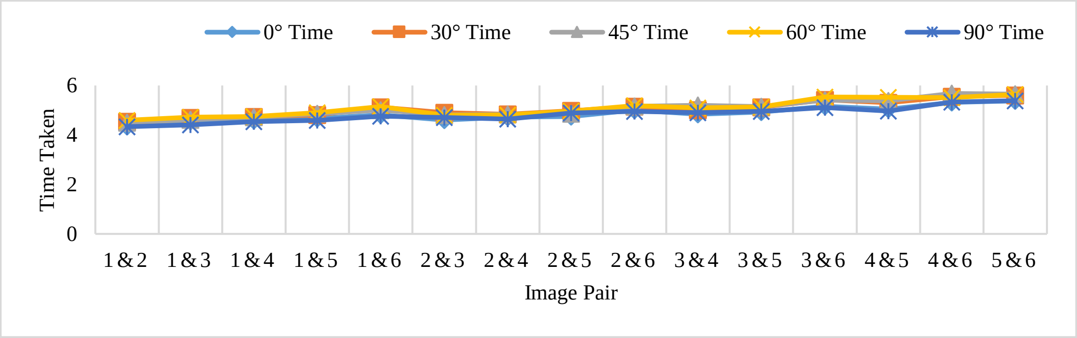
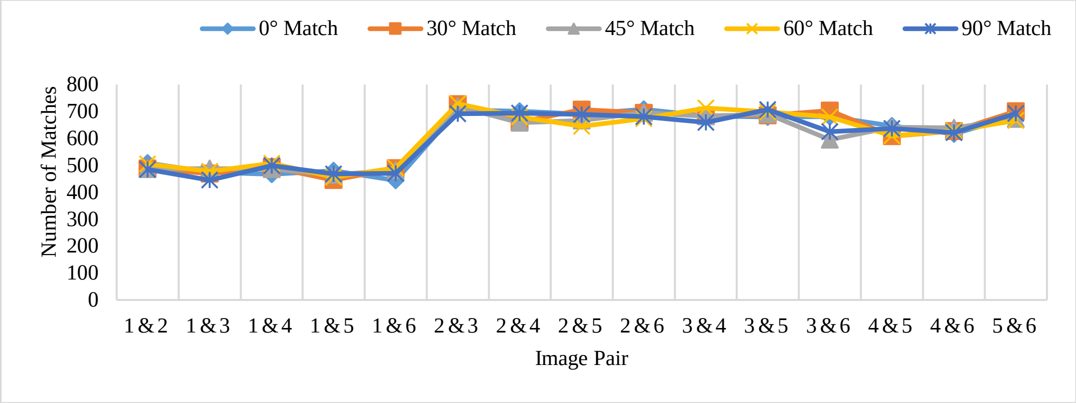
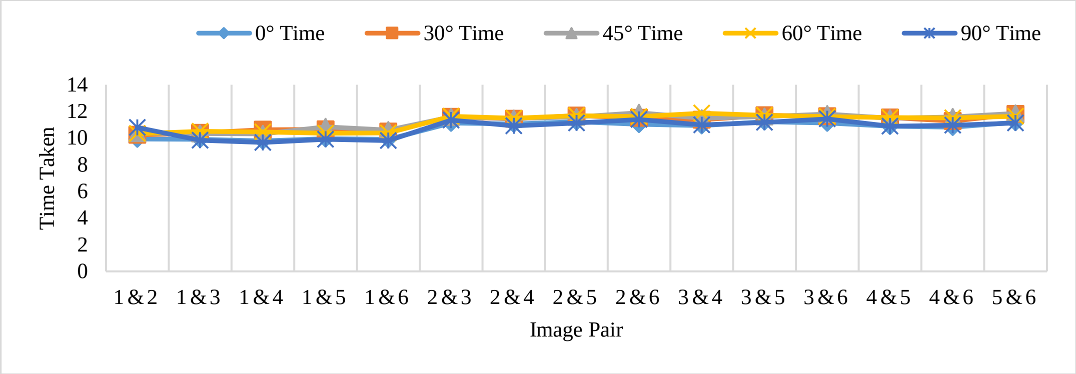
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Graffiti	 

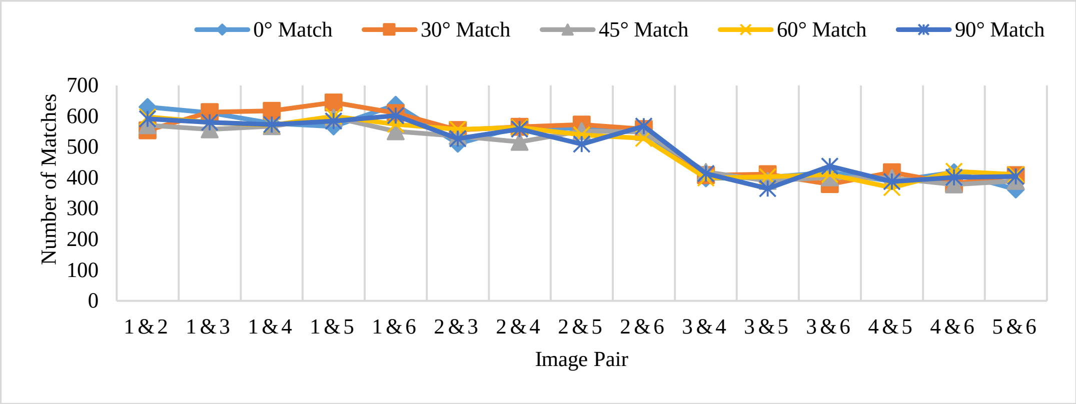
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]

Wall





Boat



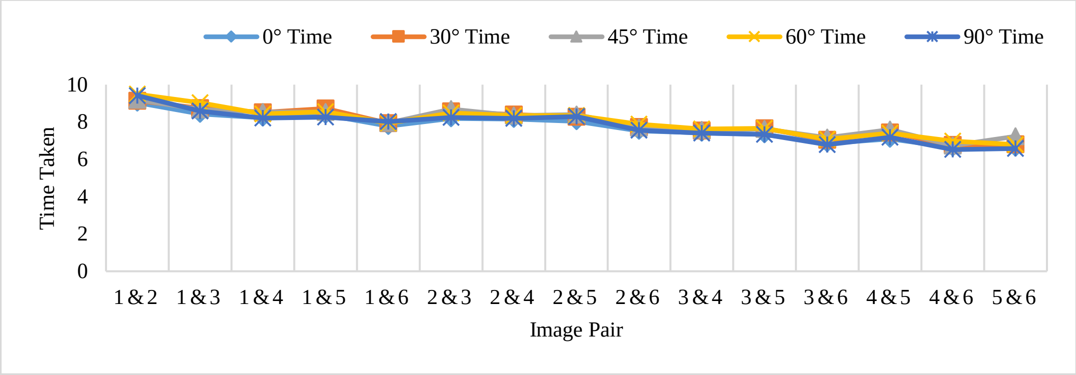
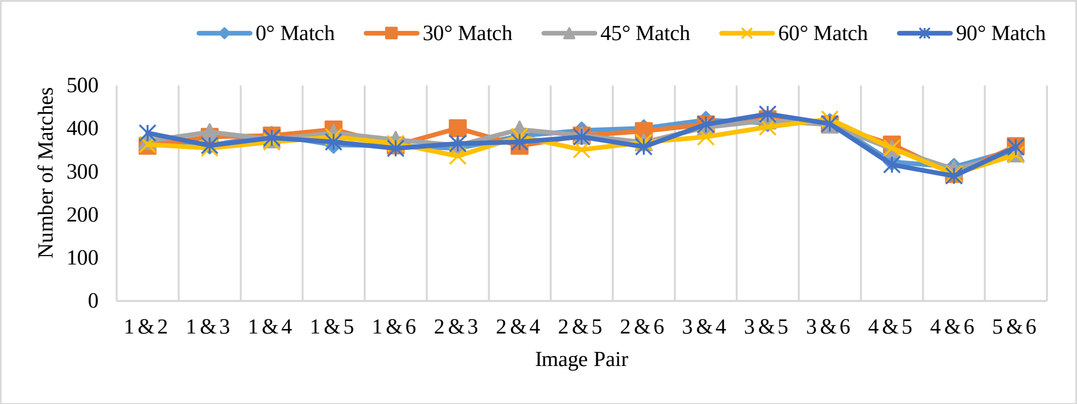
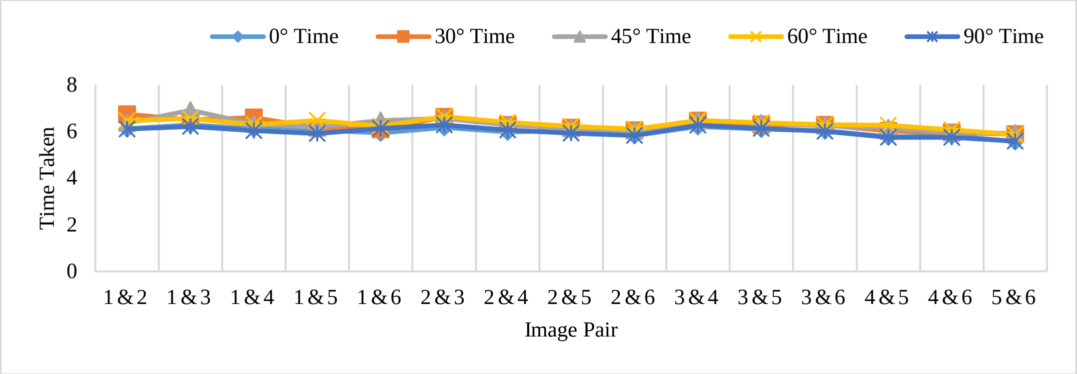


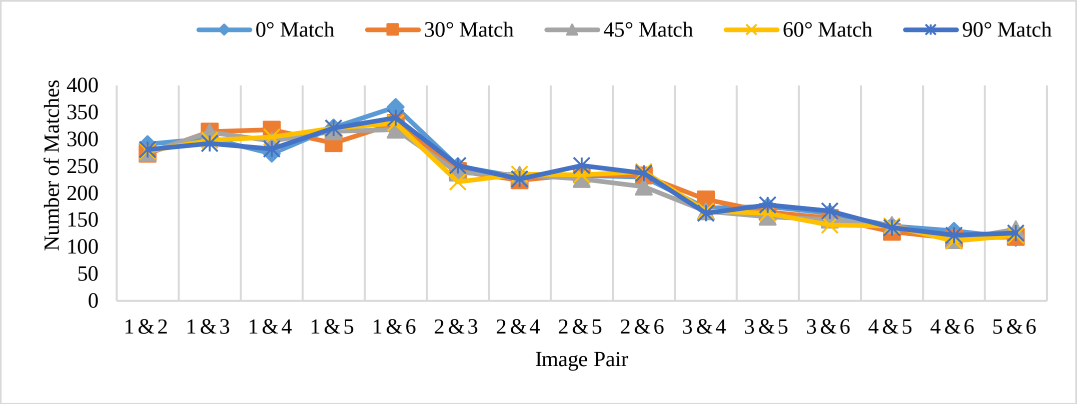
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]

Bark





Bikes



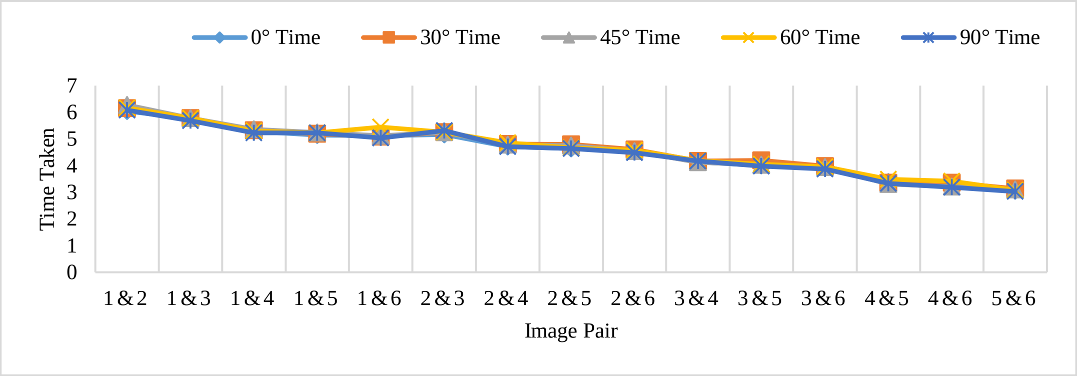
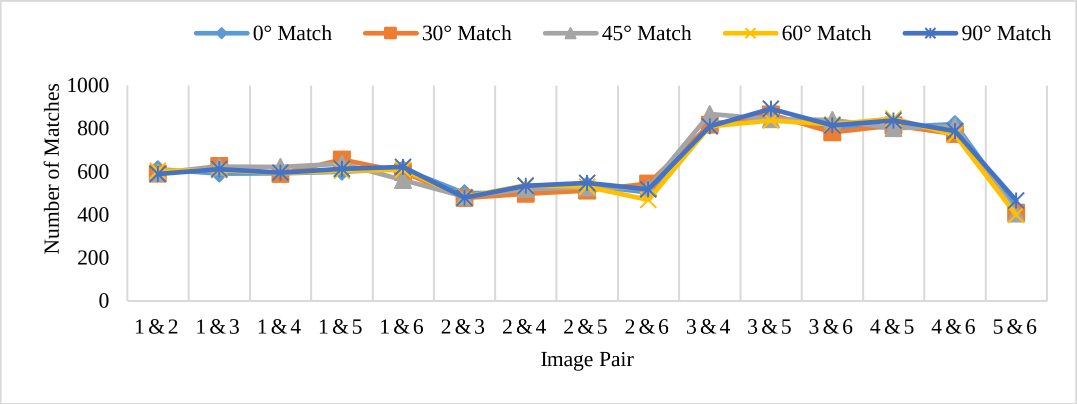
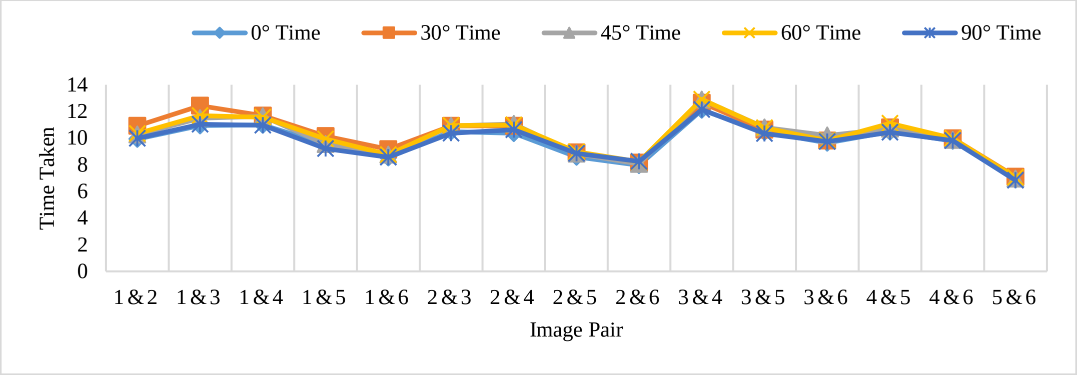


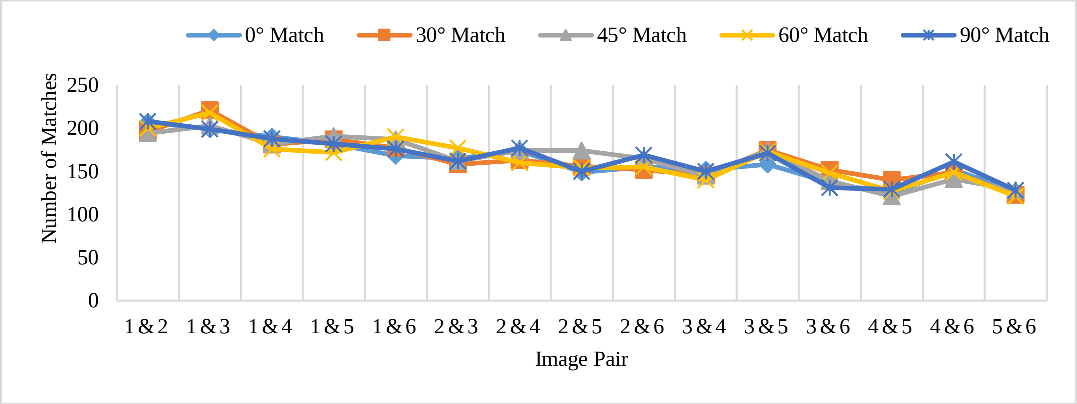
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]

Trees





Leuven



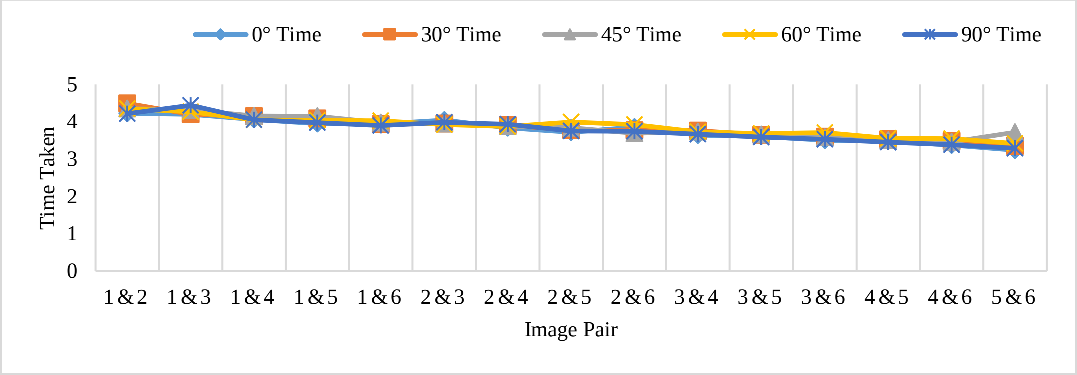


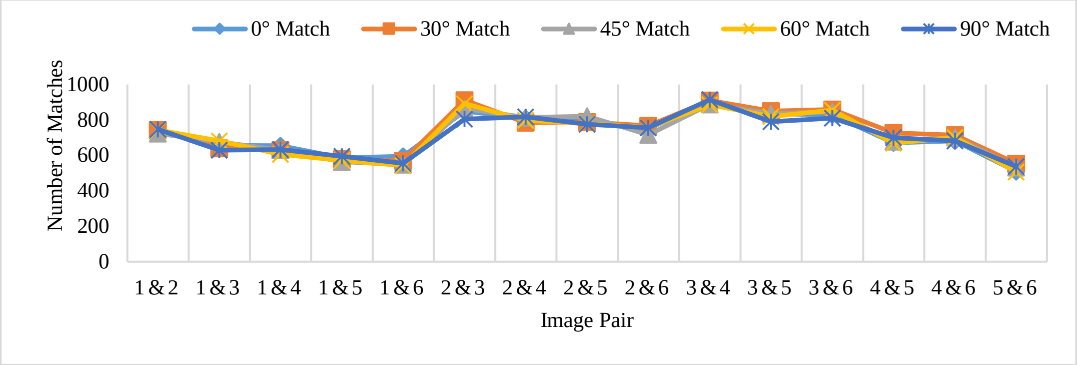
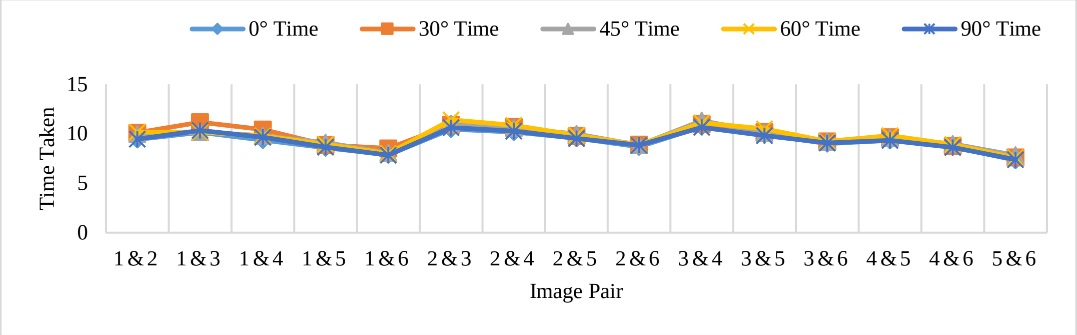
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Ubc	 

Table 7.4. Feature Matching Comparative Performance

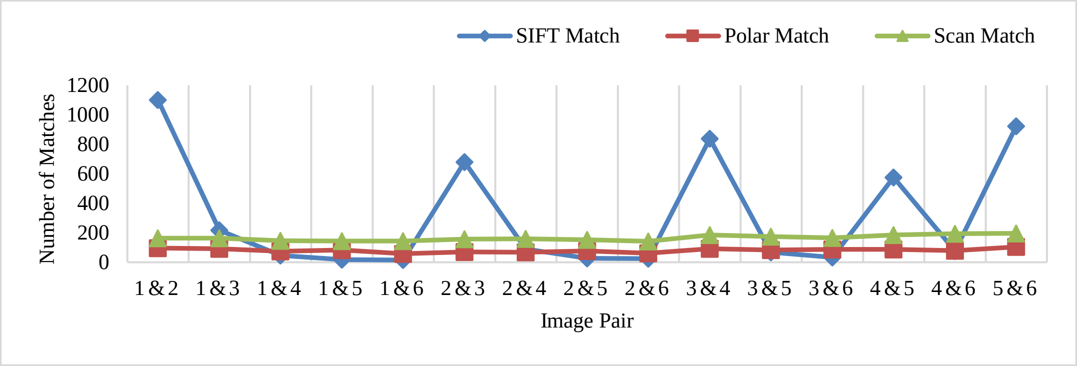
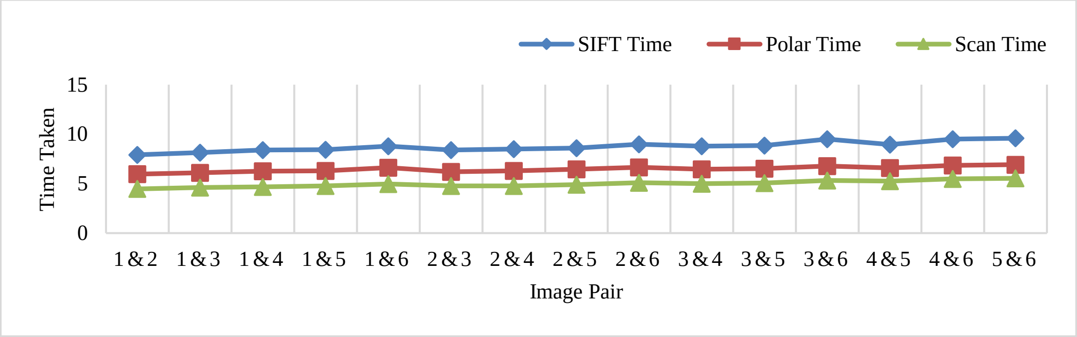
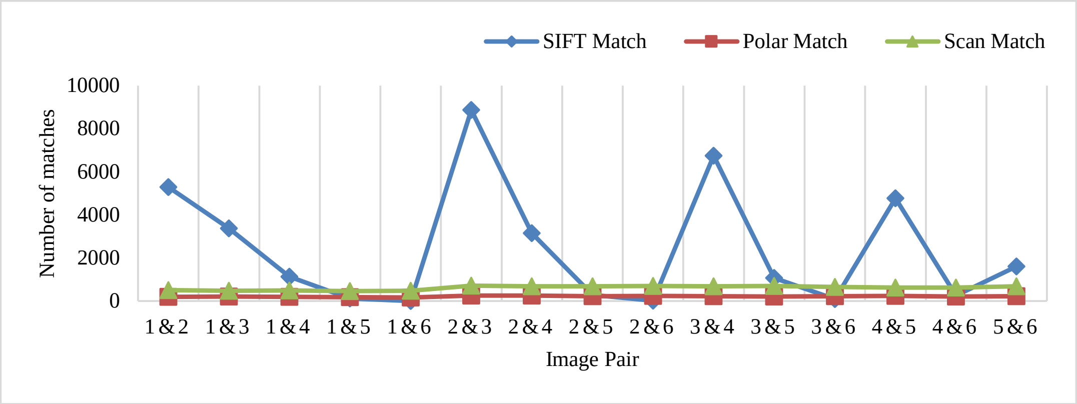
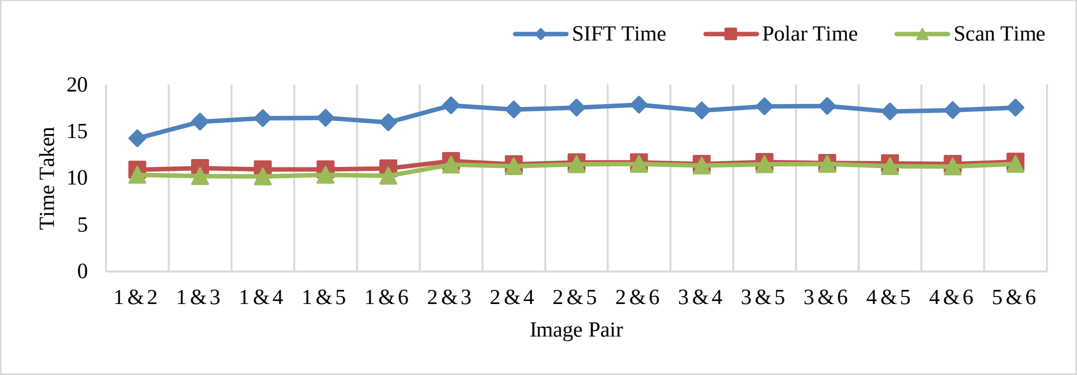
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Graffiti	 

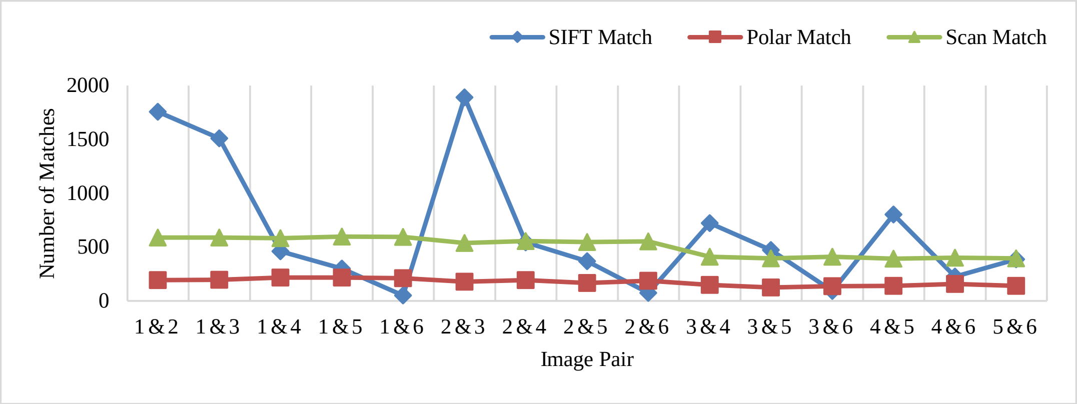
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]

Wall





Boat



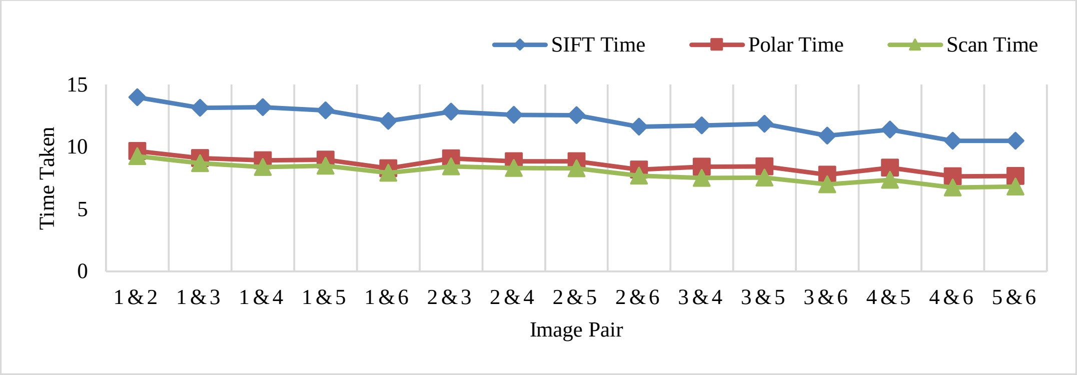


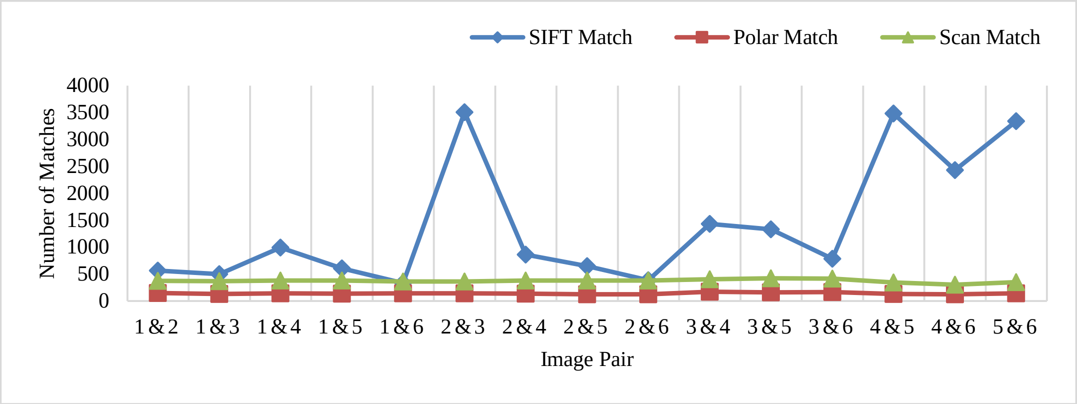
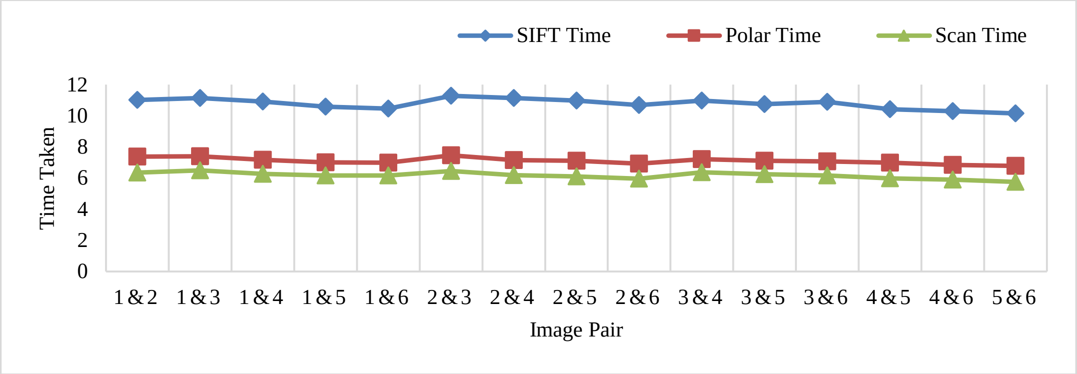
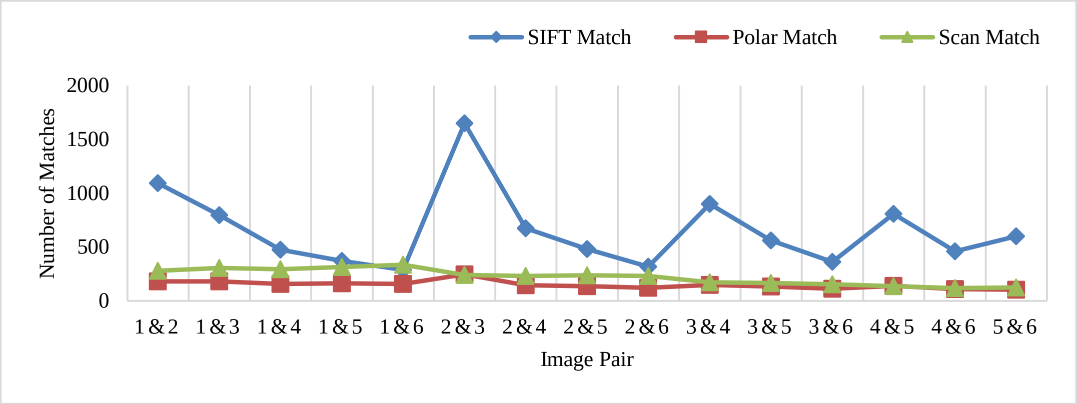
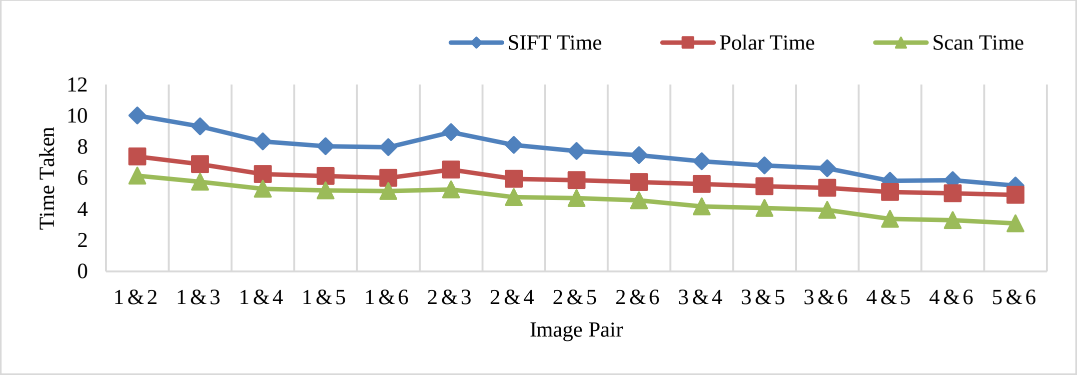
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Bark	 
Bikes	 

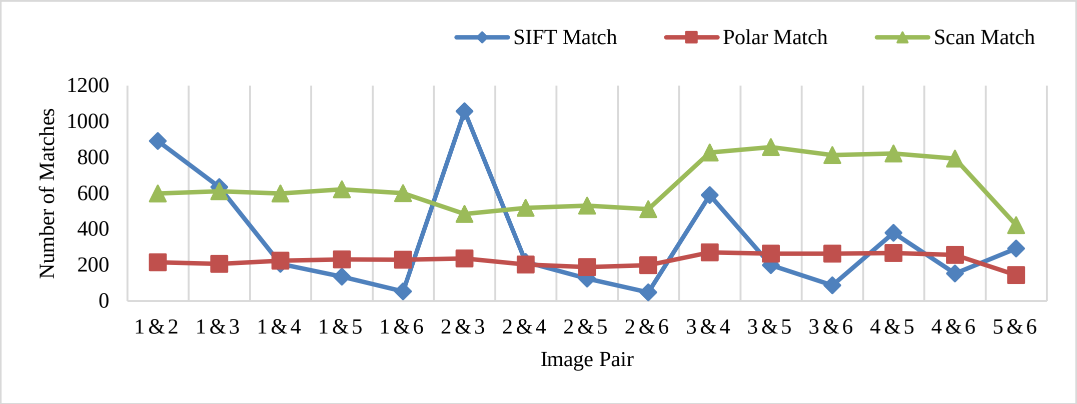
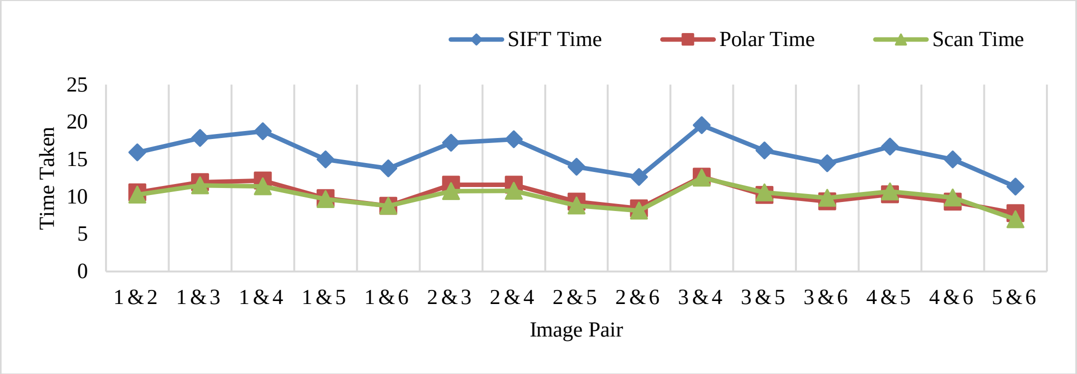
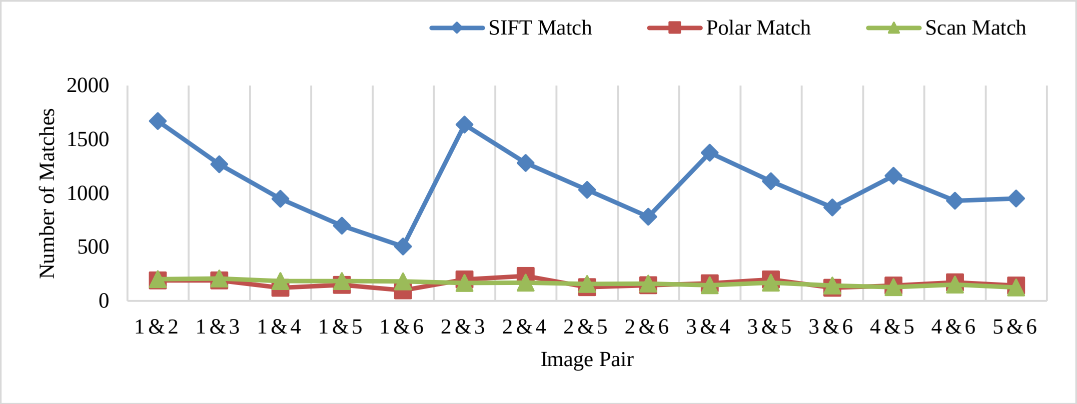
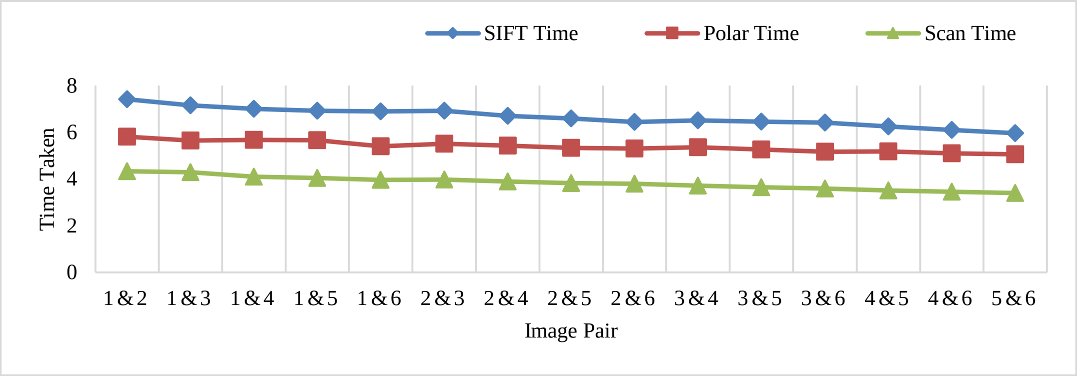
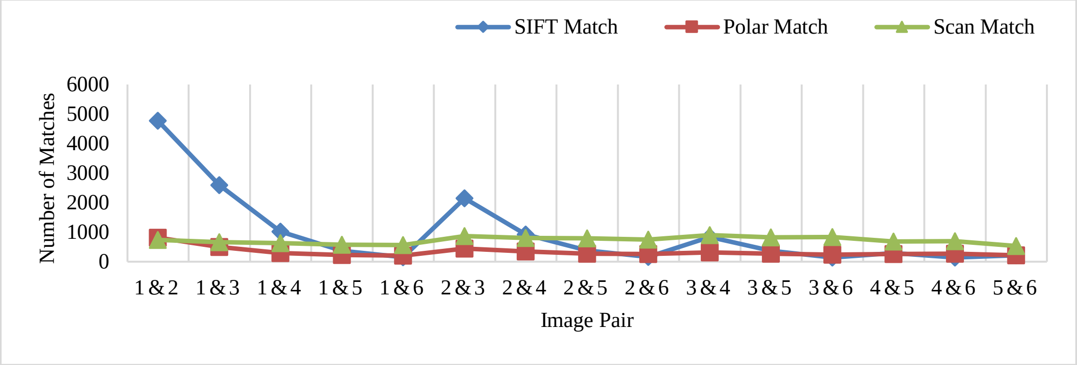
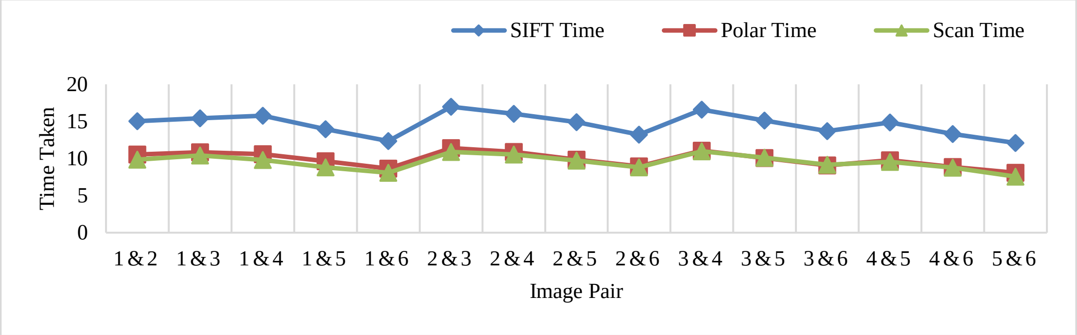
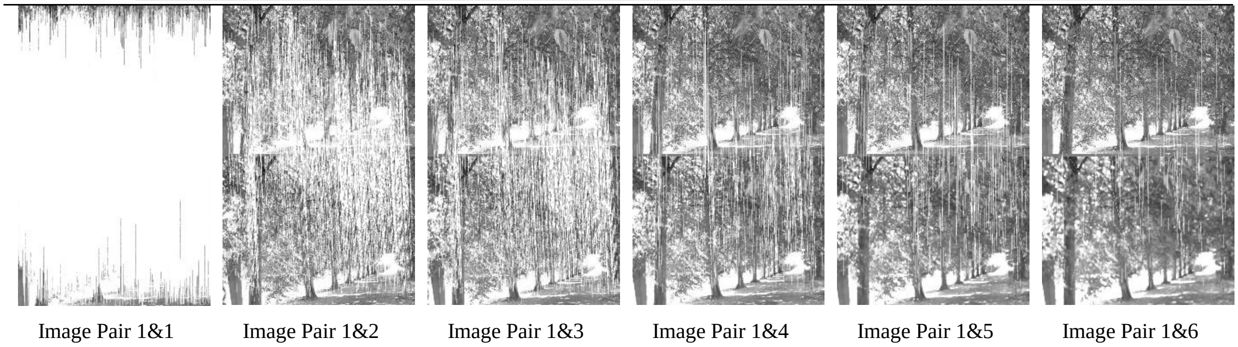
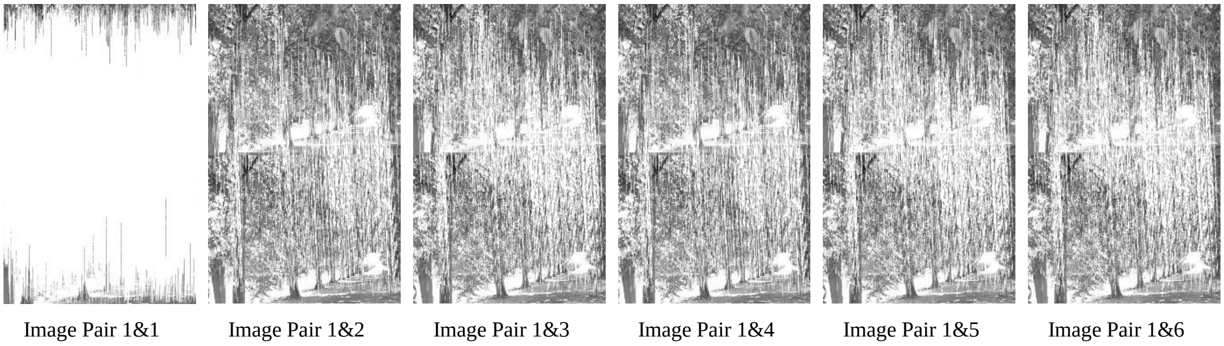
Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Trees	 
Leuven	 

Image-set	Graphs representing number of matches and time taken (in seconds) for all image pairs in each image-set of the Mikolajczyk dataset [Mikolajczyk 2007, Appendix A.1]
Ubc	 



(a) Number of matches obtained for six image pairs in Trees Image-Set using SIFT



(b) Number of matches obtained for six image pairs in Trees Image-Set using the proposed method (using curve tracking)

Fig. 7.5. Number of matches obtained for six image pairs in Trees Image-Set using SIFT and the proposed method

Table 7.5. Average Speed-Up

Image-Set	Graffiti	Wall	Boat	Bark	Bikes	Trees	Leuven	Ubc
Average Speed-up Using Parametric Equations	1.3330	1.4435	1.4545	1.5069	1.3373	1.5299	1.2554	1.4437
Average Speed-up Using Curve Tracking	1.7764	1.5437	1.5286	1.7212	1.5817	1.5751	1.7106	1.5469
Average Speed-up Using Parametric Equations =1.4129								
Average Speed-up Using Curve Tracking =1.6230								

Table 7.5 presents the average speed-up for the image-sets and subsequently also lists the overall speed-up value achieved by the proposed descriptor over the SIFT descriptor. As observed from the table, the proposed descriptor using curve tracking is faster than SIFT by a factor of 1.6 on an average basis while producing sufficient number of matches between an image pair to perform image registration (Table 7.4).

7.8 Result Analysis and Interpretation

This chapter details a novel proposed feature descriptor based on circular and elliptical local sampling of pixels that determines the neighborhood of the extracted feature using circular and elliptical sampling. From the results it is observed that:

1. The number of matches in case of SIFT is higher than the proposed method only in the case when two images describe an image scene that differs with very low degree of change in imaging condition or are of similar quality (quality of the respective image or similarity index between every pair of image in each image-set can be referred from Chapter 4, Section 4.6) Moreover, whenever the quality of images are differing much, the proposed method always exhibits more number of matches in an image pair as compared to SIFT. Therefore, the proposed method is invariant to various parameters like viewpoint change, scale change, variations in image blur and illumination etc.

2. Presenting an argument for low number of correct matches between an image pair by the proposed method, MSLinear-MSER+SIFT feature detector proposed in Chapter 6 is tested on the same image-set [Mikolajczyk 2007, Appendix A.1] and is observed to perform well. In the same chapter, a prototype of an Augmented Reality (AR) system is developed and demonstrated using the same approach and even with approx. twenty matches, the image registration is done correctly (Figure 10 (d) [Gupta and Rohil 2018], also reproduced in Chapter 6, Section 6.5, Figure 6.7(d)). Therefore, it is always not necessary to obtain the number of matches between an image pair in thousands or hundreds to image registration in an AR system correctly. Also, the time taken for image registration in an AR system serve as a main factor

as long as we are able to perform image registration correctly with less or more number of matches between an image pair.

3. The main advantage of the proposed descriptor is fast and robust matching results under varied imaging conditions. It is faster than SIFT by a factor of 1.6 on an average basis while producing sufficient number of matches between an image pair to perform image registration in an AR system.

7.9 Summary

In this chapter a novel feature descriptor based on circular and elliptical sampling of extracted keypoint neighboring pixels is proposed and discussed in detail. The proposed descriptor is based on the fact that local features of an image tend to provide a robust way of image matching if the local descriptor describing the neighborhood of the feature is designed in such a way that it is invariant to large variations in scale, viewpoints, illumination, rotation and affine transformations.