Large scale visual search

Wu, S.

Chapter 2

A Comprehensive Evaluation of Salient Point Methods

As salient point methods can represent distinctive and affine invariant points in an image, various types of salient point methods have been proposed over the past decade. Each method has particular advantages and limitations and may be appropriate in different contexts. In this chapter, we evaluate the performance of a wide set of salient point detectors and descriptors. First, we compare diverse salient point methods with regard to the repeatability of detectors, and the recall and precision of descriptors. Next, we integrate the salient point methods with the framework of fully affine space and evaluate their performance under major viewpoint transformations. The presented comparative experimental studies can support researchers in choosing an appropriate detector and descriptor for their specific computer vision applications.

2.1 Introduction

Salient point methods which can describe meaningful, stable, and representative local features in an image have become prevalent in diverse areas in computer vision, such as object and scene recognition [77, 78], 3D object reconstruction [79], visual tracking [80, 81] and multimedia information retrieval [3, 18, 82, 83, 84, 85, 86, 87, 88]. Most of the salient point algorithms contain two parts: a detector and a descriptor. The detector locates a set of distinctive points which can be invariant to various transformations (e.g., scaling, translation, viewpoint changes), and the descriptor encodes the important information from the local patch centered on the salient point into a feature vector, which makes it possible to reliably match correspondences across different transformations of the same object or the same scene.

Typically, object recognition, 3D reconstruction and visual tracking mainly rely on the correctly matched correspondences between two compared images. These applications start by extracting local descriptors from each image and insert the obtained local descriptors into an index space for efficient correspondence matching. The RANSAC algorithm [89] is further adopted to eliminate outlier matches and to estimate the homography between the compared images. Therefore, a salient point detector with high repeatability and a local descriptor with discriminatory power is required for these applications.

However, accurate correspondence matching under large viewpoint changes is still a major challenge, because greater image viewpoint transformations result in a significant decrease of saliency and repeatability of salient points. Yu et al. [90] proposed to use the framework of fully affine space to overcome this issue. The basic idea behind the framework of fully affine space is that the projective transformation induced by camera motion around a smooth surface can be approximated by an affine transformation. A notable method is ASIFT which generates all image views in the whole affine space and extracts SIFT local features in these synthetic images to increase the matching precision. As the high dimensionality of the SIFT descriptor leads to a high computational complexity in the framework of fully affine space, we combine the recent lower computational

complexity salient point algorithms with the framework of fully affine space and evaluate their performance under the extreme viewpoint changes.

This chapter is an extension of our previous projects [87, 88] which provide a comparison guide of recently proposed salient point detectors and descriptors. The main contributions of this chapter are summarized as follows:

First, the repeatability performance and the computational cost of each salient point detector are presented.

Second, the efficiency and accuracy of both the real valued descriptors and binary string descriptors in terms of recall and precision on two benchmark datasets are evaluated.

Third, we calculate the accuracy and time complexity of each salient point method in the framework of fully affine space such that researchers could make a trade-off between precision and efficiency under extreme viewpoint changes.

2.2 Background

Early research on salient point methods mainly focused on finding high variance or corner points in the image. One of the first detectors was developed by Moravec [91] and it is defined according to the average intensity changes in different directions within the local region around a point. The Harris corner detector [92] defines a corner structure point, if its second-moment matrix has two large eigenvalues. The similar Hessian corner detector [93] determines a corner point in the image, if it is the local extrema of the Hessian matrix determinant. As both the Harris and Hessian detectors find the corner points at a fixed scale, the Harris-Laplacian and Hessian-Laplacian [94, 95] are designed to be scale invariant. Harris-Laplacian and Hessian-Laplacian locate corner candidates on each level of the scale space. Those points for which the Laplacian simultaneously attains local extrema over scales are selected as corner points. The FAST [96] detector identifies the corner points according to the criterion whether a set of

contiguous pixels in a circle are all brighter or all darker than the intensity of the centre point.

Since conventional corner point detectors are only invariant to scale, translation, and noise, affine covariant region detectors were developed to reduce the influence of viewpoint changes. The Harris-Affine detector and the Hessian-Affine detector [97] find the initial candidate points by using the Harris-Laplacian corner detector and Hessian-Laplacian corner detector, respectively, and then fit an elliptical region to each point via the second moment matrix of the intensity gradient. MSER [98] computes the connected binary regions through a large set of multiple thresholds, and the selected regions are those that maintain unchanged shapes over these thresholds. As edges are typically rather stable structures that can be detected over a range of image changes, EBR [99] starts by detecting corner points in an image and identifies the affine covariant region of each point by exploiting the edge information present nearby. IBR [100] detects intensity extrema at multiple scales and captures the intensity pattern along rays emanating from each extremum to define a region of arbitrary shape. The region of IBR is delineated by the image points defined over these rays where the intensity suddenly increases or decreases, and then uses an ellipse to fit the region. However, the operation of elliptical region fitting in the affine covariant detector could result in partial information loss.

Recent salient point methods focus on the repeatability and precision of the detector, as well as the distinctiveness, computational efficiency and low memory requirement of the local descriptor. The most representative one is SIFT, which efficiently builds the scale space by employing the Difference of Gaussians to approximate the Laplacian of Gaussians and represents the local descriptor using a gradient orientation histogram. Meanwhile, some variants of SIFT are proposed with the aim to increase the discrimination of the SIFT descriptor. PCA-SIFT [101] utilizes PCA to reduce the dimension of the original SIFT descriptor to further speed up the process of local descriptor matching. Color-SIFT [102] takes the color gradients, rather than intensity gradients in the local region around the salient point to generate the feature. Rank-SIFT [103] adopts a data-driven approach to learn a ranking function to sort the salient points such that the

unstable points can be discarded. Root-SIFT [104] adds a square root operation to the normalized SIFT features and uses the Hellinger kernel to increase the matching accuracy. DSP-SIFT [105] generates the descriptor through pooling the gradient histogram across different domain sizes of each salient point into a feature and it even outperforms the high level convolutional neural network feature [48]. Affine-SIFT (ASIFT) [90] is proposed with the aim to be perspective invariant and it does this by simulating images under various views to cover the whole affine space and extracting SIFT descriptors in all these simulated images for matching. Different from these variants of SIFT, other approaches target on improving the efficiency of scale space establishment or accuracy of salient points localization. For example, the SURF detector makes use of a box-filter and the integral image to speed up the scale space building. The ORB and BRISK detectors use a Gaussian image pyramid to efficiently establish the scale space. As the construction of scale space by linear multi-scale Gaussian pyramids easily results in the blurring and the loss of boundary details, KAZE [106] combines a nonlinear scale space with additive operator splitting (AOS) and special conductance diffusion to reduce noise while retaining the object boundary structure. The advantage of the nonlinear scale space in KAZE is that it could provide more accurate positions for salient points.

In order to meet the requirements of real time systems and devices with limited computational and storage resources, binary string local descriptors were recently introduced. Binary string representations make use of a pixel-pair intensity comparison to generate the binary code. The resulting binary code holds some significant advantages: first, the operation of intensity comparison is fast, the memory requirement of binary codes is low and matching binary codes via the Hamming distance is much faster than the Euclidean metric. A representative descriptor is BRIEF, which randomly samples a set of pixel-pairs from a Gaussian distribution in the smoothed local patch around the salient point and produces a binary string descriptor via the intensity comparison of pixel-pairs. The ORB descriptor integrates rotation invariance into BRIEF by estimating the orientation via the intensity centroid method. Additionally, ORB makes use of an unsupervised learning scheme to select pixel-pairs, rather than the random sam-

pling of BRIEF. BRISK and FREAK generate the binary string descriptors by comparing pair-wise intensities over a pre-defined pattern, a concentric ring-based sampling pattern and a retina sampling pattern, respectively. In contrast to those hand-crafted patterns, learning based approaches are proposed with the goal of closing the performance gap with real valued representations while maintaining the benefits of binary representations. BinBoost learns a set of hash functions using boosting and projects the image patch into a binary representation. LATCH proposes to learn patch triplet arrangements in the image and compares the intensity of triplet patches rather than the intensity of pixel-pairs to generate the binary codes.

Several related reviews present the performance evaluation of various salient point methods. Schmid et al. [107] uses the measure of "repeatability rate" and "information content" to evaluate the performance of different salient point detectors. Mikolajczyk et al. [108] made a performance evaluation of local descriptors by measuring the accuracy of matching and recognition. Accuracy and computational efficiency trade-offs [109] have been studied where different indexing structures were employed (such as approximate KD-trees). Heinly et al. [110] and Figat et al. [111] investigate the recall and precision of recent binary string representations under different image deformations. Gauglitz et al. [81] presents a comparison of different salient point methods on video object tracking. Moreels and Perona [112] made a performance evaluation of both feature detectors and descriptors on 3D object matching. Mukherjee et al. [113] made a performance evaluation for each combination of recent detectors and descriptors on object matching. To our knowledge, our review is the first one that evaluates the viewpoint invariance of each salient point approach in the fully affine space.

2.3 Overview of Evaluated Salient Point Methods

The aim of salient point methods is to extract distinctive invariant features from images that can be used to perform image correspondence matching and to per-

form the image representation. Recent salient point methods consist of four main procedures: the first step is to establish the scale space and find the extrema across all scales to achieve scale invariance. The second step is to determine the locations of the extrema and to define a local region for each according to the scale information. Then, each defined region is normalized and assigned a domain orientation to be rotation invariant. Finally, the region content is rotated based on the calculated orientation, after which, the discriminative information in the rotated region is encoded into a local descriptor. The existing schemes of local descriptor generation can be categorized into hand-crafted schemes and automatically learned schemes. The recent literature focuses more on the automatic learning of local descriptors. The learning based schemes usually optimize an objective function to generate robust and distinctive local descriptors. In particular, the most common objective functions are designed to minimize the distance between the descriptors from the same 3D coordinate (scale and location) or same class label extracted under varying imaging conditions and different viewpoints, meanwhile, maximizing that distance between patches from different 3D coordinates or different class labels. Table 2.1 gives an overview of all the evaluated salient points approaches in the experiments section.

2.3.1 SIFT (detector/descriptor)

SIFT proposed by Lowe [14] is the most popular salient point approach. The implementation of SIFT begins by building the Gaussian scale space which approximates the Laplacian-of-Gaussian function by the computationally efficient Difference-of-Gaussian function. It searches extrema over all scales to identify the potential salient points. Since the extreme points are detected in discrete scale space, it then uses the derivative of the Taylor expansion of the DoG function to determine the accurate scale and location for each salient point and simultaneously rejecting unstable extrema with low contrast. Furthermore, because a poorly defined extremum in the DoG function has a large principal curvature across the edge but a small one in the perpendicular direction, a Hessian matrix is employed to compute the principal curvatures and to eliminate points which are

Table 2.1: Overview of the evaluated salient point approaches in this chapter by Detector (Det.), Descriptor (Desc.), Scale Space, Orientation, and Descriptor Generation.

Methods	Det./Desc.	Scale Space	Orientation	Descriptor Generation
SIFT	yes/yes	Difference of Gaussian	local gradient histogram	local gradient histogram
SURF	yes/yes	box-filter	local Haar-wavelet responses	local Haar-wavelet responses
MSER	yes/no	no	no	no
HESSIAN-AFFINE	yes/no	no	no	no
FAST	yes/no	no	no	no
CenSurE	yes/no	bi-level filter	no	no
GFTT	yes/no	no	no	no
KAZE	yes/no	nonlinear scale space	no	no
BRIEF	no/yes	no	no	intensity comparison of pair-wise pixels in the random sampling pattern
ORB	yes/yes	Gaussian image pyramid	intensity centroid calculation	oriented BRIEF descriptor
BRISK	yes/yes	Gaussian image pyramid	average of the sum of the local gradient	intensity comparison of pair-wise pixels in concentric circles pattern
FREAK	no/yes	no	average of the sum of the local gradient	intensity comparison of pair-wise pixels in retina sampling pattern
BinBoost	no/yes	no	no	projection by learned hash function
LATCH	no/yes	no	no	intensity comparison of patch triplet arrangements

potentially sensitive to edge responses. To be invariant to rotation, an orientation is assigned to the obtained stable points according to the local gradient orien-

tation histogram within a region around the point. In addition, it accumulates the orientations of a 16×16 neighborhood of sample points around the salient point location into orientation histograms by summarizing the contents over 4×4 sub-regions. A 128-dimensional descriptor vector is finally generated to represent each point.

2.3.2 SURF (detector/descriptor)

SURF is an efficient and robust scale and rotation-invariant method proposed by Bay et al. [12] with the aim for fast salient point location and descriptor generation. SURF is based on a Hessian matrix, where the components of the Hessian matrix are generated by convolution of the Gaussian second-order derivative with the image pixels. Box-filters together with integral images are exploited to approximate the Hessian matrix which is used to measure the salient points. The Gaussian scale space of SURF is established computationally efficiently by up-scaling the size of the box-filter. The extrema of the determinant of the Hessian matrix are selected as salient points and the scale and location are updated through an interpolating process. Each of the obtained salient points is assigned an orientation which is estimated by summing the horizontal and vertical Haar-wavelet responses within a sliding orientation window covering an angle of 60 degrees. For the SURF descriptor generation, first the square region centered on and oriented along the salient point is divided into a number of 4×4 sub-square regions. Then, it calculates the value and absolute value of Haar-wavelet responses along horizontal and vertical directions within each sub-region. Finally the total 64-dimensional $(4 \times 4 \times 4)$ descriptor can be generated efficiently by making use of the integral image.

2.3.3 MSER (detector)

Maximally stable extremal regions (MSER), proposed by Matas et al. [98], is an affine invariant region detector. MSER computes the connected binary regions through a large set of multiple thresholds, and the selected regions are those

that maintain unchanged shapes over a range of thresholds. During the affine invariant regions detection, the area of each connected component is stored as a function of intensity and the "maximally stable" ones are selected as candidates by analyzing the changes of function values for each potential region. The final maximally stable extremal regions are the ones that maintain an unchanged or similar function value over a large range of multiple thresholds. The shape of each obtained region is further estimated by elliptical regions by computing the eigenvectors of their second-moment matrix. Then the local neighborhoods are normalized into circular regions to achieve affine invariance.

2.3.4 HESSIAN-AFFINE (detector)

The Hessian-Affine region detector proposed by Matas et al. [97] is based on the Hessian matrix. A related variant of the Hessian-Affine detector is the Harris-Affine detector which employs the Harris detector to find the salient points. Since the second derivatives in the Hessian matrix offer strong responses on blobs and ridges, the extrema of the determinant of the Hessian matrix are searched by applying non-maximum suppression using a 3×3 window over the entire image. To deal with the scale invariance, given an extremum location, a scale-dependent signature function is defined on its local neighborhood and the corresponding scale can be determined by searching for scale-space extrema of the signature function. The estimation of the affine shape is applied to each extremum and an elliptical region is fit around each point using the second moment matrix of the intensity gradient. Finally, the affine region is normalized into a circular region. In this chapter, the improved Hessian-Affine detector [114] is used, which proposes the gravity vector assumption to fix rotation uncertainty.

2.3.5 FAST (detector)

The high-speed corner point detector named features from accelerated segment test (FAST) was proposed by Rosten and Drummond [96]. The simple scheme of FAST corner detection is based on a circle (the radius of the circle is three

pixels) of sixteen pixels around the candidate point. If there exists a set of twelve contiguous pixels in the circle which are all brighter or all darker than the intensity of the candidate point pixel value plus a threshold, the point will be classified as a corner point. However, this scheme has a limitation for sampling less than twelve pixels and the efficiency of the corner detector depends on the distribution of corner appearances. To overcome the above weaknesses, a machine learning approach is employed on training sets to establish a decision tree for fast and accurate corner detection. Moreover, the issue of multiple features being detected adjacent to one another, can be solved by applying non-maximum suppression on the detected candidate corner points.

2.3.6 CenSurE (detector)

The scale invariant center-surround salient point detector (CenSurE) is proposed by Agrawal et al. [115]. CenSurE determines the salient points by exploiting the extrema of the Hessian-Laplacian matrix across all scales and locations. Inspired by SIFT which uses the Difference of Gaussian function to approximate the Laplacian of Gaussian function, CenSurE employs a simplified center-surround filter called bi-level filter to approximate the Laplacian of Gaussian for fast computation. The CenSurE detector computes the response of the bi-level filter at all locations and all scales, and detects the extrema in a local neighborhood (based on the non-maximum suppression method, which is the same as SIFT and SURF). For each obtained extremum, the accurate location of the potential points can be determined directly, since the responses are calculated on the original image. Furthermore, through computing the Harris measure for the potential points, those points with weak corner responses will be eliminated.

2.3.7 GFTT (detector)

Good feature to track (GFTT) is a salient point detector proposed by Shi and Tomasi [116], which is derived from an image motion model. GFTT is used as a method for feature selection, tracking and monitoring, and it performs well

under affine image transformations. According to the proposed feature selection criteria, a candidate point is accepted if it is defined as a good feature which can be tracked well. GFTT is based on the Harris corner detector and it defines points with large eigenvalues of a special matrix as corners. To ensure the robustness of corners, potential corners with minimum eigenvalues less than a threshold are eliminated. Candidates which are closer than a certain distance-threshold to a strong corner are also rejected.

2.3.8 KAZE (detector)

Most salient point approaches (SIFT, SURF) construct the scale space based on linear multi-scale Gaussian pyramids. However, the Gaussian function does not respect the natural boundaries of objects and smoothes the details and noise at the same level, which leads to loss of localization accuracy and distinctiveness. The use of a nonlinear scale space is expected to reduce noise but to retain the object boundary structure in order to obtain accurate positions of salient points. The traditional method is based on the forward Euler scheme for solving nonlinear diffusion but requiring significant computational complexity. Therefore, the nonlinear scale space in KAZE [106] proposes to use the additive operator splitting algorithm (AOS) for efficient nonlinear diffusion filtering. The framework of KAZE first convolves the image with a Gaussian kernel of standard deviation, and then builds the nonlinear scale space in an iterative way using the AOS scheme. Based on the response of the scale-normalized determinant of the Hessian matrix at multiple scale levels, the extrema responses can be detected as salient points by non-maximum suppression and the position of the salient points can be estimated with sub-pixel accuracy using quadratic fitting.

2.3.9 BRIEF (descriptor)

Binary robust independent elementary features (BRIEF), designed by Calonder et al. [117], uses an efficient binary string descriptor to represent the salient points. With regard to the BRIEF descriptor generation, Gaussian smoothing

is first utilized to reduce the effect of noise sensitivity such that it can achieve good performance in complex scenes. The value of each bit in the binary string depends on the intensity comparison of two points inside the local patch centered on each salient point (provided by detectors, as BRIEF is a descriptor), i.e., if the value of first point is larger than the second then it is set to "1", otherwise to "0". The pixel-pairs sampling patterns are randomly selected using a Gaussian distribution (locations that are closer to the center of the patch are preferred) around the smoothed patch center. Similarity of two binary string descriptors is calculated using the Hamming distance, which is significantly more efficient than the common Euclidean distance. The BRIEF descriptor is not rotation invariant.

2.3.10 ORB (detector/descriptor)

ORB (oriented FAST rotated BRIEF) [118] is a combination of the FAST detector and the BRIEF descriptor. The ORB detector applies the FAST corner detector to find potential salient points. However, FAST does not offer scale information, and has large responses along edges. ORB builds a scale pyramid of the image and keeps the top N number of keypoints by the Harris corner measure at each level in the scale pyramid. The scale information is the scale factor of the specific level of the image pyramid. The direction of points is computed using their intensity centroid [15]. The intensity centroid approach assumes that the intensity of a keypoint is offset from its center, and it can be used to compute the moments of a patch and also to find its centroid. The orientation is defined as the direction of the vector between the keypoint location and the centroid position in the patch. The generation of the ORB binary string descriptor also uses the comparison of intensities of pixel-pairs based on the oriented BRIEF descriptor. Additionally, a combination of earning and greedy search is introduced for de-correlating BRIEF features under rotational invariance, leading to a better performance.

2.3.11 BRISK (detector/descriptor)

In the implementation of BRISK [16], the scale space is also based on the simple image pyramid. For the salient points detection, BRISK first employs AGAST [119] which is essentially an extension for accelerated performance of the FAST detector to locate the potential keypoints at each layer in the scale space. Then it measures their saliency via comparing FAST scores with respect to its eight neighbors in the same layer and 3×3 neighbors in the layer above and below. The local maxima of FAST score points will be identified as salient points. The accurate location and scale of each salient point are obtained in the continuous domain via refinement of quadratic function fitting. BRISK presents a novel sampling pattern which consists of sample points equally distributed on concentric circles centered around the salient point. It weights each respective circle in the pattern with a standard deviation Gaussian, and then divides all the sampling-point pairs in the pattern into short-distance pairs and long-distance pairs based on the defined threshold. The direction of the patch is determined via the average of the sum of the local gradients of all selected long distance pairs. The bit-vector descriptor is assembled by comparing all the short-distance pair-wise intensities.

2.3.12 FREAK (descriptor)

Similar to the BRISK scheme which uses a pre-defined pattern to estimate the orientation and for generating the binary string features, the FREAK [17] descriptor is based on the retina sampling pattern. The retina sampling pattern simulates the distribution of ganglion cells over the retina which reduces exponentially with the distance to the center. The orientation is calculated mainly based on selected pairs with symmetric receptive structure with respect to the center point of the patch. The direction of the patch is also obtained by averaging the sum of the local gradient of the defined pairs in the structure. In the descriptor creation of FREAK, less correlated pairs over a retina pattern are selected based on a similar learning algorithm performed in ORB and the intensities are then compared to generate the binary strings.

2.3.13 BinBoost (descriptor)

The approach of BinBoost is a supervised learning framework to generate a low dimensional but highly discriminative local binary representation. A hash function is implemented as a sign operation on a linear combination of non-linear weak classifiers which are gradient based image features, and the hash function is learned by the optimization of a loss function with the aim to reduce the Hamming distances between binary representations of similar patches in training data, while increasing the Hamming distances between binary representations of dissimilar patches in the training data.

2.3.14 LATCH (descriptor)

LATCH extracts learned patch triplet arrangements in a salient region, and compares the intensity of the triplet patches to form the binary string codes. The learning procedure of LATCH is based on training data with labels, and possible triplet arrangements are extracted from the training data. It defines the quality of an arrangement by summing the number of times it correctly yielded the same binary value for positive pairs and different values for negative pairs. A candidate arrangement is selected, if its absolute correlation with all previously selected arrangements is smaller than a certain threshold such that the obtained triplet arrangements are with less correlation.

2.4 Fully Affine Space Framework

The main idea behind the framework of fully affine space is that the projective transformation induced by camera motion around a smooth surface can be approximated by an affine transformation, and it consists of all possible affine distortions caused by the change of the camera's optical axis orientation from a frontal view. The reason to employ this scheme is that we expect two salient points to be correctly matched under certain perspective transformations. The

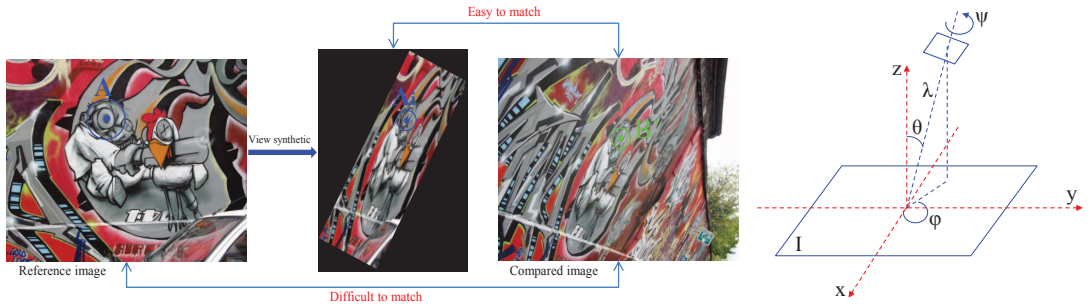

Figure 2.1: (a) Illustration of the synthetic view generation for correct correspondence matching. (b) Illustration of the camera model under affine transformation.

fully affine space framework could also be viewed as a data augmentation technology which expands the training data by systematically adding transformed samples. The transformed samples are typically generated to be label-preserving such that they can encourage the system to become invariant to different transformations. As illustrated in Figure 2.1 (a), it is difficult to match point A in the reference image to point B' in the compared image, but it is easy to match point A_i which is located in the deformed view image arising from viewpoint changes to point B'. Generating a deformed view image can be modeled by an affine transformation of the original image, where the affine transformation can be decomposed into a zoom, rotation, tilt, and rotation around the optical axis [120].

$$\begin{aligned} A &= \lambda R(\psi) T_t R(\varphi) \\ &= \lambda \begin{bmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\varphi) & -sin(\varphi) \\ sin(\varphi) & cos(\varphi) \end{bmatrix} \end{aligned} \tag{2.1}$$

where $\lambda > 0$ is a zoom factor, $R(\psi)$, $R(\psi)$ are rotations and t is the tilt, as shown in Figure 2.1 (b). The parameter $\psi \in [0, 2\pi)$ denotes the angle of planar rotation around the optical axis. The angle θ between the z axis and the optical axis is called the latitude and $t = 1/cos(\theta)$. The angle $\varphi \in [0, \pi)$ between the x axis and the projection of the optical axis is called the longitude. Then, each synthesized view can be described by the parameters of λ (zoom), $R(\psi)$

(planar rotation), $t = 1/cos(\theta)$ (the rotation angle of the latitude) and $R(\varphi)$ (the rotation angle of the longitude). The simulated latitudes θ correspond to tilts $t = 1, a, a^2, ..., a^n$, with $a > 1$, and a is set $\sqrt{2}$ for a good compromise between accuracy and efficiency. Each tilt in the fully affine space is a t sub-sampling. The number of rotated images for each tilt is *2.5t*. Thus, the complexity is proportional to the amount of tilts. As the fully affine space can significantly increase the precision of correspondence matching, we integrated the recent salient point methods with the fully affine space framework and evaluated their accuracy and efficiency.

Generally, the Nearest Neighbor Distance Ratio (*NNDR*) is used as the matching strategy to find the similar descriptors in the image pairs. *NNDR* defines that two points will be considered to be matched only if $||D_A - D_B||/||D_A - D_C|| < threshold$, where D_B is the first and D_C is the second nearest neighbor to D_A. However, for the matched correspondences in the specific fully affine space, lots of repeatable salient points are present in the synthetic view images which results in the *NNDR* to be close to one for some correct correspondences, thus, those correct correspondences will be easily defined as false according to the threshold (less than one) of *NNDR*. In order to address this issue, we propose to use the *K-order NNDR* matching strategy for correspondence matching in the fully affine space. Unlike the standard *NNDR* which only takes the first and second nearest neighbors into account, *K-order NNDR* fully explores the relationship among the group of K nearest neighbors, such that it can address the problem faced by *NNDR* but without increasing the computational cost. The *K-order NNDR* is characterized as follows:

$$K\text{-}order\ NNDR = R_k \times (1 - \frac{w}{\prod_{i=2}^{k-1} R_{i-1}}) \tag{2.2}$$

where $R_k = ||D_A - D_1||/||D_A - D_k||$ and D_k is the k^{th} nearest descriptor to D_A. w is a weight which is set to 0.01 in the experiments to achieve good performance.

2.5 Experimental Setup

The experimental environment for the evaluation is a Intel Quad Core i7 Processor (2.67GHz), 12GB of RAM, 64 bit OS. The implementations of Hessian-affine, KAZE, LATCH and BinBoost are from the authors, others are implementations from OpenCV. The parameters of each salient point method were set to the defaults and we used 8 randomized forests in the KD-tree index, 20 hash tables in the multi-probe LSH index. Our evaluation implementations are available at: http://press.liacs.nl/researchdownloads/.

2.5.1 Datasets

The performance of salient point detectors and descriptors is evaluated on the Oxford dataset proposed by Mikolajczyk and Schmid [108] and the dataset designed by Fischer et al. [121]. The Oxford dataset contains eight groups, and each group consists of six image samples (a total of 48 images) with various transformations (rotation, viewpoint, scale, JPEG compression, illumination and image blur). The Fischer dataset is a large scale dataset that includes 16 groups and each group contains 26 images generated synthetically by applying 6 types of transformations (zooming, blurring, illumination, rotation, perspective and nonlinear). Some examples of each dataset used for evaluation are illustrated in Figure 2.2.

2.5.2 Evaluation Criteria

The criteria employed to measure the performance of the salient point methods in each application are summarized in Table 2.2. We follow the commonly used evaluation protocol [87, 107, 108, 122]. The score of repeatability, recall and precision, and the number of correct correspondences are used as evaluation criteria in the experiments.

(a)

(b)

Figure 2.2: Examples from each dataset for the evaluation of salient point methods. (a) Examples from the Oxford dataset [108] used for the evaluation of the accuracy of correspondence matching. (b) Examples from the Fischer dataset [121] used for the evaluation of the accuracy of correspondence matching.

Table 2.2: Overview of the evaluation criteria used in the experiments.

Criteria	Function description
Repeatability [107]	Measures the performance of the detector: the higher the repeatability score, the better the performance.
Recall and precision [108]	Measures the accuracy of correspondence matches: a distinctive descriptor shows high recall at any precision.
Number of correct correspondences	Total amount of correct correspondences between two compared images, a robust method shows a high score.

2.6 Results and Discussions

2.6.1 Detector Evaluation

In this section, we test the performance of each salient point detector on the benchmark Oxford dataset [108] and the Fischer dataset [121]. The evaluated salient point detectors are: SIFT, SURF, ORB, BRISK, FAST, CenSurE, GFTT,

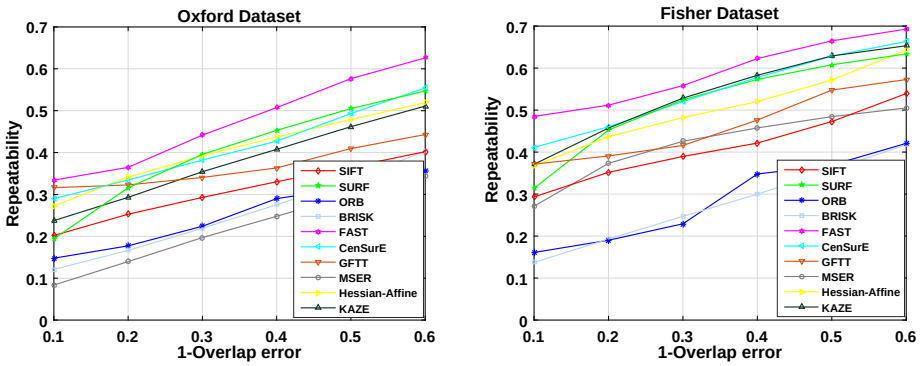

Figure 2.3: The performance evaluation of salient point detectors based on the criterion of repeatability.

KAZE, MSER and Hessian-Affine.

An important evaluation criterion from the research literature is repeatability [107]. The repeatability score is calculated as the ratio between the number of correspondences and the minimum of m_1 and m_2, where m_1, m_2 denote the number of points in the reference and the query images after projecting the reference image points by the ground truth homography and discarding those points outside the common area, respectively.

$$repeatability = \frac{C(m_1, m_2)}{min(m_1, m_2)} \tag{2.3}$$

$C(m_1, m_2)$ is the number of correspondences between m_1 and m_2. An overlap error is used to identify the correspondence. For a keypoint region in the query image which is the nearest one to a projection keypoint region in the reference image by using homography: if the ratio between the intersection of the two regions and the union of the two regions is larger than the overlap error, it will be considered as a correspondence. We compute the average repeatability scores on the whole dataset, respectively, thus, the detection performance of each method can be estimated in a comprehensive perspective. The trend of average repeatability under varying overlap errors (in the range from 0.4 to 0.9) is shown in Figure 2.3.

The evaluation results based on the two datasets illustrate that an increase in

the repeatability scores is clearly indicated when the value of 1-overlap error becomes larger. We can also notice that the FAST detector had the highest repeatability and the ORB and BRISK detectors obtained the lowest scores. The detectors SURF, Hessian-Affine, KAZE, and CenSurE have a similar rank on both datasets. The performance of the nonlinear scale space detector KAZE reveals superior results to the well known SIFT detector. All detectors can reach a stable and acceptable performance when the value of overlap error is 0.5, so the overlap error will be set at 0.5 to identify the correspondences in the following experiments.

Since the salient point detection mechanism in each salient point method is based on a different scheme, which results in a different computational complexity, and a different set of feature points can be extracted from the same image, time costs should be compared statistically. We applied different types of detectors to various test images, in order to determine statistically significant results. The average number of detected points and the time cost of the compared salient point methods are shown in Table 2.3.

Table 2.3: Comparison of average number of detected points and detection time

Method	Oxford Dataset [108]		Fischer Dataset [121]	
	Average number of points	Time cost(ms) of 1000 points	Average number of points	Time cost(ms) of 1000 points
SIFT	5472	40	5607	52.02
SURF	5368	22.8	6138	34.8
ORB	497	27.0	490	29.5
BRISK	1498	20.2	1607	19.3
FAST	15857	0.31	17388	0.27
CenSurE	915	20.1	920	25.1
GFTT	1000	31.2	984	35.6
MSER	750	341.8	793	360.5
HESSIAN-AFFINE	3680	247.8	3693	260.1
KAZE	2940	59.8	3108	73.5

The results listed in Table 2.3 reveal that the most efficient detector is FAST. FAST detected the largest number of salient points on both datasets, which is almost ten times higher than what was obtained by other detectors. FAST defines the salient points according to simple intensity comparisons, thus, the time cost is only 0.31 ms for a total of 15857 points on the Oxford dataset [108] and 0.27 ms for 17388 points on the Fischer dataset [121]. The most time-consuming detectors are MSER and Hessian-Affine, because they need do the ellipse fitting for each salient point. The detectors SIFT, SURF, ORB, BRISK and KAZE all contain scale space and rotation estimation procedures. KAZE builds the nonlinear scale space in an iterative way using the AOS scheme which is much more time consuming than the linear scale space calculation. As SURF, ORB and BRISK speed up building the scale space, they are more efficient than the SIFT detector.

2.6.2 Descriptor Evaluation

The Oxford and Fischer datasets are also utilized in the local descriptors evaluation. Note that some of the salient point detectors from the previous section do not define descriptors and are not compared here. In order to make an objective comparison of different salient point descriptors, SURF was applied as the salient point detector, as the SURF detector is scale invariant and it provides a high repeatability score according to its performance in the detector evaluation. We combined SURF detectors with local descriptors including SIFT, SURF, ORB, BRIEF, BRISK, FREAK, BinBoost and LATCH. The evaluation starts by extracting salient point features from the reference images and establishing a KD-tree or LSH index space for the obtained local features. Then, we extract features from the query image and match them against the features from each reference image based on the approximate nearest neighbor search. In the matching procedure, a KD-tree index is established for real value descriptors and the Euclidean distance is used for matching, while binary string descriptors are matched in an LSH index using the Hamming distance.

The *NNDR* is used as the matching strategy to find similar descriptors in image pairs. In addition we use recall and 1-precision [108] (not to be confused with precision@1) as criteria to measure the performance of various salient point descriptors. Recall denotes the number of correct matches with respect to the number of correspondences between two compared images, and the precision is the number of correct matches with respect to the total number of matches.

$$recall = \frac{\#correct\_matches}{\#correspondences} \tag{2.4}$$

$$precision = \frac{\#correct\_matches}{\#total\_matches} \tag{2.5}$$

We varied the value of the threshold in the *NNDR* to obtain the curves of the tendency of the average recall vs. 1-precision under each transformation. Figure 2.4 and Figure 2.5 show the results on each dataset. We also provide the area under the recall vs. 1-precision curve, averaged over all image transformations in each dataset, as shown in Table 2.4 and Table 2.5. A distinctive descriptor would give a high score of area under each curve (AUC).

Table 2.4 and Table 2.5 summarized the results of AUC under each transformation as well as the average score. SIFT, BRISK, and FREAK show good performance for all image degradations on the two datasets. Looking at the performance on the Oxford dataset [108], all descriptors perform better on image changes (blur, illumination and JPEG compression) than on affine deformation changes (rotation, scale and perspective). The descriptors created by SIFT, BRISK, FREAK, SURF, and BinBoost are more robust and distinctive than ORB, BRIEF and LATCH under affine deformation. This is mainly because the BRIEF descriptor only conducts pixel-pair intensity comparisons and is not rotation invariant, while the ORB descriptor as an improved BRIEF descriptor is rotation invariant and resistant to noise, but not scale invariant. The LATCH descriptor uses the same scale information causing it not to be scale invariant. For the scores under changes of blur and JPEG compression, the BinBoost descriptor obtains the lowest score, thus, it is more sensitive to those types of noise. An illumination change

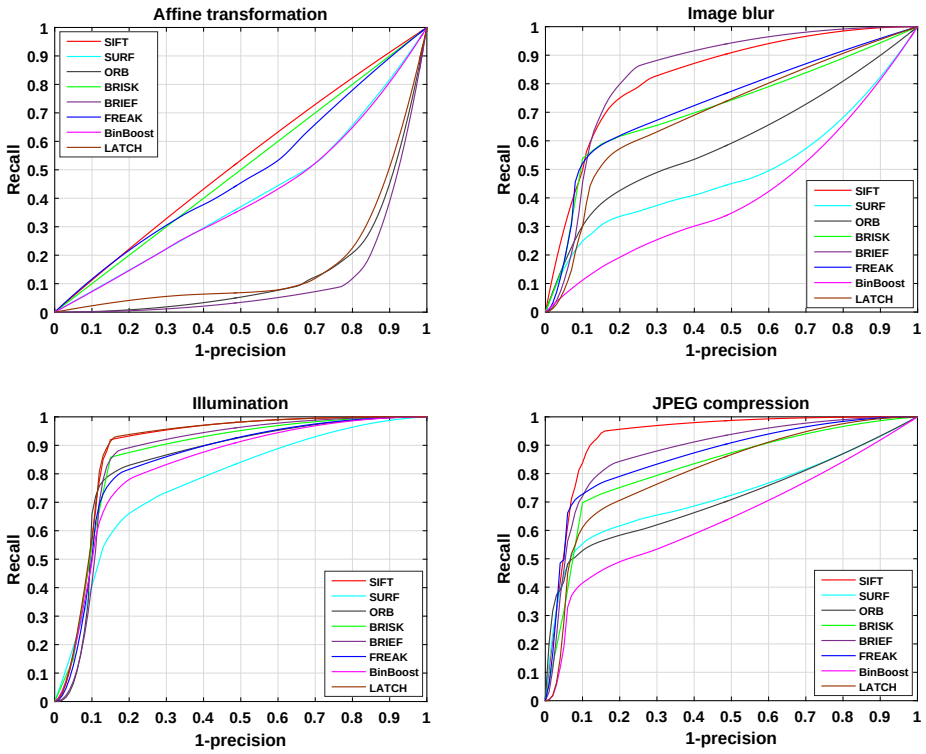

Figure 2.4: Comparison of various descriptors using recall vs 1-precision under different image degradations. The evaluation results are for the Oxford dataset [108].

has a big influence on the SURF descriptor, while the other descriptors are robust to illumination changes and show scores close to each other. The evaluation results on the Fischer dataset [121] show the same tendency under the changes of image blur and perspective when compared to the results on the Oxford dataset [108]. In addition, the descriptors of ORB, BRIEF and LATCH also show their weakness under the change of image zoom.

The time and memory complexity of local descriptor extraction is also statistically analyzed in this section. The average time costs for generating local descriptors based on the Oxford dataset [108] and the Fischer dataset [121] are shown in Table 2.6. It is clear that binary string descriptors are more efficient than real valued descriptors in terms of memory requirement. The SIFT descriptor has the highest time complexity, followed by the BinBoost descriptor. The SURF

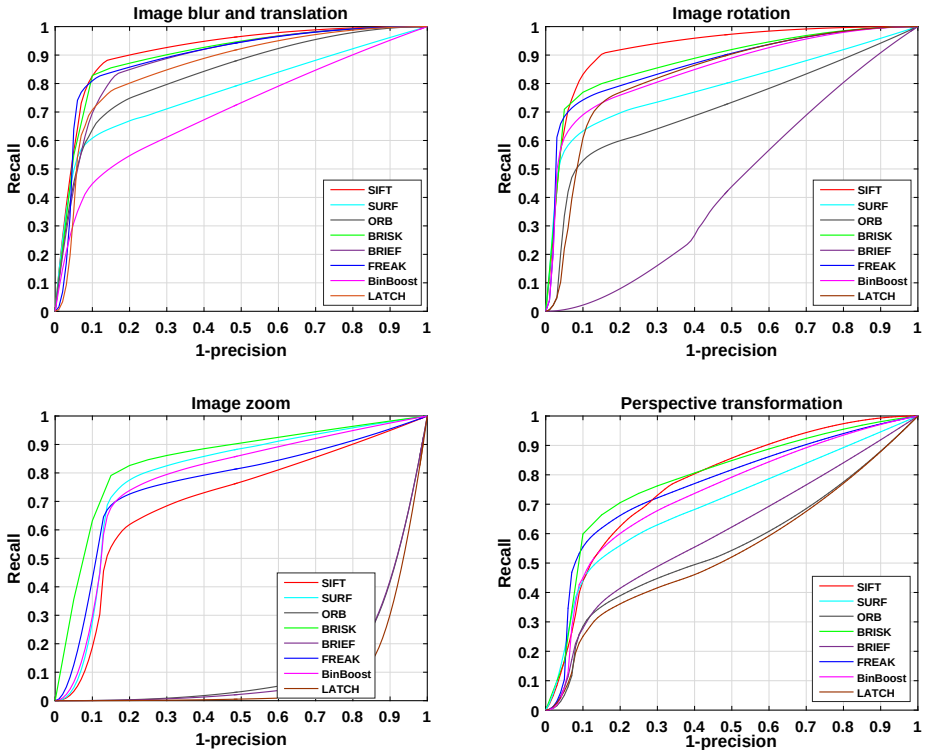

Figure 2.5: Comparison of various descriptors using recall vs 1-precision under different image degradations. The evaluation results are for the Fischer dataset [121].

descriptor is more efficient than the SIFT descriptor. However, binary string descriptors like ORB, BRIEF, BRISK and FREAK perform much faster than the other local descriptors. Thus, the binary string descriptors ORB, BRIEF, BRISK and FREAK are more appropriate for real-time applications.

2.6.3 Affine Invariant Evaluation

According to the above performance evaluation, most of the salient point methods are significantly influenced by affine transformations. As the framework of fully affine space could improve the accuracy of correspondence matching under huge viewpoint changes,we evaluate each salient point method in the framework of fully affine space and employ the proposed *K-order NNDR* matching strat-

egy to define the final correspondences. The evaluated salient point methods in the framework of fully affine space contain SIFT+SIFT (detector+descriptor), SURF+SURF, SURF+BRIEF, ORB+ORB, BRISK+BRISK, SURF+FREAK, SURF+BinBoost and SURF+LATCH. We also use randomized KD-trees to establish an index space and Euclidean distance for real valued descriptor matching. Binary descriptors are matched in a LSH index space with Hamming distance.

For the extracted local features of salient points in two compared images I and I', the obtained set of matches can be defined as:

$$M_{I-I'} = \{p_I^i \leftrightarrow p_{I'}^j\} \tag{2.6}$$

point $p_{I'}^j$ in image $I^{'}$ is the closest neighbor to point p_I^i in image I. We need to note the situation that the same point in the index space could be the nearest neighbor to different points in the query space (many-to-one matches), we then enforce a one-to-one constraint through a cross-check operation. The cross-check operation starts by building an index space for the local descriptors in the query image, and searching the k closest neighbors for each point in the reference image. Then we build the index space for the local descriptors in the reference image, and find k nearest neighbors for each point in the query image. Only if they

Table 2.4: The Oxford benchmark results [108]. Numerical results summarizing area under the recall vs. 1-precision curve for different transformations. Higher results are better.

Descriptor	Affine	Blur	Illumination	JPEG	Average
SIFT	**0.523**	0.832	0.892	**0.931**	**0.794**
SURF	0.404	0.49	0.774	0.723	0.598
ORB	0.141	0.596	0.844	0.711	0.573
BRISK	0.5	0.716	0.866	0.824	0.727
BRIEF	0.113	**0.841**	0.864	0.879	0.674
FREAK	0.484	0.735	0.843	0.863	0.731
BinBoost	0.4	0.412	0.83	0.641	0.571
LATCH	0.164	0.697	**0.894**	0.809	0.641

Table 2.5: The Fischer benchmark results [121]. Numerical results summarizing area under the recall vs. 1-precision curve for different transformations. Higher results are better.

Descriptor	Blur+Translation	Perspective	Rotation	Zoom	Average
SIFT	**0.915**	0.776	**0.925**	0.705	0.83
SURF	0.777	0.702	0.791	0.796	0.766
ORB	0.837	0.556	0.715	0.128	0.559
BRISK	0.902	**0.79**	0.887	**0.85**	**0.857**
BRIEF	0.882	0.606	0.443	0.117	0.41
FREAK	0.893	0.767	0.871	0.763	0.824
BinBoost	0.707	0.735	0.85	0.78	0.768
LATCH	0.859	0.54	0.832	0.1	0.583

satisfy formula (2.7), they can be considered a match.

$$M = \{M_{I-I'} = \{p_I^i \leftrightarrow p_{I'}^j\} \ \wedge M_{I'-I} = \{p_{I'}^j \leftrightarrow p_I^i\}\} \tag{2.7}$$

We use the proposed *K-order NNDR* matching strategy, replacing the original *NNDR* matching strategy, to define the matched correspondences:

$$C = \{p_I^i \leftrightarrow p_{I'}^j | \textit{K-order NNDR}(p_I^i, p_{I'}^j) < \textit{threshold}\} \tag{2.8}$$

where *K-order NNDR*$(p_I^i, p_{I'}^j)$ denotes that two similar descriptors satisfy the *K-order NNDR* threshold and $(p_I^i, p_{I'}^j) \in M$.

As the salient point extraction in the fully affine space could result in duplicate correspondences, we eliminate these duplicates according to the spatial distance (2

Table 2.6: Comparison of average description time cost on both two datasets

Method	Feature dimensions	Memory requirement (1000 points)	Oxford Dataset [108] Average time cost(s)/5400	Fischer Dataset [121] Average time cost(s)/6000
SURF+SIFT	128 float	0.488M	4.3	4.8
SURF+SURF	64 float	0.244M	0.24	0.26
SURF+BRIEF	256 bit	0.03M	0.013	0.015
SURF+ORB	256 bit	0.03M	0.015	0.018
SURF+BRISK	512 bit	0.06M	0.028	0.032
SURF+FREAK	512 bit	0.06M	0.02	0.025
SURF+BinBoost	256 bit	0.03M	3.03	3.27
SURF+LATCH	256 bit	0.03M	0.25	0.28

pixels) of point location in both image. To further determine whether a matched correspondence is correct or not, each correspondence obtained by the *K-order NNDR* is determined as correct only if its corresponding point is geometrically the closest point within the defined pixel coordinate error, and the final correct correspondences are evaluated by the ground-truth homography:

$$Correct\_matches = \{p_I^i \leftrightarrow p_{I'}^j | D(H(p_I^i), p_{I'}^j) < \varepsilon\} \tag{2.9}$$

where $D(H(p_I^i), p_{I'}^j)$ is the position error after the ground-truth homography H projection for the point in image I, and in all cases, the ε is set as 2 pixels.

Following common practice in evaluation protocols, we use the total number of correct matches between two compared images as criterion for the evaluation of correspondences matching. As ASIFT set the *NNDR* matching threshold to 0.73×0.73, we use the same threshold in our *K-order NNDR*. Moreover, in the framework of fully affine space, the parameter of tilt t controls the number of generated synthetic images in the affine space, and we need to note that larger value of the parameter t leads to higher computational complexity of the framework of fully affine space. For the evaluation, we set the parameter of t to 5, 6, and 7 corresponding to the numbers of the generated synthetic images 27, 41, and 61, respectively.

2.6.3.1 Parameter of K in *K-order NNDR*

In this part, we evaluate the impact of size K in the *K-order NNDR*. The images under viewpoint changes in the Oxford dataset [108] and the images for perspective changes in the Fischer dataset [121] are used. The impact of K in the *K-order NNDR* is shown in Figure 9. The test is based on the SIFT+SIFT, where the tilt in the scale space is set to 5. Figure 9 displays that the amount of correct correspondences shows a tendency to increase when K becomes larger, and for the SIFT detector with the SIFT descriptor, the *K-order NNDR* shows superior results to the original *NNDR*. Since the increase of magnitude of the correct correspondence is not significant when K varies from 4 to 6 and larger

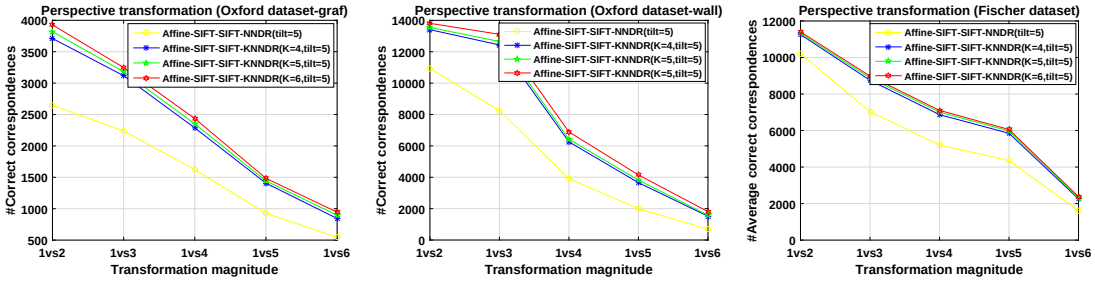

Figure 2.6: The demonstration of parameter K in the *K-order NNDR* (*KNNDR*) used in the fully affine space framework.

value of K reduces the efficiency of *K-order NNDR*, we set K equal to 4 in the following experiments.

2.6.3.2 Correspondence Matching Using the Framework of Fully Affine Space

For an objective comparison, we first evaluated the performance of each method without using the fully affine space framework. Figure 2.7 displays the amount of correct correspondences on the Oxford dataset, as well as the average numbers of correct correspondences on the Fischer dataset. It is clear that the SIFT+SIFT performs best on both datasets, and ORB+ORB, BRISK+BRISK are more sensitive to the affine changes (scale, rotation and perspective changes) than the other salient point methods. However, when the magnitude of perspective transformation becomes larger, all methods show poor performance.

As all salient point methods can only tolerate a small magnitude of viewpoint transformation, we apply the fully affine space framework and the proposed *K-order NNDR* scheme to evaluate their performance. Figure 2.8, Figure 2.9 and Figure 2.10 depict the evaluation results for real valued and binary string descriptors. It can be observed that a similar tendency is demonstrated on both datasets. When comparing the results of salient point methods using the fully affine space framework with the previous results, the performance has been significantly improved under large viewpoint transformations. We can note that Affine-

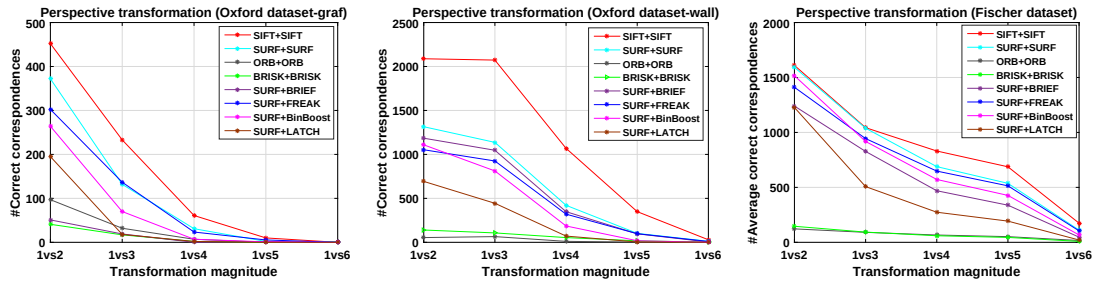

Figure 2.7: The demonstration of the amount of correct correspondences under perspective changes for each salient point method without using the fully affine space framework.

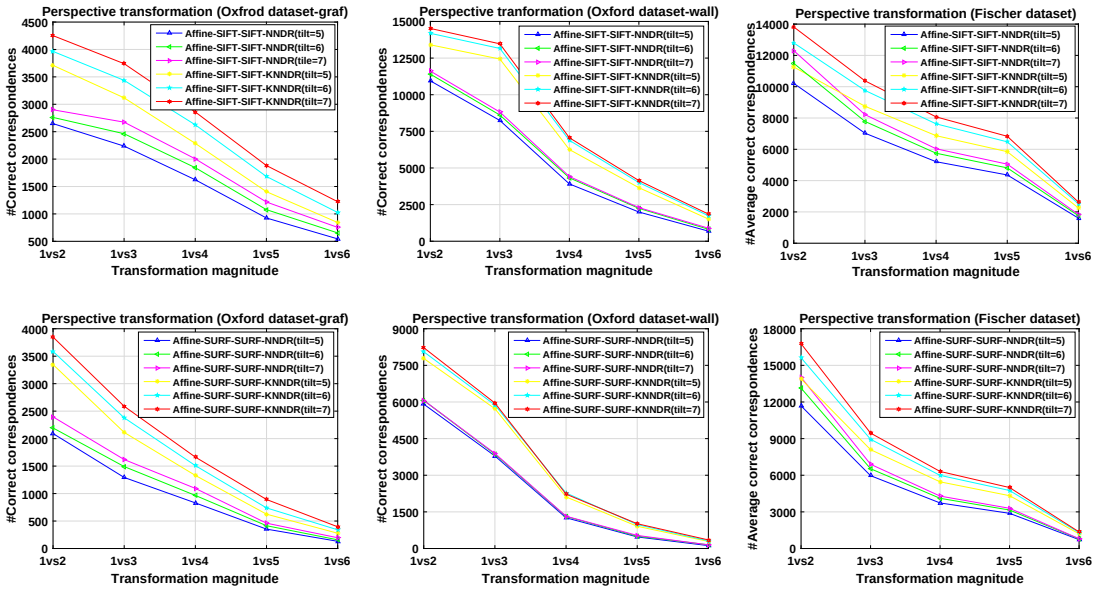

Figure 2.8: Evaluation results of salient point methods with real valued descriptor. The fully affine space framework is applied (the tilt varied from 5 to 7), and both *NNDR* and *K-order NNDR* (*KNNDR*) are compared.

SIFT+SIFT obtained the highest number of correct matches in all cases, and this is mainly due to the distinctiveness of the SIFT local descriptor. Moreover, the real valued descriptors are more distinctive than binary string descriptors.

In addition, for the comparison between *NNDR* and *K-order NNDR*, the evaluation results show the advantages of the *K-order NNDR* matching strategy. The

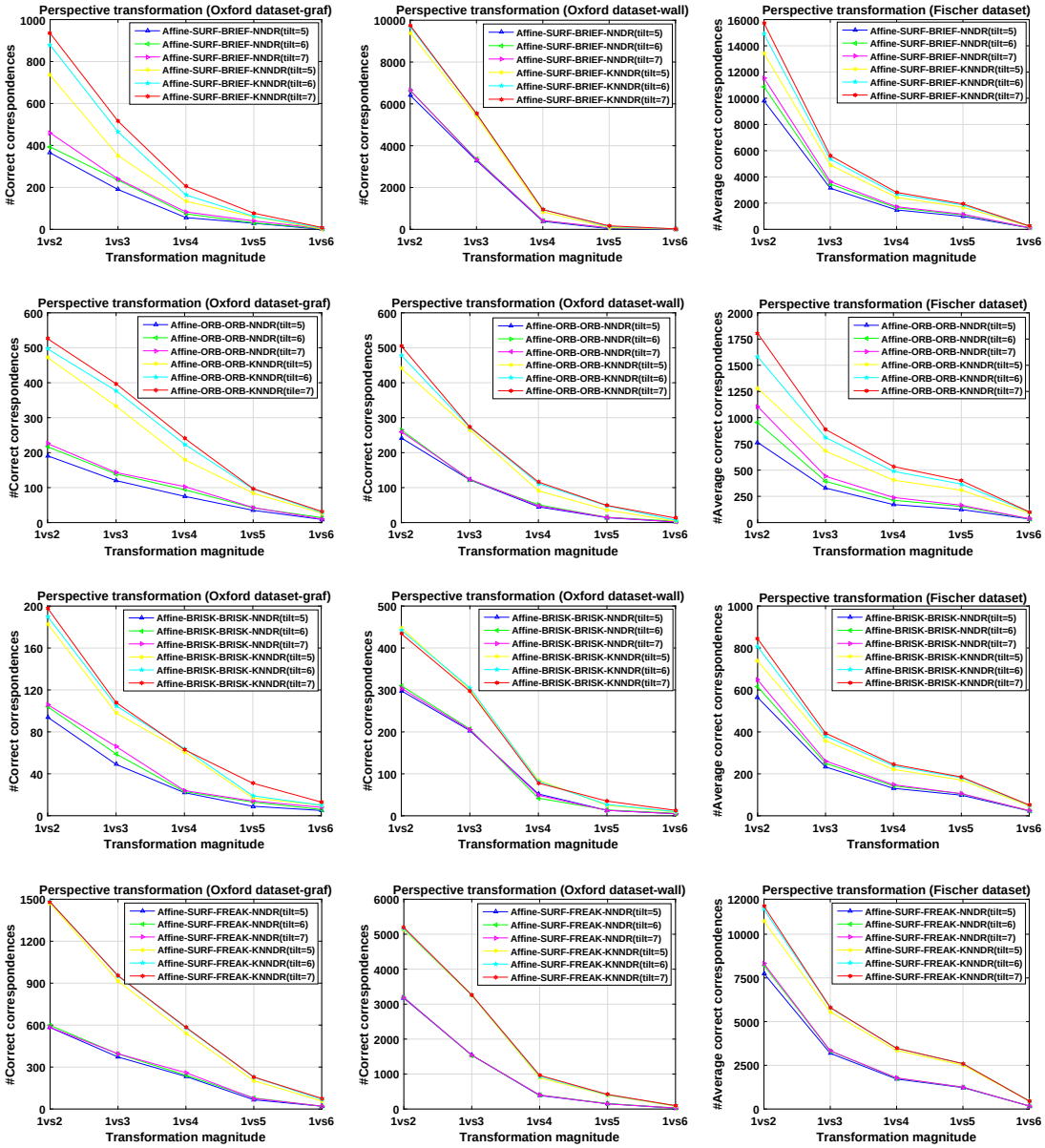

Figure 2.9: Evaluation results of salient point methods with hand-crafted binary string descriptor. The fully affine space framework is applied (the tilt varied from 5 to 7), and both *NNDR* and *K-order NNDR* (*KNNDR*) are compared.

K-order NNDR is effective for all the salient point methods. We can observe that *K-order NNDR* finds roughly double the number of correct correspondences compared to the original *NNDR*. Moreover, the results of *K-order NNDR* with tilt

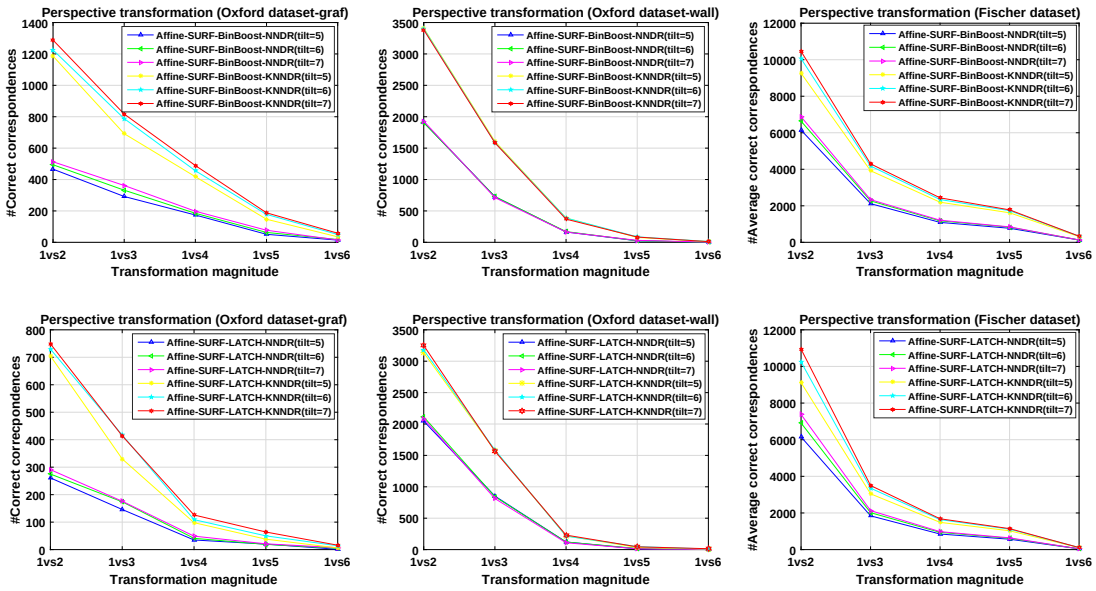

Figure 2.10: Evaluation results of salient point methods with supervised learning based binary string descriptors. The fully affine space framework is applied (the tilt is varied from 5 to 7), and both *NNDR* and *K-order NNDR* (*KNNDR*) are compared.

equal to 5 is even much better than *NNDR* with tilt equal to 7. This means that *K-order NNDR* can get high accuracy even at a low computational complexity of the fully affine space framework. Although the discrimination of binary string features is insufficient, binary string descriptors using the *K-order NNDR* can also offer competitive results compared to real valued descriptors using *NNDR*.

According to the above evaluation results on both datasets, we can also note that the original salient point methods failed to find the correct matches under huge viewpoint changes, but they all get expected performance levels by using the fully affine space framework and the proposed *K-order NNDR* matching strategy. Especially for the BRIEF, ORB, BinBoost and LATCH local descriptors which are easily influenced by scale, rotation and viewpoint changes, good performance was obtained for these changes by the framework of fully affine space and *K-order NNDR*.

Table 2.7: The comparison of computational cost and memory requirement in the framework of fully affine space.

Method	Tilt=5		Tilt=6		Tilt=7	
	Average numbers of points	Average memory requirement	Average numbers of points	Average memory requirement	Average numbers of points	Average memory requirement
Affine-SIFT+SIFT	55635	27.15M	65384	31.9M	74095	36.16M
Affine-SURF+SURF	79341	19.36M	99341	24.24M	119627	29.2M
Affine-SURF+RIFF	79341	21.74M	99341	27.22M	119627	32.77M
Affine-SURF+BRIEF	79341	2.38M	99341	2.98M	119627	3.58M
Affine-ORB+ORB	13314	0.4M	19263	0.58M	25805	0.77M
Affine-BRISK+BRISK	17565	1.05M	20583	1.24M	23066	1.38M
Affine-SURF+FREAK	79341	4.76M	99341	5.96M	119627	7.16M

2.6.3.3 Computational Cost and Memory Requirement

Computational cost and memory requirement are also important to the framework of fully affine space, because they reflect the computational complexity of the framework as well as the potential for the requirement of real-time systems. Considering that each salient point method extracts different amounts of local features in the fully affine space, we evaluated the average number of detected salient points and average memory requirement per image. The statistical results are summarized in Table 2.7.

It is worth noting that Affine-SIFT+SIFT and Affine-SURF+SURF consumed a huge amount of memory for the salient points detection and descriptor extraction in the fully affine space. For Affine-SIFT+SIFT and Affine-SURF+SURF, a large amount of salient points is extracted in the fully affine space framework and it increases the memory consumption correspondingly. We can also note that the memory requirement of binary string descriptors is less than that of real valued features. Moreover, as the performance show that the binary string features also achieved expected results under major viewpoint changes, integrating binary string features with *K-order NNDR* matching strategy in the framework of fully affine space is a good candidate for real-time systems.

2.7 Conclusions

In this chapter, we presented a comparison of detectors and descriptors on diverse image distortions and also evaluated their performance in the framework of fully affine space. According to the evaluation results, the FAST detector had the highest repeatability score compared to the score of other detectors, moreover it had the least detection time cost per point. Regarding the criterion of recall-precision, our experiments showed that the descriptors of SIFT, BRISK, and FREAK performed the best as affine invariant descriptors, and the time complexity showed that the binary descriptors provide very efficient feature description and matching.

In addition, for the special case of finding correspondences, we proposed the *K-order NNDR* matching strategy for the correspondences matching in the framework of fully affine space, and the experimental results show that the *K-order NNDR* is effective and obtained high accuracy correspondences under challenging image transformations. Furthermore, Affine-SIFT+SIFT showed the best performance on the correct correspondences in the framework fully affine space. When taking into account the computational complexity and memory requirement, binary string descriptors using the *K-order NNDR* matching strategy are a good trade-off between the accuracy and efficiency.