A Novel Filtering Approach for Robust and Fast Keypoint Matching in Mobile Environment

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

***Abstract*—The abstract goes here.**

***Index Terms*—IEEEtran, journal, LATEX, paper, template.**

I. Introduction

IMAGE matching is a fundamental problem in a variety of computer vision applications, including simultaneous localization and mapping [1], [2], object recognition [3], panorama stitching [4], [5], augmented reality [6], [7], and visual odometry [8], [9]. To enhance the image matching quality in various environments, many related techniques have been proposed, such as keypoint-based local matching, histogram-based global matching, template-based matching, . Among them, keypoint detection and matching has created great interest since it can provide relatively high matching quality against severe occlusion and do not require segmentation. Also, recent work has concentrated on making invariant to image transformation with low computing power [10], [11].

The overall flowchart of keypoint matching and recognition is shown in Fig. 1. These procedure can be divided into two main phases: offline(training) and online (testing) procedure. Offline learning is prerequisite to online matching process. In offline learning phase, a set of reference images to be recognized is analyzed and stored as as types of descriptors in a database. In online learning phase, a newly captured image is analyzed and compared with the reference images in the database to find a nearest reference image. In each phase, common procedures for matching are keypoint detection, description, and matching. To analyze training images, at first, keypoints are detected from the images. Then, from those keypoints, local textures are analyzed and described. In this procedure, to provide robustness against rotation, scale, perspective transform, descriptors are constructed. Then, to be used in online phase, efficient matching structures, as databases, are constructed, such as kd tree [12], hashing. In the online matching phase, the database is used to find the most similar corresponding keypoints pair with a given query image. To find the most similar keypoint paris, with given a query image, keypoints are detected, detected keypoints are described about local texture, and compared with the preconstructed database.

Conventional keypoint matching methods stores almost every keypoints which are detected by keypoint detection process. Keypoint detection processes are designed to extract keypoints repeatable and robust against arbitrary image transformation. Then, detected keypoints are independent to the follow matching procedures, and do not reflect quality of descriptors. Therefore, as seen Fig. some keypoints are not distinguishable, and they tend to cause inter-keypoint confusion and miss matching. `bad keypoint` 이미지 추가 Also, those detected keypoints are stored in database and are compared with keypoints in query images in every frame while matching. Then, it decreases matching speed. To overcome these problem, in offline learning procedure, detected keypoints are evaluated with respect to proposed matching quality criteria and filtered by the goodness score. With this filtering method, only a small subset of keypoints is stored in the database. Accordingly, it provides more improved matching performance with faster matching speed.

Especially, in recent years, mobile computing devices have been widely deployed to customers, the interest on mobile image matching is increasing. With the help of enhanced computing power, small devices now have enough capability to process complex visual computations. However, mobile computing devices has insufficient computing power and limited memory, effective processing algorithms are necessary. However, conventional keypoint matching approaches stored redundant keypoints into database, and these redundant keypoints may compared in every frame. So, matching speed will be decreased and this causes problem in the mobile computing devices. So, the proposed method does not store redundant keypoints in the database, it reduces the number of comparison

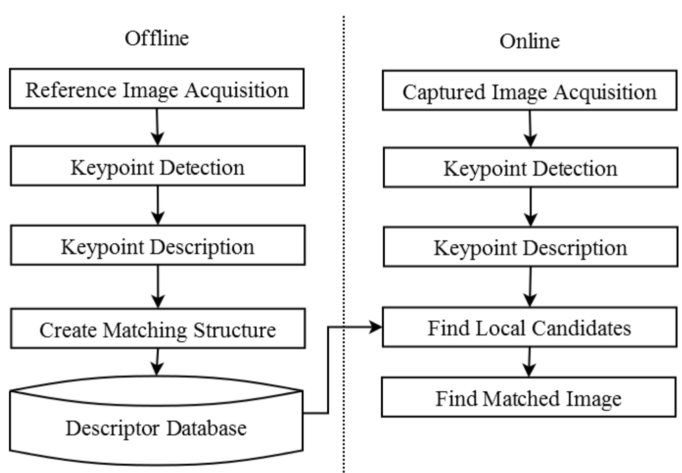

Fig. 1. Overall process of conventional keypoint-based matching

M. Shell is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).
J. Doe and J. Doe are with Anonymous University.
Manuscript received April 19, 2005; revised December 27, 2012.

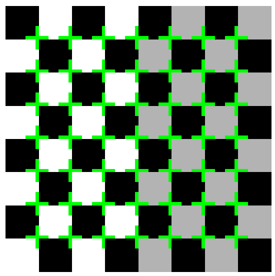

Fig. 2. Example of high repeatable but poor distinguishable points. Conventional keypoint matching systems do not consider the discriminability of keypoints, so these keypoints usually stored and negatively affected matching.

while matching process and increase matching speed even in the mobile computing environment.

This paper is structured as follows: In Section 2, we discuss literature on interesting point detection and matching systems, as well as conventional keypoint filtering algorithms. Section 3 defines the proposed keypoint score function, and we validated the function with matching ground truth data. Then, we compared image patches between match-friendly keypoints and other keypoints set. In Section 4, we executed experiments to prove the proposed keypoint filtering method in various dataset and algorithms and compared over several evaluation metrics. Finally, Section 5 presents the conclusion.

II. Related Works

In general, the framework of keypoint matching systems consists of three steps - ~~i.e. the detection describe match (DDM) framework [13].~~

A. Keypoint Detectors

The first step of keypoint matching is keypoint detection, ~~keypoint is used as corner, interest point, or feature. The aim of this study is to extract an interest point which is stable across image transformation.~~ Generally, researches use corner(Harris [14], SUSAN [15], etc) or center of silent region(SIFT [16], SURF [17], etc) as the interesting point since they are stable and easy to locate and describe [13].

1) Corner Detectors: Corners are ~~among~~ the first low-level features used for image matching. Considering corners as intersection of two edges, these features have no spatial extension and, therefore, there is no ambiguity in their location. Moravec [18] computes ~~the~~ sum-of-squared-differences between a patch around a candidate corner and patches shifted a small distance in a number of directions. Based on Moravec's, Harris and Stephens [14] developed the Harris Corner Detector which is probably one of the most popular corner extraction methods. It is based on the first order Taylor expansion of the second derivative of the local SSD with respect to the shift. Mikolajczyk and Schmid [19] proposed an approach to make the Harris detector scale invariant. Based on the assumption of affine image deformation, Shi and Tomasi [20] obtained the same equation by analyzing the optical flow equation proposed by Lucas and Kanade [21]. Also, other intensity-based corner detectors include the algorithms of Beaudet [22], which uses the determinant of the Hessian matrix, and Kitchen and Rosenfeld [23], which measures the change of direction in the local gradient field.

To avoid costly window or filter operations, examining a small patch of an image to see if it "looks" like a corner is proposed. Since second derivatives are not computed, a noise reduction step (such as Gaussian smoothing) is not required. Consequently, these corner detectors are computationally efficient since only a small number of pixels are examined for each corner detected. Smith and Brady proposed so called "Smallest Uni-value Segment Assimilating Nucleus (SUSAN)" [15] for corner detection. The brightness of the center pixel, the nucleus, is compared to its circular pixel neighborhood, and the area of the uni-value segment assimilating nucleus(USAN) is computed. Corner and edges can be detected by evaluating this area, or it can also be used for noise reduction. Trajkovic and Hedley [24] used a similar idea: the pixel value at the center of a discretized circle is compared to the values on the circle. Based on this idea, Rosten and Drummond proposed "Features from Accelerated Segment Test (FAST)" [25] which combines the machine learning approach to speedup the comparison. This method has seen significant performance increase for real-time Computer Vision applications. Also, this method has proven in several applications to be reliable due to high repeatability (see [26]). Some applications which use FAST are, e.g., Klein's PTAM [6] and Taylor's robust feature matching ~~in 2.3 μs~~ [27]. From this idea, Mair et al.~~,~~ proposed "Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST)" corner detector [28]. This method generalized FAST corner detector to improve performance even in the generalized environment.

2) Silent Region Detectors: Instead of trying to detect corners, one may use local extrema of the responses of certain filters as interest points. In particular, many approaches aim at approximating the Laplacian of a Gaussian(LoG), which, given an appropriate normalization, was shown to be scale invariant if applied at multiple image scales [29]. Lowe [16] obtains scale invariance by convolving the image with a Difference of Gaussians (DoG) kernel at multiple scales, retaining locations which are optima in scale as well as space. DoG is used because it is good approximation for the LoG and much faster to compute. An approximation to DoG has been proposed which, provided that scales are $\sqrt{2}$ apart, speeds up computation by a factor of about two, compared to the striaghtforward implementation of Gaussian convolution [30]. Scale-space techniques have also been combined with the Harris approach in [19] which computes Harris corners at multiple scales and retains only those which are also optima of the LoG response across scales. Also, Hessian detector [17] is proposed which is based on efficient-to-compute approximations to the Hessian matrix at different scales.

In recent years, scale invariance has been extended to consider features which are invariant to affine changes [31]–[33]. Affine-invariant detectors provide higher repeatability for large affine distortions [16], [31], but are typically expensive to compute [34], [35].

B. Keypoint Descriptors

The next process of keypoint matching is constructing a keypoint descriptor for the local patch with regard to the detected keypoint. The aim of this process is to capture the most important and distinctive information content enclosed in the detected keypoints, such that the same structure can be recognized if encountered. To accomplish the distinctiveness in real time, the inherent difficulty lies in balancing two competing goals: high-quality description and low computation requirements. Considered in these aspects, the description algorithms are classified in Real-value based [16], [17], [36] and Binary-value based [37]–[40].

1) Real-Value based Descriptors: The algorithms in this category rely on feature vector of an image region, where each dimension is a floating-point type (or a discretization of a float excluding binary). These algorithms use ~~image gradients, spatial frequencies, etc.~~ to describe the local image patch and to ~~test for similarity by using~~ the L^2 norm, Mahalanobis distance, etc. These descriptors have proven to be effective, and tackle issues such as scale, rotation, viewpoint, or illumination variation. A variety of features derived from the local image intensities have been proposed to derive robust feature descriptors. Early ideas include derivatives for rotationally invariant features [41], derivatives of Gaussians of different order [42], filter banks derived from complex functions [33], phase information [43], and others. Many of these have been evaluated by [11].

Amongst the best quality features currently in the literature is the SIFT [16]. A 128-dimensional vector is obtained from a grid of histograms of oriented gradient. ItsThe high descriptive power and robustness to illumination and viewpoint changes has rated it as the reference keypoint descriptor for the past decade. However, the high dimensionality of this descriptor makes SIFT prohibitively slow. PCA-SIFT [36] reduced the description vector from 128 to 36 dimensions using principal component analysis. The matching time is reduced, ~~but~~ the time to build the descriptor is increased ~~leading to~~ a small gain in speed and a loss of distinctiveness. The GLOH descriptor [11] belongs to the family of SIFT-like methods and has been shown to be more distinctive but also more expensive to compute than SIFT. The robustness to change of viewpoint is improved in [44] by simulating multiple deformations to the descriptive patch.

One of the widely used keypoints at the moment is clearly SURF [17]. It has similar matching performances as SIFT but is much faster. It also relies on local gradient histograms. The Haar-wavelet responses are efficiently computed with integral images leading to 64 or 128-dimensional vectors. However, the dimensionality of the feature vector is still too high for large-scale applications such as image retrieval or 3D reconstruction.

2) Binary-Value based Descriptors: Another category is binary descriptors. These descriptors have a compact binary representation and limited computational requirements, computing the descriptor directly from pixel-level comparisons. This makes them an attractive solution to many modern applications, especially for mobile platforms where both compute and memory resources are limited. Each bit in the descriptor is the result of one comparison, and the descriptor is built from a set of pairwise intensity comparisons. Because the descriptor is constructed by simple comparison operation, these algorithms provide simple and inexpensive complexity. Also, in this category, Hamming distance (bitwise XOR followed by a bit count) is used to compute a similarity, and it replaces the usual Euclidean distance. So, matching complexity also will be decreased.

The first approach of this category is "Binary Robust Independent Elementary Features (BRIEF)" [37]. It uses a sampling pattern consisting of 128, 256, or 512 comparisons (equating to 128, 256, or 512 bits), with sample points selected randomly from an isotropic Gaussian distribution centered at the feature location. The obtained descriptor is not invariant to scale and rotation changes unless coupled with detector providing it. Calonder et al. also highlighted in their work that usually orientation detection reduces the recognition rate and should therefore be avoided when it is not required by the target application. Rublee et al. proposed the "Oriented Fast and Rotated BRIEF (ORB)" descriptor [40] to overcome the lack of rotation invariance of BRIEF. The descriptor is not only invariant to rotation, but also robust to noise. The sampling pattern employed in ORB uses 256 pairwise intensity comparisons, but in contrast to BRIEF, is constructed via machine learning, maximizing the descriptor's variance and minimizing the correlation under various orientation changes. Leutenegger et al. proposed a binary descriptor invariant to scale and rotation so called "Binary Robust Invariant Scalable Keypoints (BRISK)" [38]. To build the descriptor bit-stream, a limited number of points in a symmetric pattern is used. Each point contributes to many pairs. The pairs are divided in short-distance and long-distance subsets. The long-distance subset is used to estimate the direction of the keypoint while the short-distance subset is used to build binary descriptor after rotating the sampling pattern. Overall, BRISK requires ~~significantly~~ more computation and ~~slightly~~ more storage space than either BRIEF or ORB. Similarly, Alahi et al. proposed "Fast Retina Keypoint (FREAK)" descriptor [39]. It is inspired by the human visual system ~~and more precisely the retina~~. A cascade of binary strings is computed by efficiently comparing image intensities over a retinal sampling pattern. FREAK requires lower storage space and shows more robust than BRISK.

C. Keypoint Matching

The next process in keypoint matching is searching for the most similar matches to high-dimensional vectors, also referred to as nearest neighbor matching. In general, the number of keypoints from a single image may be hundred to thousand, so in this process, huge number of comparison is executed, and this process becomes a bottleneck to entire process. To improve this problem, some literatures to construct an efficient matching data structure ~~are researched~~. There are two aspect to effect matching quality and speed - feature dimension and matching method. The former is concerned with keypoint description ~~algorithm~~. As described in the former section, binary descriptors offers fast matching speed because they can be computed by executing the XOR operation followed by a few bitwise instructions that can be performed quickly, especially on modern central processing units(CPUs). Usually millions of binary codes are compared only in less than a second [45]. Even though the distance between binary codes can be computed efficiently, using linear nearest neighbor search for exact matching is practical only for small datasets. For large-scale datasets, exact nearest neighbor search will lose its time performance. To solve this problem, various approximated nearest neighbor(ANN) search algorithms were proposed. These ANN algorithms are classified in two categories: partitioning trees and hashing techniques [46].

1) Partitioning Trees: The kd-tree [47], [48] is one of the best known nearest neighbor algorithms. Arya et al. [49] proposed an error bound approximate search method based on kd-tree. The algorithm used a priority queue to speed up the search. Meanwhile, Beis and Lowe [50] proposed a time bound approximate search. In practice the time-constrained approximation criterion has been found to give better results than the error-constrained approximate search. Various extensions of k-d tree algorithm were proposed [51]–[54]. In [55], several approximated nearest neighbor algorithms were compared, and the multiple randomized k-d tree was the most effective.

Another class of partitioning trees decompose the space using various clustering algorithms instead of using hyper planes as in the case of the k-d tree and its variants. Example of suck decompositions include the hierarchical k-means tree [56], the GNAT [57], the anchors hierarchy [58], and the spill-tree [59]. The vocabulary tree [3] is searched by accessing a single leaf of a hierarchical k-means tree. Schindler et al. [60] proposed a new way of searching the hierarchical k-means tree.

Also, another class of partitioning trees combined keypoint description method to construct matching structure [61]–[63]. Lepetit and Fua [61] formulated keypoint matching as a classification problem using Randomized Trees as classifiers. Özuysal et al. [62] simplified this approach structurally by adopting a naïve Bayes approach, thus simplifying the trees to ”ferns.” Taylor et al. [27] presented another training-based keypoint recognition approach, which, during the training step, builds coarsely quantized histogram based representations. As, these approaches utilized machine learning method, these approaches trained with huge number of training images which are synthesized by various transformation. So, these approaches take a long time to train a offline database, but these are able to improve online matching quality.

2) Hashing: In contrast to tree approaches, hashing is usually used for binary features. [64] proposed the notion of semantic hashing when they learn a deep graphical model that maps documents to small binary codes. When the mapping is performed such that close features are mapped to close codes (in Hamming space), the nearest neighbor matching can be efficiently performed by searching for codes that differ by a few bits from the query code. A similar approach is used by Torralba et al. [65] who learn compact binary codes from images with the goal of performing real-time image recognition on a large dataset of images using limited memory. Weiss et al. [66] formalize the requirements for good codes and introduce a new technique for efficiently computing binary codes.

Performing ANN search by examining all the points in a Hamming radius works efficiently when the distance between the matching codes is small. When this distance gets larger the number of points in the Hamming radius gets exponentially larger, making the method unpractical. This is the case for binary value features. where the minimum distance between matching features can be larger than 20 bits. In cases, the best known hashing based nearest neighbor technique is locality sensitive hashing(LSH) [67], [68], which uses a large number of hash functions with the property that the hashes of elements that are close to each other are also likely to be close. Variants of LSH such as multi-probe LSH [69] improves the ~~high~~ storage costs by reducing the number of hash tables, and LSH Forest [70] adapts better to the data without requiring hand tuning of parameters.

The different hashing algorithms provide theoretical guarantees on the search performance and have been successfully used in a number of projects, ~~however, [46] shows~~ that in practice they are usually outperformed by algorithms using space partitioning structures such as the randomized k-d trees ~~and similar approaches.~~

These approaches similarly, to provide efficient searching the approximated nearest neighbor, in the offline training phase, they executed large amount additional process to construct efficient searching structure. Because it is better to take more process in offline phase and provide more efficient searching result, so these techniques are widely executed. This idea is similar with our approach, our proposed algorithm performs large amount of offline process to filter out undistinguishable keypoints, and provides more efficient and robust matching quality.

D. Keypoint Filtering

The previous processes described how to extract repeatable keypoint, how to describe a local texture, and how to construct an efficient searching structures. However, they do not consider which keypoints to be stored to provide precise matching quality. To cover this issue, several literatures are proposed.

1) Filtering in keypoints detection: The main approach to decide which keypoints to be stored is performed in keypoints detection. The keypoints detection algorithms calculates corner response function(C.R.F.) which represents how much the points is repeatably extracted(*Repeatability*). There are two filtering scheme using the C.R.F: thresholding and non-maximum suppression. In the early stage of keypoint detection, thresholding scheme was widely used. Harris and Stephan [14] defined the Harris score from the second-order moment image gradient matrix. Similarly, Shi and Tomasi [20], by mathematical analysis, led that it is better to use the smallest eigenvalue of image gradient matrix as the corner strength function. A number of suggestions [71], [72] have been made of how to compute the corner strength from the matrix and filtered by thresholding. Also, while Harris score is scale variant, scale invariant keypoint detectors are also proposed. In [73], keypoints are filtered by setting a threshold for local peaks of a scale space in SIFT because a low peak coming from low contrast of image intensity is unstably computed. In the recent stage, non-maximum suppression(NMS) is used more popular, because this scheme can considers spatial relationship of keypoints in addition to keypoint response. Since keypoints with high response may appear in continuous pixels, pixels with only local maximum of the responses are selected and the others are suppressed. Some researches such as [16], [19], [40] used NMS based on the Harris score. [17] used the determinant of the Hessian matrix. So, each of keypoints detection algorithms defined their own corner response functions and based on those, they filters corner by non-maximum suppression. These non-maximum suppression has been extended to block based method [74], an adaptive method [75] and an efficient method with Suppression via Disk Covering [76].

2) Filtering Considering Distinctiveness: Even though keypoints are detected in different viewpoints with high repeatability, they are not considered distinctiveness. For example, as shown Fig. 2, corners in a checkerboard can be stably detected, but it is hard to correctly match or distinguish them in different viewpoint. Therefore, such points are not appropriate for robust keypoint matching and those should be filtered out. Therefore, stored keypoints must consider the distinctiveness to provide precise matching performance. However, it seems that ~~slight~~ literatures realized this problem. In [77], [78], a feature vector is computed from local texture and then compared with other feature vectors. By removing pixels that have similar feature vectors, only distinctive keypoints can be selected. In contrast those researches, we propose keypoint filtering criteria considering with not only repeatability, but also distinctiveness and each keypoint's variance.

E. Keypoint Matching Systems

Combined with these keypoint matching algorithms, there are lots of keypoints matching applications proposed. In table I, these systems are listed along with ~~the algorithms that are used in~~. This compilation is not meant to be exhaustive, and the short bullet points do not do justice to specific features and contributions of the listed systems. Rather, it is meant to give an overview of the applications of visual tracking and the algorithms that have been employed for different components.

III. Proposed Method

본 논문에서는 matching quality 를 기준으로 detected keypoint 를 evaluate하여 filtering하는 방법을 제안한다. 이를 통하여 matching quality가 높은 점들만을 학습함으로써 online matching 과정에서 높은 matching preciseness와 빠른 속도를 얻을 수 있다. 본 장에서는 matching quality 에 따라서 keypoint를 evaluation 하기 위한 criteria 를 정의하고, ground truth data 를 기준으로 이러한 criteria 가 동작됨을 증명한다. 또한, 이러한 기준에 의하여 분류된 keypoint 들의 image patch 를 비교하여 matching에 좋은 특징점과 그렇지 않은 특징점의 xxx 측면에서의 차이점을 비교하였다.

A. Statement of the Problem

In general, keypoint matching methods 일반적으로 키포인트 기반의 매칭 방법은 미리 학습된 키포인트 데이터베이스 K^R와 입력된 영상을 분석하여 생성된 키포인트 집합 K^I를 비교하여, 가장 유사한 키포인트 pair 집합 $C = \{(k_i^R, k_j^I)|\underset{k_i\in K^R}{\operatorname{argmin}}\,\underset{k_j\in K^I}{\operatorname{argmin}}\,|k_i^R - k_j^I|\}$ 을 계산하는 과정이다. 기존의 키포인트 매칭 방법은 검출된 키포인트 집합 K^R 을 그대로 사용하였으나, 본 논문에서는 키포인트 평가 함수($s(k)$)를 제안하여 이러한 평가 함수에 의하여 필터링된 집합 $K' = \{k|s(k)is \quad high\} \in K^R$ 을 계산하고, 이러한 필터링 된 부분집합 K'는 필터링 되지 않은 K^R에 비하여 더 높은 인식성능을 보여줌을 증명하고자 한다. 조금만 더 늘여쓰자

B. Keypoint Score Function

General keypoint matching generates the feature database, the subjects of comparison, by way of the offline training of the reference images before online matching procedure. In particular, as shown in the foregoing study of matching data structure, a method that composes a matching structure by applying a various computations to offline process to increase the online recognition speed and recognition rate is proposed. Such established kaypoints matching methods used simply all of the keypoints detected in feature detection module for training. However, since the feature detection algorithm is performed independently from the description algorithm, the descriptor is not able to ensue the matching performance for the detected features. 이유를 들라

Thus, in this paper, the keypoints of the subjects of training in the offline training process were assessed and only the keypoints providing rigid real-time recognition performance were selected. Accordingly, both recognition rate and speed can be enhanced by only saving discriminant (good) features in the database based on proposed method.

1) Definition of Good Keypoints: The proposed filtering method selects only good keypoints by analyzing the characteristics of detected keypoints and measuring the degree of the effectiveness for image matching.

There are several factors which good keypoints should follows:

Repeatability: First, good keypoints need to be stably detected even in various transformed environment. In actual matching environments, a wide range of transformation may occurred, such as the rotation, size, noise and lighting of the targeted images. The good keypoints has to be stably extracted even those transformation.

The detection of stable keypoints can be measured by *Repeatability* condition. *Repeatability* is calculated by the ratio between the total number of synthesized images and the number of cases where the transformed keypoints are existent in the synthesized images.

TABLE I
KEYPOINTBASED IMAGE MATCHING SYSTEMS

Reference	Detector	Descriptor	Matching	Filtering
Bleser and Stricker (2008) [79]	FAST	patch, warped		
Carrera et al. (2007) [10]	Harris	SURF	Linear nearest neighbor	NMS (Determinant of Hessian)
Chekhlov et al. (2007) [80]	Shi-Tomasi	SIFT-like		
Davison et al. (2007) [2]	Shi-Tomasi	patch, warped		
Eade and Drummond (2006) [81]	FAST	patch, warped		
Klein and Murray (2007) [6]	FAST	patch, warped		
Lee and Höllerer (2008) [82]	DoG	Optical flow & SIFT	k-d tree	NMS (DoG)
Lepetit and Fua (2006) [61]		Randomized Trees	Randomized Trees	
Muja and Lowe (2012) [12]	DoG	SIFT	FLANN	
Nistér et al. (2004) [9]	Harris	patch		
Özuysal et al. (2007) [62]		Ferns	Ferns	
Park et al. (2008) [83]	not specified	Ferns	Ferns	not specified
Se et al. (2002) [84]	DoG	scale, orientation		
Skrypnyk and Lowe (2004) [85]	DoG	SIFT	k-d tree	
Taylor et al. (2009) [27]	FAST	trained histograms		
Wagner et al. (2009) [7]	FAST	patch & reduced SIFT	a forest of spill trees	NMS (DoG)
Wagner et al. (2010) [5]	FAST	patch, warped		
Wiliams et al. (2007) [86]	FAST	Randomized lists		

$$p_{rep}(p_i) = \frac{n_i^{overlap}}{N} \tag{1}$$

where $p_{rep}(p_i)$ represents repeatability score of given point p_i, $n_i^{overlap}$ is calculated by the frequency of the existence of transformed keypoint(p_i) in the set of keypoints($T(p_i) \in K_t'$) of synthesized images $T_t(I)$; N is the total number of synthesized images ; and all keypoints have single value.

Similarity: Good keypoints need to be well-matched with identical keypoints even though targeted images change in various ways(*Similarity* condition). With regard to a certain keypoint(p_i) of reference images, genuine distribution' and imposter distribution' for the corresponding keypoint can be measured by calculating the matching between the descriptors of all the sets of keypoints(p_i) in images($T_t(I)$) transformed in various ways during the training process. At this time, to reduce the failure in matching the corresponding keypoints and the descriptors in the transformed images, the genuine distribution needs to have small value, being far enough away from match distance threshold. To this effect, it was measured using the mean of genuine distribution. As shown in Equation (2), the keypoints with the decreasing the genuine distribution are better, so the evaluation function was calculated by normalizing the mean of the genuine distribution and subtracting its value from 1.

$$p_{sim}(p_i) = 1 - \frac{\mu_{gen,i} - \min_i \mu_{gen,i}}{\max_i \mu_{gen,i} - \min_i \mu_{gen,i}} \tag{2}$$

where $p_{sim}(p_i)$ represents similarity score of given point p_i, $\mu_{gen,i}$ represents mean of genuine distribution of give point p_i.

Separability: Trained keypoints and other keypoints shall not be matched(*Separability*), which is associated with the imposter distribution of each keypoint. Of the keypoints extracted from the images converted in various images, the distribution of the matching with other keypoints rather than the converted keypoints themselves are referred to as imposter distribution. Thus for a specific keypoint to show the low success rate of matching with other keypoints rather than themselves, in is necessary that the genuine distribution and imposter distribution are well classified. To this effect, in this paper, *Fisher's Discriminant Ratio [87]* was used. It measures the distance between two classes by the mean and distribution of sample in 1-dimensional, two class problems. Since the second Similarity condition ensures the genuine distribution is small enough, the nonexistence of the matching with the keypoints in the imposter distribution is ensured if the importer distribution is far enough away compared with the genuine distribution. Separability value also requires the normalization process as shown in equation (4).

$$FDR(p_i) = \frac{(\mu_{gen,i} - \mu_{imp,i})^2}{\sigma_{gen,i}^2 + \sigma_{imp,i}^2} \tag{3}$$

$$p_{sep}(p_i) = \frac{FDR(p_i) - \min_i FDR(p_i)}{\max_i s_i} \tag{4}$$

where $p_{sep}(p_i)$ represents similarity score of given point p_i, $\mu_{gen,i}, \mu_{imp,i}$ represent mean of genuine and impostor distribution of give point p_i, respectively. $\sigma_{gen,i}, \sigma_{imp,i}$ represent standard deviation of genuine and impostor distribution, respectively.

The score functions of each keypoint can be defined using 3 criteria calculated as above. The 3 conditions are dependent, so can be defined as shown in Equation (5).

$$gf(p_i) = p_{rep}(p_i)p_{sim}(p_i)p_{sep}(p_i) \tag{5}$$

C. Proof of Criteria

1) Validation Design: To validate the proposed keypoint evaluation criteria, we examined a relationship between criteria and correct matching count of each keypoints. At first, to provide robust image matching, we synthesized image dataset by various image transformation. Then, based on this dataset, we counted correct matching count for each keypoint, and this correct matching count is a basis of matching quality. 이러한 correct matching count가 높은 특징점은 fixed image dataset 에서 더 높은 matching quality를 보여준다고 볼 수 있기 때문에 본 논문에서 제안하는 Matching에 더 적합한 keypoint로

볼 수 있다. 반대로 correct matching count가 낮은 특징점은 특징점이 반복적으로 검출되지 않거나, 모호성이 높아 inter-keypoint miss-match가 많이 발생하는 특징점으로 matching 에 적합하지 못한 keypoint로 볼 수 있다. 따라서, 이러한 correct match count 와 제안하는 keypoint evaluation score function (see, Eq. 5) 간의 상관관계를 관찰함으로써 제안하는 score function 의 적절성을 검증할 수 있다.

2) Dataset: 검증에 사용된 이미지는 서울 관광 가이드북 [88]의 *Seoul Tour Map* 16장을 사용하였다. 우리는 이러한 이미지를 대상으로 rotate(0.5 − 2.0-folds, at the interval of 0.1-fold), scaling(0° − 360°, at the interval of 10 intervals), and blurring (Gaussian blur, $r \in \{0, 3, 5, 7\}$ pixels) 의 transform 을 적용하여 총 36,864 장의 dataset을 생성하였다. 이 중 랜덤으로 training Set 16,114 장, Test set 16,142 장을 선택하여 실험을 진행하였다.

3) Images Patches: 그림 3와 같이, correct matching count 를 기준으로 상위 10개의 keypoint 와 하위 10개의 keypoint 들의 특징을 비교하였다. 상위 10개에 대한 패치는 비교적 단순한 사각형 형태에서 많이 검출되었다. Genuine과 Impostor Histogram의 값을 정규화하여 표현된 Normal Distribution의 분포를 보면 Genuine과 Impostor 분포가 확연하게 구분되는 것을 확인할 수 있다. 반면, 하위 10개에 대한 패치는 글자 또는 단순한 패턴이 반복되는 형태에서 많이 검출되었다. Genuine과 Impostor Histogram의 값을 정규화하여 표현된 Normal Distribution의 분포를 보면 Genuine과 Impostor 분포가 많은 부분 겹쳐있어 구분이 어려운 것을 확인할 수 있다. 인식에 좋은 특징점은 큰 숫자 패치와 같이 단순한 색상으로 패턴이 큰 숫자 표지와 같은 특징점이 인식에 좋은 성능을 보여주었으며, 반대로 작은 설명 글씨와 같은 특징점들은 인식 성능이 좋지 못하였으며, 이러한 점들을 제거하고 학습을 수행하는 것이 좋다.

(a) the best 100-images

(b) the worst 100-images

Fig. 3. The Best/Worst 100-Images with Regard to Correct Matching Count

IV. EXPERIMENTS

The improvement of recognition performance after the training using the keypoints filtering algorithm proposed earlier was measured. As for the experimental images, 16 images of Seoul Guide Map Pamphlet was selected. These images were deformed by way of rotating(0.5 $\sim$ 2.0-folds, at the interval of 0.1-fold) scaling (0° $\sim$ 360°, at the interval of 10 intervals) and blurring (Gaussian blur, $r = 0, 3, 5, 7$ pixels). As a result, 32,256 images were obtained. Of them, 16,114 training images and 16,142 test images were selected at random. First, the keypoints of training images were detected and then the score function($gf(p_i)$) was calculated using the detected keypoints.

The keypoints database is composed of the set of all keypoints($K_{all}, n(K_{all}) = 3000$) that did not consider the score function and the set of the keypoints($K_{50}, K_{100}, K_{300}, K_{500}$) composed of top 50, 100, 300 and 500 keypoints filtered in the score function.

First the improvement of recognition speed was measured. As for the proposed method, since the keypoints were reduced to be saved in the training phase, the number of the keypoints, the subjects of comparison, decreases, which in turn increases the computing speed.

As shown in Figure **??**, the computing speed improves in proportion to the number of the set of keypoints. In particular, the time spent for training deceases to $1/n$ compared to the training of whole keypoints when training was performed with 100 keypoints. In a lightweight implementation environment like smartphone, the reduction of computation provides rapid interaction. The proposed method is expeted to increases the speed and to improve the overall recognition performance. The test image recognition performance was measured using keypoints database. As for the match method for measuring recognition performance, we used the match method [89] which prevent false matches.

The results of the measurement of recognition rate using the above method were demonstrated in Figure 5. In comparison with the keypoints database using the whole of keypoint sets(K_{all}), K_{500} and K_{300} showed slight degradation of recog-

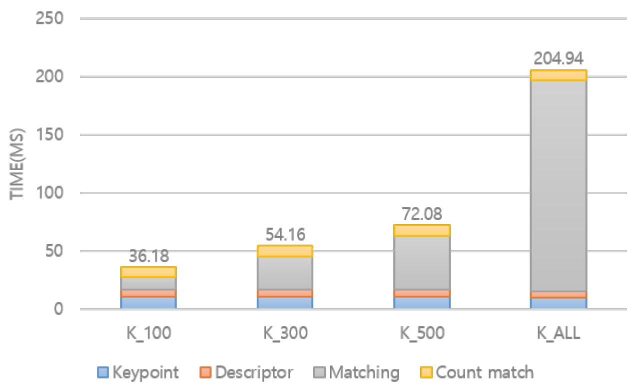

Fig. 4. Time Comparison Among Conventional Full Database and Proposed Filtered Database

nition rate whereas K_{100} and K_{50} showed the improvement of performance.

When performing keypoint filtering, the bad keypoints causing miss-match are eliminated, which in turn increases the reliability of the match results. To prove this, the precision [90] in the feature-level was calculated. The precision can be calculated as the ratio between the number of the correspondence pairs obtained after matching and the correct matches, indicating the insignificant proportion of mass-match and significant proportion of correction match in the match results. The increase of the ratio between correct match and match results subsequently affects the performance of robust pose estimation.

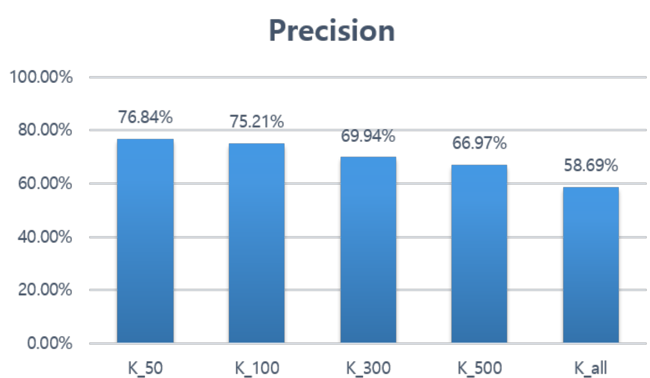

Fig. 6. Precision of filtered keypoint database

The results of precision are demonstrated in Table II and Table 6. The filtered keypoints sets showed higher precision compared to the whole of keypoints set (K_{all}). The number of the detected keypoints decreased but the ratio of correct match increased, which showed high precision. Such results are able to improve the speed and performance of robust pose estimation.

TABLE II
PRECISION OF FILTERED MATCHING

	K_{50}	K_{100}	K_{300}	K_{500}	K_{all}
Avg. Match Result	10.098	15.618	26.747	31.409	44.859
Avg. Correct Match	7.759	11.747	18.705	21.033	26.326
Precision	76.8%	75.2%	69.9%	67.0%	58.7%

V. CONCLUSION

The conclusion goes here.

APPENDIX A
PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

APPENDIX B

Appendix two text goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] H. Chang, C. S. G. Lee, Y.-H. Lu, and Y. Hu, "P-SLAM: Simultaneous localization and mapping with environmental-structure prediction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 281–293, Apr. 2007.

[2] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2161–2168.

[4] M. Brown and D. Lowe, "Recognising panoramas," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, Oct. 2003, pp. 1218–1225 vol.2.

[5] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," in *2010 IEEE Virtual Reality Conference (VR)*, Mar. 2010, pp. 211–218.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007*, Nov. 2007, pp. 225–234.

[7] D. Wagner, D. Schmalstieg, and H. Bischof, "Multiple target detection and tracking with guaranteed framerates on mobile phones," in *8th IEEE International Symposium on Mixed and Augmented Reality, 2009. ISMAR 2009*, Oct. 2009, pp. 57–64.

[8] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 54–62, Jun. 2006.

[9] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, vol. 1, Jun. 2004, pp. I–652–I–659 Vol.1.

[10] G. Carrera, J. Savage, and W. Mayol-Cuevas, "Robust feature descriptors for efficient vision-based tracking," in *Progress in Pattern Recognition, Image Analysis and Applications*, ser. Lecture Notes in Computer Science, L. Rueda, D. Mery, and J. Kittler, Eds. Springer Berlin Heidelberg, Jan. 2007, no. 4756, pp. 251–260. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-76725-1_27

[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[12] M. Muja and D. Lowe, "Fast matching of binary features," in *2012 Ninth Conference on Computer and Robot Vision (CRV)*, May 2012, pp. 404–410.

[13] Y. Yu, K. Huang, W. Chen, and T. Tan, "A novel algorithm for view and illumination invariant image matching," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 229–240, Jan. 2012.

[14] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[15] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, May 1997. [Online]. Available: http://link.springer.com/article/10.1023/A%3A1007963824710

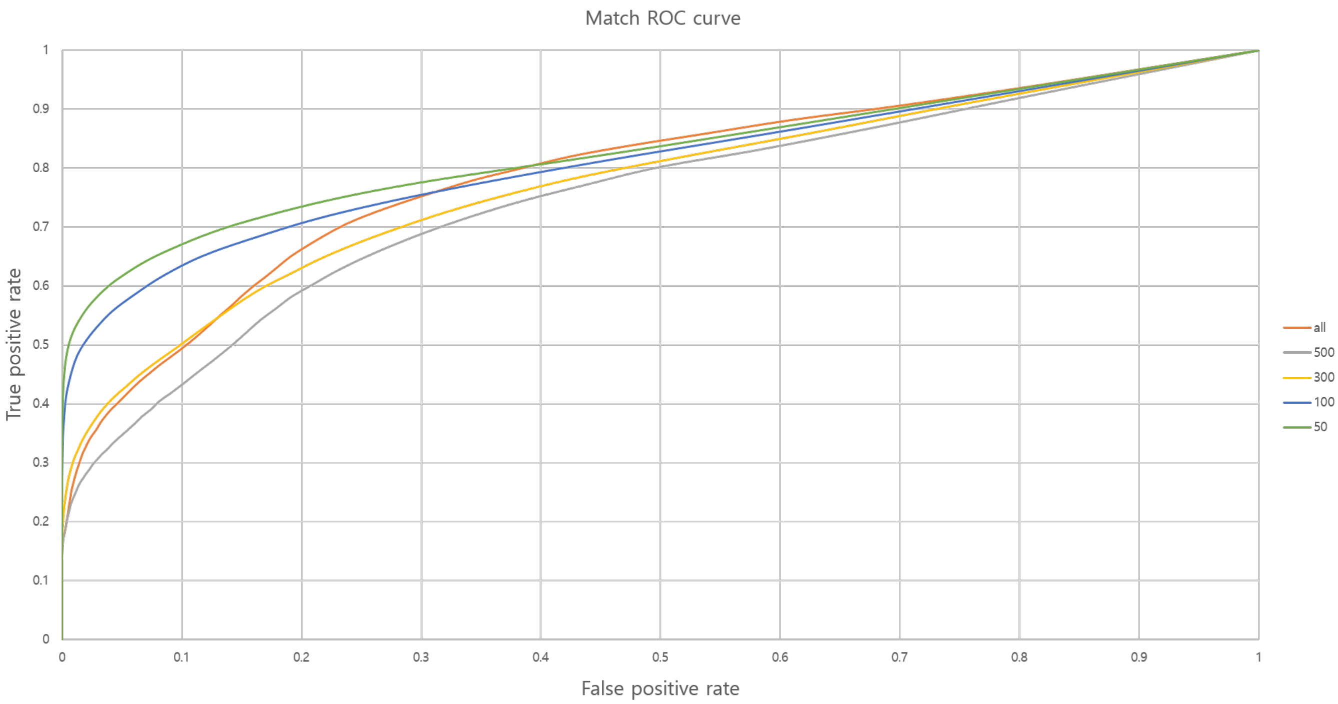

Fig. 5. ROC curve for match rate

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://link.springer.com/article/10.1023/B%3AVISI.0000029664.99615.94

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314207001555

[18] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1980, AAI8024717.

[19] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, vol. 1, 2001, pp. 525–531 vol.1.

[20] J. Shi and C. Tomasi, "Good features to track," in *, 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94*, Jun. 1994, pp. 593–600.

[21] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." in *Proceedings of DARPA Image Understanding Workshop*, Apr. 1981, pp. 121–130.

[22] P. R. Beaudet, "Rotationally invariant image operators," in *Proceedings of the International Joint Conference on Pattern Recognition*, 1978, pp. 579–583.

[23] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, vol. 1, no. 2, pp. 95–102, Dec. 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167865582900204

[24] M. Trajković and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, no. 2, pp. 75–87, Feb. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885697000565

[25] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, Jan. 2006, no. 3951, pp. 430–443. [Online]. Available: http://link.springer.com/chapter/10.1007/11744023_34

[26] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, Jan. 2010.

[27] S. Taylor, E. Rosten, and T. Drummond, "Robust feature matching in 2.3 #x00b5;s," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009*, Jun. 2009, pp. 15–22.

[28] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Computer Vision – ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, Jan. 2010, no. 6312, pp. 183–196. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-15552-9_14

[29] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, vol. 21, pp. 224–270, 1994.

[30] J. L. Crowley, O. Riff, and J. H. Piater, "Fast computation of characteristic scale using a half-octave pyramid," in *In: Scale Space 03: 4th International Conference on Scale-Space theories in Computer Vision, Isle of Skye*, 2003.

[31] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision — ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin Heidelberg, Jan. 2002, no. 2350, pp. 128–142. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-47969-4_9

[32] M. Brown and D. Lowe, "Invariant features from interest point groups," in *In British Machine Vision Conference*, 2002, pp. 656–665.

[33] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?"," in *Computer Vision — ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin Heidelberg, Jan. 2002, no. 2350, pp. 414–431. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-47969-4_28

[34] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, Nov. 2005. [Online]. Available: http://link.springer.com/article/10.1007/s11263-005-3848-x

[35] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, Jul. 2007. [Online]. Available: http://link.springer.com/article/10.1007/s11263-006-9967-1

[36] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 2, Jun. 2004, pp. II–506–II–513 Vol.2.

[37] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Computer Vision – ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, Jan. 2010, no. 6314, pp. 778–792. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-15561-1_56

[38] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[39] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2012, pp. 510–517.

[40] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 IEEE International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 2564–2571.

[41] C. Schmid and R. Mohr, "Local greyvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 530–535, 1997.

[42] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[43] G. Carneiro and A. Jepson, "Multi-scale phase-based local features," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, vol. 1, Jun. 2003, pp. I–736–I–743 vol.1.

[44] G. Yu and J.-M. Morel, "A fully affine invariant image comparison method," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, Apr. 2009, pp. 1597–1600.

[45] Y. Ma, H. Xie, Z. Chen, Q. Dai, Y. Huang, and G. Ji, "Fast search of binary codes with distinctive bits," in *Advances in Multimedia Information Processing – PCM 2014*, ser. Lecture Notes in Computer Science, W. T. Ooi, C. G. M. Snoek, H. K. Tan, C.-K. Ho, B. Huet, and C.-W. Ngo, Eds. Springer International Publishing, Jan. 2014, no. 8879, pp. 274–283. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-13168-9_31

[46] M. Muja and D. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

[47] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975. [Online]. Available: http://doi.acm.org/10.1145/361002.361007

[48] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, Sep. 1977. [Online]. Available: http://doi.acm.org/10.1145/355744.355745

[49] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, vol. 45, no. 6, pp. 891–923, Nov. 1998. [Online]. Available: http://doi.acm.org/10.1145/293347.293348

[50] J. Beis and D. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings*, Jun. 1997, pp. 1000–1006.

[51] C. Silpa-Anan and R. Hartley, "Optimised KD-trees for fast image descriptor matching," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, Jun. 2008, pp. 1–8.

[52] R. F. Sproull, "Refinements to nearest-neighbor searching in k-dimensional trees," *Algorithmica*, vol. 6, no. 1-6, pp. 579–589, Jun. 1991. [Online]. Available: http://link.springer.com/article/10.1007/BF01759061

[53] S. Dasgupta and Y. Freund, "Random projection trees and low dimensional manifolds," in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, ser. STOC '08. New York, NY, USA: ACM, 2008, pp. 537–546. [Online]. Available: http://doi.acm.org/10.1145/1374376.1374452

[54] Y. Jia, J. Wang, G. Zeng, H. Zha, and X.-S. Hua, "Optimizing kd-trees for scalable visual descriptor indexing," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2010, pp. 3392–3399.

[55] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *In VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.

[56] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Transactions on Computers*, vol. C-24, no. 7, pp. 750–753, Jul. 1975.

[57] S. Brin, "Near neighbor search in large metric spaces," Zurich, Switzerland, 1995. [Online]. Available: http://ilpubs.stanford.edu:8090/113/

[58] A. W. Moore, "The anchors hierarchy: Using the triangle inequality to survive high dimensional data," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 397–405. [Online]. Available: http://dl.acm.org/citation.cfm?id=2073946.2073993

[59] T. Liu, A. W. Moore, A. Gray, and K. Yang, "An investigation of practical approximate nearest neighbor algorithms." MIT Press, 2004, pp. 825–832.

[60] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, Jun. 2007, pp. 1–7.

[61] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1465–1479, Sep. 2006.

[62] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, Jun. 2007, pp. 1–8.

[63] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 448–461, Mar. 2010.

[64] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International journal of approximate reasoning*, vol. 50, no. 7, pp. 969–978, 2009. [Online]. Available: http://cat.inist.fr/?aModele=afficheN&cpsidt=21738718

[65] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, Jun. 2008, pp. 1–8.

[66] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1753–1760. [Online]. Available: http://papers.nips.cc/paper/3383-spectral-hashing.pdf

[67] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, ser. VLDB '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 518–529. [Online]. Available: http://dl.acm.org/citation.cfm?id=645925.671516

[68] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *47th Annual IEEE Symposium on Foundations of Computer Science, 2006. FOCS '06*, Oct. 2006, pp. 459–468.

[69] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: Efficient indexing for high-dimensional similarity search," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, ser. VLDB '07. Vienna, Austria: VLDB Endowment, 2007, pp. 950–961. [Online]. Available: http://dl.acm.org/citation.cfm?id=1325851.1325958

[70] M. Bawa, T. Condie, and P. Ganesan, "LSH forest: Self-tuning indexes for similarity search," in *Proceedings of the 14th International Conference on World Wide Web*, ser. WWW '05. New York, NY, USA: ACM, 2005, pp. 651–660. [Online]. Available: http://doi.acm.org/10.1145/1060745.1060840

[71] J. A. Noble, "Descriptions of image surfaces." Ph.D. dissertation, University of Oxford, 1989.

[72] C. Kenney, B. Manjunath, M. Zuliani, G. Hewer, and A. Van Nevel, "A condition number for point matching with application to registration and postregistration error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1437–1454, Nov. 2003.

[73] J. J. Foo and R. Sinha, "Pruning SIFT for scalable near-duplicate image matching," in *Proceedings of the Eighteenth Conference on Australasian Database - Volume 63*, ser. ADC '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 63–71. [Online]. Available: http://dl.acm.org/citation.cfm?id=1273730.1273738

[74] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition, 2006. ICPR 2006*, vol. 3, 2006, pp. 850–855.

[75] M. Brown, R. Szeliski, and S. Winder, "Multi-image matching using multi-scale oriented patches," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 510–517. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.235

[76] S. Gauglitz, L. Foschini, M. Turk, and T. Hollerer, “Efficiently selecting spatially distributed keypoints for visual tracking,” in *2011 18th IEEE International Conference on Image Processing (ICIP)*, Sep. 2011, pp. 1869–1872.

[77] M. Knapek, R. Oropeza, and D. Kriegman, “Selecting promising landmarks,” in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA ’00*, vol. 4, 2000, pp. 3771–3777 vol.4.

[78] A. Oerlemans and M. S. Lew, “Interest points based on maximization of distinctiveness,” in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, ser. MIR ’08. New York, NY, USA: ACM, 2008, pp. 202–207. [Online]. Available: http://doi.acm.org/10.1145/1460096.1460130

[79] G. Bleser and D. Stricker, “Advanced tracking through efficient image processing and visual-inertial sensor fusion,” in *IEEE Virtual Reality Conference, 2008. VR ’08*, Mar. 2008, pp. 137–144.

[80] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway, “Robust real-time visual SLAM using scale prediction and exemplar based feature description,” in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR ’07*, Jun. 2007, pp. 1–7.

[81] E. Eade and T. Drummond, “Scalable monocular SLAM,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Jun. 2006, pp. 469–476.

[82] T. Lee and T. Hollerer, “Hybrid feature tracking and user interaction for markerless augmented reality,” in *IEEE Virtual Reality Conference, 2008. VR ’08*, Mar. 2008, pp. 145–152.

[83] Y. Park, V. Lepetit, and W. Woo, “Multiple 3d object tracking for augmented reality,” in *7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008. ISMAR 2008*, Sep. 2008, pp. 117–120.

[84] S. Se, D. Lowe, and J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks,” *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, Aug. 2002. [Online]. Available: http://ijr.sagepub.com/content/21/8/735

[85] I. Skrypnyk and D. Lowe, “Scene modelling, recognition and tracking with invariant image features,” in *Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004. ISMAR 2004*, Nov. 2004, pp. 110–119.

[86] B. Williams, G. Klein, and I. Reid, “Real-time SLAM relocalisation,” in *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, Oct. 2007, pp. 1–8.

[87] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1936.tb02137.x/abstract

[88] “Seoul travel guide,” Feb. 2014. [Online]. Available: http://www.visitseoul.net/en/guide/ebook.do?_method=list&m=0004011019003&p=04

[89] H. Choi, G. C. Han, and I.-j. Kim, “Smart booklet: Tour guide system with mobile augmented reality,” in *2014 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2014, pp. 353–354.

[90] J. Heinly, E. Dunn, and J.-M. Frahm, “Comparative evaluation of binary features,” in *Computer Vision – ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Springer Berlin Heidelberg, Jan. 2012, pp. 759–773. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-33709-3_54

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.