COnfusion REduction (CORE) algorithm for local descriptors, floating-point and binary cases

Emilien Royer[a,b], Thibault Lelore[a,b], Frédéric Bouchara[a,b]

[a] *Aix Marseille Université, CNRS, ENSAM, LSIS UMR 7296, 13397 Marseille, France*
[b] *Université de Toulon, CNRS, LSIS UMR 7296, 83957 La Garde, France*

Abstract

In this paper, we propose a generic pre-filtering method of point descriptors which addresses the confusion problem due to repetitive patterns. This confusion often leads to wrong descriptor matches and prevents further processes such as object recognition, image indexation, super-resolution or stereo-vision. Our method sorts keypoints by their unicity without taking into account any visual element but the feature vectors's statistical properties thanks to a kernel density estimation approach. Both binary descriptors and floating point based descriptors are studied, regardless of their dimensions. Even if highly reduced in number, results show that keypoints subsets extracted are still relevant and our algorithm can be combined with classical post-processing methods.

Keywords: keypoints filtering, computer vision, feature matching, kernel density estimator

In computer vision, many applications share the same first steps known as keypoint extraction and associated features computation. These two steps have been

Email addresses: emilien.royer@univ-tln.fr (Emilien Royer), {thibault.lelore@gmail.com (Thibault Lelore), bouchara@univ-tln.fr (Frédéric Bouchara)

more and more studied over the last years from the result of an increasing need of, as example, efficient robotic vision or image retrieval. Some major contributions, such as the SIFT descriptor [1] by D. Lowe are based on oriented gradient histograms. Its high efficiency, proven [2], has brought it wide popularity as it is one of the most used feature descriptor, even being rewritten for GPU architectures several years ago [3]. It also has inspired many others such as SURF [4] or [5]. However, even for nowadays computation capabilities, this class of algorithms shares some relatively high computational cost which often prevents us from using them in real-time applications, especially with low-end hardware such as embedded devices.

Yet, the rise of the smartphone industry has increased the need for light-processing algorithms and less memory consumption. Thus, in 2010, Calonder et al. introduced the BRIEF descriptor [6], slightly inspired by Local Binary Pattern (LBP) [7], which has led the way to a new class of methods called binary descriptors like ORB [8], BRISK [9], FREAK [10] or D-BRIEF [11]. Originaly less efficient than the SIFT-based ones (but still, good enough for real applications) there has been a trend in keeping the fast processing aspect while improving the matching capabilities; the last ones, like BinBoost [12], BOLD [13] and LATCH [14], claim to have similar performances as the best floating-point descriptors. With the major exception of BinBoost and D-BRIEF, binary descriptors are often based on the simple but yet efficient following procedure: pixel pairs are sampled all over a small blured region of the chosen keypoint and for each pair the difference of the pixel values is computed, acting as an element of the feature vector. As it is common for feature descriptors, each new variation has brought slight modification of the original idea. For example, BRIEF originally uses different ways

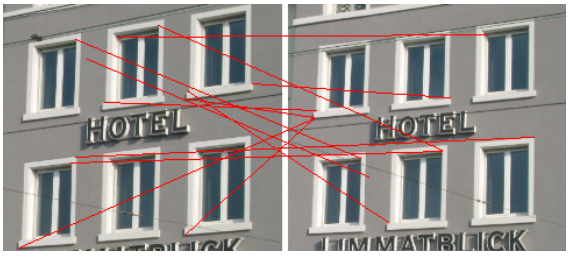


Figure 1: Example from the Zurich Building Image Database of repetitive patterns leading in ”good-false” matches with the SIFT descriptor.

of random pair sampling, ORB has proposed machine learning to learn efficient pair sampling and BRISK uses a mandatory pattern which is bio-inspired. These are feature descriptors modifications, but same is done with keypoint detection in order to improve keypoint selection. Again, ORB orders the FAST [15] responses by a Harris corner measure [16]. With our contribution, we propose a solution to both generally improve the selection and to address a specific case that we are presenting in the next section.

0.1. The repetitive patterns problem

A frequent and troublesome problem easily encountered when trying to match pairs in different images is the repetitive pattern case, as we can see in figure 1 : the exact same pattern is present in multiple occurrences within the image. These visual features make it highly responsive to saliency analysis, returning numerous keypoints that have almost the same feature vectors, which results in high confusion during matching phase. Usually, the mismatch problem is handled from a given putative point correspondence by different kinds of approaches. A first kind of methods is based on a robust statistic estimation such as LMS (Least Median of Squares) or M-estimators. In [17] Deriche *et al.* applied the LMS for the ro-

bust estimation of the fundamental matrix. In a similar approach Torr *et al* [18] proposed a method for the estimation of both the fundamental matrix and motion estimation. Another robust estimation methods can be found in the literature such as the algorithms proposed by Ma *et al* [19, 20].

Another kind of methods, known as resampling methods, act by trying to get a minimum subset of mismatch-free correspondence. Methods belonging to this category are usually extensions of the well known RANSAC (RANdom SAmple Consensus) [21] such as MLESAC [22] or SCRAMSAC [23]. We can also cite [24]. Other algorithms are based on different approaches as the ICF (Identifying point correspondences by Correspondence Function) proposed by Li *et al* [25].

Another way to consider the mismatch problem is to filter out repetitive patterns in each image. Such a priori approaches may be combined with the previous methods that are performed a posteriori from a given putative point correspondence. When looking at the literature, detecting repetitive pattern is a known issue in several different applications although it is reputed to be difficult. Repetitive structures can be detected through symmetry analysis [26, 27, 28] and despite being mostly 2D analysis, recent propositions try to take into account non-planar 3D repetitive elements [29, 30]. Mortensen *et al.* enrich the SIFT descriptor with information about the image global context [31], inspired by shape contexts [32]. The SERP [33] descriptor and the CAKE [34] keypoint extractor both rely on kernel density estimation [35]. The first one uses mean-shift clustering on SURF descriptors, whereas the second one builds a new keypoint extractor based on Shannon's definition of information. As we're about to see in the next section, our approach does also rely on kernel density estimation but in a different way.

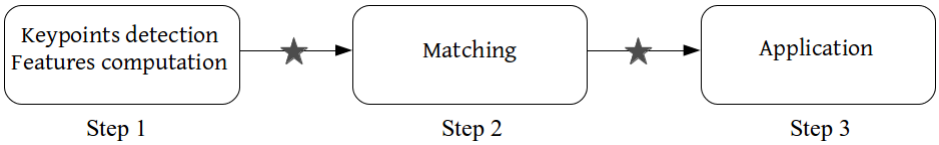


Figure 2: Classical processing pipeline of an image processing application requiring keypoints detection and features computation. $\star$ symbols corresponds to generic *pre* and *post* processing steps.

As we can see by passing in review the methods found in the literature, they are to be applied during different steps of the usual processing pipeline. Moreover, the choice of some of these steps are not without consequences ; for example methods like CAKE entirely replace the detection step since it's a different detector. This prevents us from using other classical detectors that might have different appreciable characteristics. If we look at figure 2, it is easy to understand that the most handy algorithms are the ones which are designed to be applied in-between steps 1 & 2 and 2 & 3 which we call here respectively pre and post processing / filtering approaches (the matching step being the "in-processing" one) such as SCRAMSAC : they provide us with genericity since they can be used with the different major, classical algorithms without altering a classical processing pipeline. With our notations, the matching step would be called the "in-processing" one. Considering the particularities of high-dimensional spaces, matching features is not a trivial issue and this subject is still being studied with dedication. For example, very important works include the FLANN (Fast Library for Approximate Nearest Neighbors) collection of algorithms, and Rabin *et al.* with [36] and [37] with their clever adaptation of the *Earth Mover's Distance* to circular histograms thanks to a dissimilarity measure.

Since SCRAMSAC objective is about building a subset of matches before

applying the RANSAC algorithm, we can classify it as a post-processing method. However, generic pre-processing methods with the objective of building a subset of features before the matching step, whatever the feature descriptor algorithm employed and with the aim of reducing the confusion appear to be lacking.

Therefore, this is the goal of this paper: we propose a new approach to cope with the keypoints confusion problem. We don't take into account the keypoints visual properties since they may vary with the type of extractor chosen, but instead we analyze the statistics properties of their associated feature vectors. We estimate a numerical value that is associated to the confusion risk of a given feature vector between another vector in a different image. With this criterion, we can, then, sort the keypoints from low confusion risk, to high confusion risk. With the right threshold, we can thus decide which points should be discarded and which ones should be kept. The rest of the paper is organized as follow: Section 1.1 will present an overview of our proposed method. In section 1.2 we will explain the criterion computation. Section 1.3 will address the problem of threshold setting. Finally, Section section 2.1 and 3 will respectively present results and conclusions.

Further in the text we will use the following notation: we let $P_x(y)$ be the probability $\Pr(x = y)$. Vectors are denoted by lower bold letters such as $\mathbf{u}$ or $\mathbf{v_i}$. The d^{th} component of such vectors are denoted u_d and $v_{\mathbf{i}d}$.

1. Proposed Method

1.1. Overview

. Let I be the image resulting of the observation (with a camera) of a specific scene. Let I' be (a potential) another observation of the same scene in which

changes result from various transformations such as perspective changes, light modifications, etc. In our model, I is deterministic whereas I' is a potential (not yet observed) different version of I and is hence considered to be stochastic. Let now $\mathbf{u_i}_{,\ \mathbf{i}\in\{\mathbf{1},\ldots,\mathbf{N}\}}$ be D-dimensional feature vectors computed on $\mathbf{N}$ keypoints of I and let $\mathbf{u'_i}_{,\ \mathbf{i}\in\{\mathbf{1},\ldots,\mathbf{N}\}}$, be their $\mathbf{N}$ respective equivalents in I'. We assume that even if descriptors try to be invariant as much as possible to most transformations, each feature vector in image I is subject to slight variations that we can assimilate in image I' as randomness. By doing so we consider $\mathbf{u'_i}$ as random vectors and we shall define a criterion associated to each keypoint of I that characterizes the confusion risk, i.e. a value correlated to the probability that in I', a vector $\mathbf{u'_j}_{,\ \mathbf{j}\neq\mathbf{i}}$ is closer to $\mathbf{u_i}$ than $\mathbf{u_j}$.

1.2. Criterion computation

For each keypoint $\mathbf{i}$ of I we define $C_\mathbf{i}$, the criterion, as the probability density that any other random $\mathbf{u'_j}_{,\ \mathbf{j}\neq\mathbf{i}}$ is equal to $\mathbf{u_i}$, *i.e.* $P_{\mathbf{u'_j}_{,\ \mathbf{j}\neq\mathbf{i}}}(\mathbf{u_i})$. This density should act as a criterion for separating relevant and high confusion risk keypoints.

From this definition, we can write:

$$\begin{aligned} C_\mathbf{i} \equiv P_{\mathbf{u'_j}_{,\ \mathbf{j}\neq\mathbf{i}}}(\mathbf{u_i}) &= \sum_{\mathbf{j}\neq\mathbf{i}} \Pr(\mathbf{k}=\mathbf{j}, \mathbf{u}=\mathbf{u_i}) && (1) \\ &= \sum_{\mathbf{j}\neq\mathbf{i}} P_{\mathbf{k},\ \mathbf{k}\neq\mathbf{i}}(\mathbf{j}) P_{\mathbf{u}/\mathbf{j}}(\mathbf{u_i}) && (2) \end{aligned}$$

where $P_{\mathbf{k},\ \mathbf{k}\neq\mathbf{i}}(\mathbf{j})$ denotes the probability of choosing keypoint $\mathbf{j}$ and $P_{\mathbf{u}/\mathbf{j}}(.)$ is the probability density function (pdf) of the feature vector given the keypoint number. We simply assume $P_{\mathbf{k},\ \mathbf{k}\neq\mathbf{i}}(\mathbf{j}) = \frac{1}{\mathbf{N}-1}$ (the $\mathbf{N}-1$ keypoints are equiprobable)

and we shall denote $P_{\mathbf{u/j}}(\mathbf{u}) = K\left(\left|\mathbf{u}-\mathbf{u_j}\right|\right)$ ($P_{\mathbf{u/j}}(\mathbf{u})$ is assumed to depend only on the distance $\left|\mathbf{u}-\mathbf{u_j}\right|$)

We thus obtain the estimation of $C_\mathbf{i}$ by the classical Parzen-Rosenblatt kernel density estimator (KDE):

$$C_\mathbf{i} = \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j}\neq\mathbf{i}} K\left(\left|\mathbf{u_i}-\mathbf{u_j}\right|\right) \tag{3}$$

By labeling each keypoint $\mathbf{i}$ with is $C_\mathbf{i}$ value, a confusion reduction (CORE) algorithm is easily designed:

Steps *(a)* and *(b)* are explained in next subsections.

1.2.1. Floating point case

We suppose that the vector variation causes are numerous and are either from natural origins or can be considered as such. Therefore, it makes sense to consider this behavior to be Gaussian. With this assumption we can define K as the classical D-dimensional uncorrelated Gaussian Kernel:

$$K(\mathbf{d}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^D \exp\left(-\frac{\mathbf{d}^2}{2\sigma^2}\right) \tag{4}$$

Thus, the criterion formula is :

$$C_\mathbf{i} = \frac{1}{(\mathbf{N}-\mathbf{1})\left(\sigma\sqrt{2\pi}\right)^D} \sum_{\mathbf{j}\neq\mathbf{i}} \exp\left(-\frac{\mathtt{d_E}(\mathbf{u_i},\mathbf{u_j})^2}{2\sigma^2}\right) \tag{5}$$

where $\mathtt{d_E}(\mathbf{u_i},\mathbf{u_j}) = \sqrt{\|\mathbf{u_i}-\mathbf{u_j}\|}$ is the Euclidean distance between vector $\mathbf{u_i}$ and $\mathbf{u_j}$. σ is the average modification of a vector component in the feature's space. Its value is specific to each feature descriptor algorithm and should be evaluated once; for example we found it to be roughly around 32.125 with the SIFT descriptor.

1.2.2. Binary case

In the binary case $\mathbf{u} = \left(u_{d,\ d\in\{1,\ldots,D\}}\right)$ is a binary vector and we let $\mu_d = \Pr(u_d \neq u'_d)$ be the probability that the value of the d^{th} component is different between the two images. In the following, we shall assume μ_d independant of the component and we shall drop the index d. $P_{\mathbf{u}/\mathbf{j}}(\mathbf{u})$ is then given by a Bernouilli distribution and $K(.)$ can therefore be written in the form:

$$K(\mathbf{u}) = \prod_{d=1}^{D} \mu^{u_d}(1-\mu)^{1-u_d} \tag{6}$$

which leads to the following expression for $C_{\mathbf{i}}$:

$$\begin{aligned} C_{\mathbf{i}} &= \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j}\neq\mathbf{i}} \prod_{d=1}^{D} \mu^{u_{\mathbf{i}d}\oplus u_{\mathbf{j}d}}(1-\mu)^{1-u_{\mathbf{i}d}\oplus u_{\mathbf{j}d}} & (7)\\ &= \frac{1}{(\mathbf{N}-1)} \sum_{\mathbf{j}\neq\mathbf{i}} \mu^{\mathtt{d_H}(\mathbf{u_i},\mathbf{u_j})}(1-\mu)^{D-\mathtt{d_H}(\mathbf{u_i},\mathbf{u_j})} & (8) \end{aligned}$$

where $u_{\mathbf{i}d} \oplus u_{\mathbf{j}d}$ represents the exclusive disjunction between $u_{\mathbf{i}d}$ and $u_{\mathbf{j}d}$ and $\mathtt{d_H}$ is the hamming distance.

1.3. Threshold estimation

Since we can associate a numerical value tied to the confusion risk for each feature vector, an immediate method to extract a subset of keypoints would be to sort them according to their $C_{\mathbf{i}}$ value and only keep the n_{th} first. However, it is quick realized that such a solution would not be relevant, it lacks the genericity spirit that lead the developments of our method : in two different situations, the n_{th} first points would not have the same $C_{\mathbf{i}}$ value if the overall confusion risk is different. Hence, again, a relevant value of the threshold C_{th} to apply on the $C_{\mathbf{i},\ \mathbf{i}\in\{1,\ldots,\mathbf{N}\}}$

can be estimated by considering the confusion problem with a probabilistic point of view.

1.3.1. Floating point case

With the notations of the previous section, let $\mathbf{u_i}$ and $\mathbf{u'_i}$ be the feature vectors computed on the same keypoint $\mathbf{i}$ of two different versions of a scene. Let now $\mathbf{v_i} = \mathbf{u'_i} - \mathbf{u_i}$, $\mathbf{v_j} = \mathbf{u'_j} - \mathbf{u_i}$, $d_\mathbf{i}^2 = \|\mathbf{v_i}\|^2$ and $d_\mathbf{j}^2 = \|\mathbf{v_j}\|^2$ where $\mathbf{u_j}$, $\mathbf{u'_j}$ are the corresponding feature vectors computed on another keypoint $\mathbf{j}$.

To estimate C_{th} we shall express $C_\mathbf{i}$ as a function of $p = \Pr(d_\mathbf{j}^2 < d_\mathbf{i}^2)$ the probability of a confusion. In our approach, p is a user-defined parameter which tunes an acceptable confusion rate. To derive this relation we need first to estimate $P_{d_\mathbf{j}^2}(.)$, (and hence $P_{\mathbf{v_j}}(.)$) which is governed by the distribution of the $\mathbf{u_j}$, $\mathbf{j}\neq\mathbf{i}$. However, we shall assume that p only depends on the behavior of $P_{\mathbf{v_j}}(.)$ in a small neighborhood of $\mathbf{u_i}$. We hence approximate $P_{\mathbf{v_j}}(.)$ by a D-dimensional uncorrelated Gaussian distribution $N(.;0,\Sigma_{\mathbf{v_j}})$ of which the central value $\Pr(\mathbf{v_j} = 0) = P_{\mathbf{v_j}}(0) = C_\mathbf{i}$ thanks to the definition of $C_\mathbf{i}$ given in the previous section. The diagonal element $\sigma_{\mathbf{v_j}}$ of the covariance matrix $\Sigma_{\mathbf{v_j}}$ is simply related to $C_\mathbf{i}$ by considering the normalization condition on $P_{\mathbf{v_j}}(.)$ which can be written:

$$C_\mathbf{i} = (2\pi\sigma_{\mathbf{v_j}}^2)^{-D/2} \tag{9}$$

From this assumption, $P_{d_\mathbf{j}^2}(.)$ is given by a chi-squared distribution with D degrees of freedom which can be approximated by a Gaussian law $N(.;E_\mathbf{j},\sigma_\mathbf{j})$ due to the large value of D. The values of $E_\mathbf{j}$ and $\sigma_\mathbf{j}$ are classically related to the values of $\sigma_{\mathbf{v_j}}$ and D by: $E_\mathbf{j} = \sigma_{\mathbf{v_j}}^2 D$ and $\sigma_\mathbf{j} = \sigma_{\mathbf{v_j}}^2\sqrt{2D}$.

Thanks to the Gaussian assumption on the $\mathbf{u}'_\mathbf{i}$ values and using the same considerations as before, we can also approximate $P_{d_\mathbf{i}^2}$ by a Gaussian law $N(.;E_\mathbf{i},\sigma_\mathbf{i})$ with $E_\mathbf{i} = \sigma^2 D$ and $\sigma_\mathbf{i} = \sigma^2\sqrt{2D}$.

From these definitions we can now write:

$$
\begin{aligned}
p &= \Pr(d_\mathbf{j}^2 < d_\mathbf{i}^2) && (10)\\
&= \int_{-\infty}^{\infty}\int_{x}^{\infty} P_{d_\mathbf{j}^2}(x)P_{d_\mathbf{i}^2}(y)dydx && (11)\\
&= \int_{-\infty}^{\infty}\int_{x}^{\infty} N(x;E_\mathbf{j},\sigma_\mathbf{j})N(y;E_\mathbf{i},\sigma_\mathbf{i})dydx && (12)\\
&= \frac{1}{2} - \frac{1}{2\sigma_\mathbf{j}\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\left[\frac{-(x-E_\mathbf{j})^2}{2\sigma_\mathbf{j}^2}\right]\times \\
&\quad \operatorname{erf}\left[\frac{x-E_\mathbf{i}}{\sigma_\mathbf{i}\sqrt{2}}\right]dx && (13)\\
&= \frac{1}{2}\left[1+\operatorname{erf}\left(\frac{E_\mathbf{i}-E_\mathbf{j}}{\sqrt{2(\sigma_\mathbf{i}^2+\sigma_\mathbf{j}^2)}}\right)\right] && (14)
\end{aligned}
$$

After a straightforward, albeit a bit tedious, calculation we obtain from (14):

$$
\sigma_{v_\mathbf{j}}^2 = \sigma^2\frac{D+2\sqrt{\gamma(D-\gamma)}}{D-2\gamma} \qquad (15)
$$

$$
\text{with} \qquad \gamma = 2\left(\operatorname{erf}^{-1}(2p-1)\right)^2 \qquad (16)
$$

From (15) and (16), the threshold C_{th} which corresponds to a specific p is then given by (9).

1.3.2. Binary case

Similarly to the floating point case, we let $\mathbf{v_i} = \mathbf{u'_i} \oplus \mathbf{u_i}$, $\mathbf{v_j} = \mathbf{u'_j} \oplus \mathbf{u_i}$, $d_\mathbf{i} = \mathtt{d_H}(\mathbf{u_i}, \mathbf{u'_i})$ and $d_\mathbf{j} = \mathtt{d_H}(\mathbf{u_i}, \mathbf{u'_j})$.

We equivalently assume that p only depends on a small neighborhood of $\mathbf{u_i}$ and we locally modeled $P_{\mathbf{v_j}}(.)$ with a Bernouilli distribution:

$$P_{v_\mathbf{j}}(\mathbf{u}) = \prod_{d=1}^{D} \nu^{u_d}(1-\nu)^{1-u_d} \tag{17}$$

From this assumption we get the following relation which links $C_\mathbf{i}$ with ν:

$$C_\mathbf{i} = (1-\nu)^D \tag{18}$$

Considering the Bernouilli expressions of $P_{\mathbf{v_i}}(.)$ and $P_{\mathbf{v_j}}(.)$, $P_{d_\mathbf{i}}(.)$ and $P_{d_\mathbf{j}}(.)$ are given by binomial distributions that we shall approximate by Poisson distributions with parameters $\lambda_i = D\mu$ and $\lambda_j = D\nu$ respectively.

The difference $d_{\mathbf{j}i} = d_\mathbf{j} - d_\mathbf{i}$ between two Poisson distributed random numbers is Skellam distributed [38]. We then get:

$$P_{d_{\mathbf{j}i}}(d) = e^{-(\lambda_j+\lambda_i)} \left(\frac{\lambda_j}{\lambda_i}\right)^{d/2} I_d\left(2\sqrt{\lambda_j\lambda_i}\right) \tag{19}$$

with I_d the modified Bessel function. The Skellam distribution is well approximated by the normal distribution $N(.; \lambda_j - \lambda_i, \sqrt{\lambda_j + \lambda_i})$ which leads to:

$$
\begin{aligned}
p &= \Pr(d_{\mathbf{j}} < d_{\mathbf{i}}) && (20)\\
&= \int_{-\infty}^{0} N(x; \lambda_j - \lambda_i, \sqrt{\lambda_j + \lambda_i})dx && (21)\\
&= \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{D(\mu - \nu)}{\sqrt{2D(\nu + \mu)}}\right)\right] && (22)
\end{aligned}
$$

By a similar calculation as previously, we finally obtain:

$$
\begin{aligned}
\nu &= \frac{2\mu D + \gamma + \sqrt{\gamma(8\mu D + \gamma)}}{2D} \text{ if } p \in [0, 0.5[&& (23)\\
\nu &= \frac{2\mu D + \gamma - \sqrt{\gamma(8\mu D + \gamma)}}{2D} \text{ if } p \in [0.5, 1[&& (24)
\end{aligned}
$$

with γ given by (16).

1.4. Computational cost

Algorithm 1 and equation 3 seem to imply a significant computational cost of our proposed filter. Therefore, even if we do not analyze the processing times in our following experiments it is a matter that should be discussed here. Since we have to compute distances for each possible feature vector couples in a given image, a straightforward implementation should have a complexity of $O(N^2)$ (with N the number of keypoints in filtered image). Considering the symmetry property of the distance, the computation time may be simply reduced by half if said distances are stored somewhere in memory. For real-time applications, however a parallel impementation (on a GPU architecture for instance) is straightforward. In addition, applications such as homography estimation will clearly gain from having to deal with a reduced set size and hence, computation time spent by the CORE algorithm may be partially compensated at the matching step.

All in all, these observations lead us to think that even if a non-planned CORE algorithm integration in a processing pipeline might be costful, some small careful plannings which are not highly complex workarounds like storing the distances at first computation should easily negates it.

2. Results

A direct application of CORE filtering for SIFT detector and descriptor can be seen with figure 3 where interesting patterns are observed: the vast majority of the chessboard image's points are removed except for some on the corners whereas the ones on the photograph are mostly kept. This behavior is confirmed on the less obvious and simple Zurich image where the keypoints removed are mostly located on the repetitive windows patterns. Last, the text document image show clustered locations of kept points and many are located on particular places such as titles.

To better understand the dynamic of thresholding the features and the repartition of the $C_{\mathbf{i}}$ values, we can refer to figure 4 which shows the usefulness of our thresholding approach: as stated previously, different images have different responses of confusion risk and thus deriving the C_{th} from p filters accordingly.

But these are only visual observations. For validating our contribution, we're looking to prove that our algorithm does actually extract a better keypoints subset less subject to confusion by analyzing the features matching results by brute-force matching between two different images of the same scene. For most of our following experiments we will apply the Lowe's ratio test [1] to keep only high-quality feature matches: we reject poor matches by computing the ratio between the best and second-best match (labeled 2NN for 2 nearest neighbors). If the ratio is below

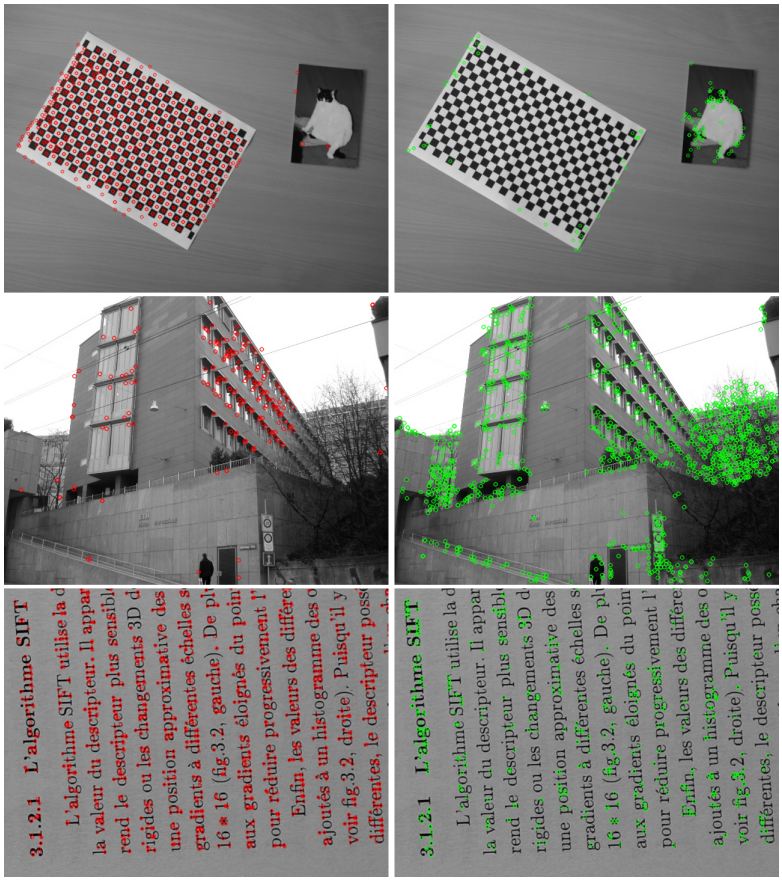


Figure 3: Examples of the 3 image types used in this paper and the filtering results with CORE algorithm of SIFT points (keypoints and features), left shows keypoints removed, right are keypoints kept. $p = 0.1$.

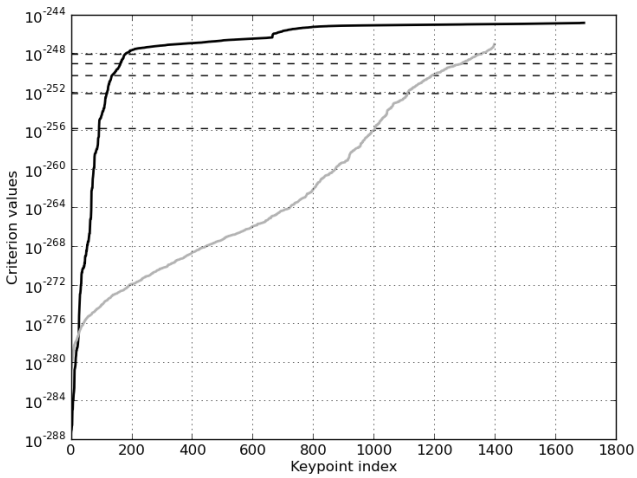


Figure 4: sorted $C_{\mathbf{i}}$ values of the first two images (Chess and Zurich) of figure 3 with SIFT points, respectively in black and gray. Dashed horizontal lines from top to bottom correspond to threshold values with $p = 0.20, 0.15, 0.10, 0.05, 0.01$. Points above respective thresholds are discarded.

a given threshold (we use 0.8), the match is discarded as being low-quality.

2.1. Floating-point case

We first ask an operator to manually evaluate each matches involving 9 image couples of the Zurich images database and 2 personal ones with chess patterns such as seen with figure 3 in 3 different scenarios, SIFT algorithm: without any filtering (plain full sets of keypoints and matches), with 2NN post-filtering and with CORE pre-filtering ($p = 0.1$) and 2NN post-filtering. Results are shown with table 1.

We can see that our contribution globally improves the good matching ratio: we find an average increasing value of 8.52% for the Zurich images. Images 4.*c* and 4.*i* show slight improvements (with respectively 1.13% and 2.72% ratio increasing) while the other ones extracted from this dataset range from 6.22% to 13.8%. An explanation could come from contextual information from the scene that could prevent some confusion. The chessboard images that hardly benefit from contextual information at all and contain real repetition jump with respectively 36.99% and 50.46%.

From now, we will focus on the application of estimating the underlying transformation between the image couples with the RANSAC algorithm. As a similar approach as used by SCRAMSAC, we evaluate the quality of the transformation found with the inlier ratio measure, *i.e.*, matches consistent with it.

We apply our next experiments on a personal set of 10 couples of document images captured by a smartphone camera. Printed document images are very good candidates for confusion reduction due to the letters and words repetitions. Moreover, their visual properties make them highly responsive to saliency analysis, resulting in a profusion of keypoints returned; usually around 30.000 for a

2560x1920 picture with default SIFT parameters. Thus, we also test our method as a way of reducing huge keypoint sets without relying on visual analysis. We proceed as follows: for each image pair, we apply our CORE algorithm on the keypoints returned by SIFT. This returns a reduced keypoints set with which we establish correspondences by brute-force matching. We then use the RANSAC algorithm to estimate the fundamental matrix and analyze the inlier ratio. For a fair comparison, we do the same with another keypoints subset by following Lowe idea of saliency analysis by a contrast threshold so we end up with a different keypoint set with equal size. On both of these approaches, we also apply the SCRAMSAC test to see how its matching filter behaves with these two different methods. At last, to serve as a control test we extract a random keypoint subset with same size in order to prove that our method (as well as Lowe's one) is better and makes more sense than randomness. We repeat this for different p values, respectively 0.5, 0.25, 0.15, 0.10 and 0.05.

In order to get results from an alternative approach, we have also achieved experiments with a filtering method of the keypoint set based on a mean shift clustering. Mean shift clustering is used in particular by the SERP algorithm to detect repeated pattern in an image [33].

Results are presented with figure 5. We see that for every p value, the number of inliers is always greater than other subsets of equal size resulting from saliency analysis. Moreover, with small p values (between 0.25 and 0.05), inlier ratio is always improved by CORE pre-processing and starting with $p = 0.15$, even if these processes take place during different places in the processing pipeline, it is worth noting that CORE pre-processing alone is doing better than post-processing

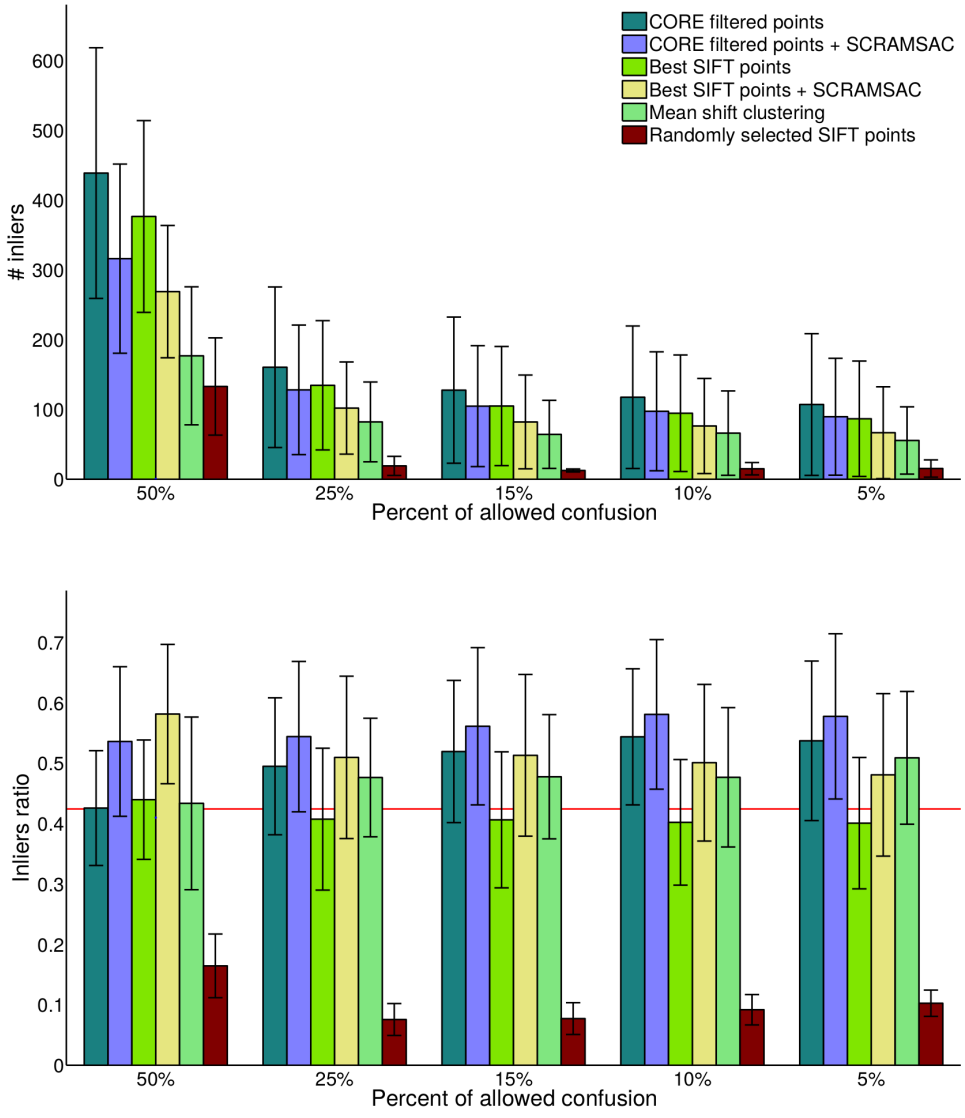


Figure 5: Average results inlier ratio evaluation with different filters. For each p value, we compare the results with subsets of equal size. The number of kept points for each image is computed by the CORE algorithm for a given pencentage of confusion (see the text for more details). Top: raw numbers of inliers, bottom: inlier ratio. Horizontal red line corresponds to SIFT inlier ratio without any filtering.

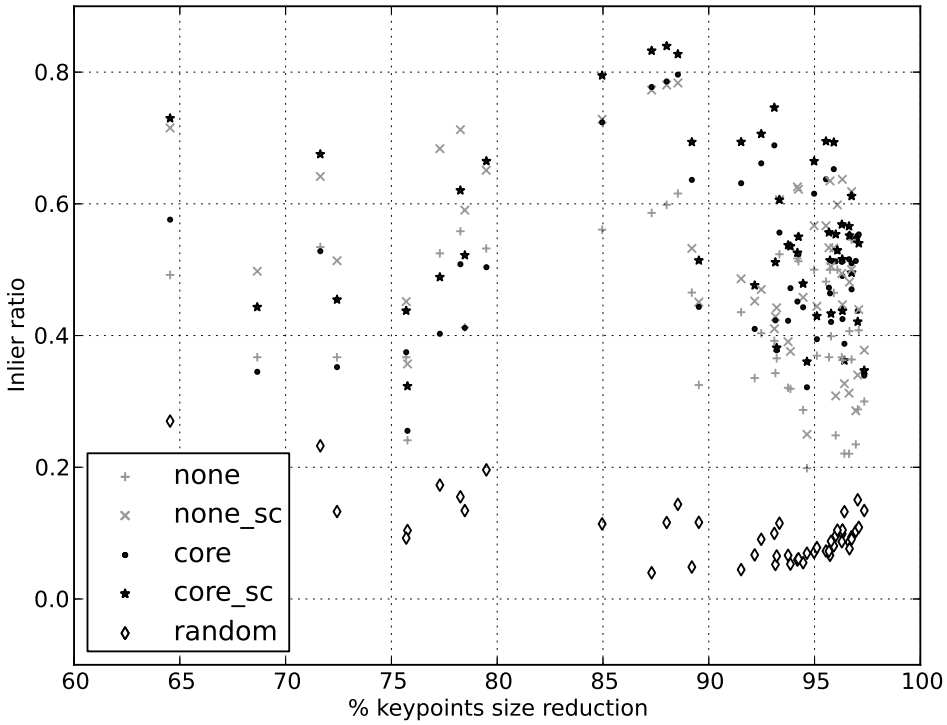


Figure 6: Individual inlier ratio results for each images couples and keypoints subsets with corresponding filterings as a function of original keypoints sets size reduction, percent based.

SCRAMSAC.

However, for $p = 0.5$ (50% of confusion tolerated), the inlier ratio is actually smaller with our method. This could come from the large confusion tolerated that doesn't remove enough keypoints: we don't take advantage of confusion reduction and some very similar keypoints were removed whereas their feature vector transformation may have not been enough to generate confusion. So we recommend using p values being inferior to 0.25 and best results seem to be achieved with 0.10. Not studied here, another advantage of our algorithm would be the speed-up gained during matching phase and model estimation as we observed the average computation time to be 20 times faster than without filtering. Finally, it is worth noting that our pre-processing filter (CORE) behaves well with post-processing (SCRAMSAC) by always increasing the inlier ratio, regardless of the p value used and the poor results from control test based on randomness prove the relevance of pre-processing.

2.2. Binary case

. Considering the trending topic of binary features we will broaden our analysis with multiple descriptors and detectors while remaining on the same inlier ratio evaluation. We chose four classical descriptors which showed increasing complexity with their chronological order of apparition. Namely BRIEF with random pair sampling, ORB with maching learned pair sampling, BRISK with handcrafted sampling pattern and FREAK with a bio-inspired one. Chosen detectors are SURF and BRISK for the relatively high number of responses and ORB for its HARRIS corner measure ordering.

. First, let's consider another way of choosing the μ parameter. For a given p value, we can plot the inlier ratio and number of matches kept after filtering as a function of μ. High values should indicate us good parameters. Since the floating-point evaluation showed us that our chess images are very good candidates for the confusion issue we will focus our evaluation on these and we will use a restrictive p value of 0.05. We apply the RANSAC algorithm with three methods: none (plain bruteforce matching, labeled *plain* further in the text), 2nn and cross-check. Results are given in figure 9. As we can see, by increasing μ we increase the number of removed keypoints, thus increasing the inlier ratio by removing high-confusion-risk points until an extrema is reached. From this extrema, removing more points is unefficient since we perform unwanted filtering on good points. This is easily explicated by the fact that μ is the state-switching probability of one bit in a feature vector; therefore, the higher μ, the higher we consider a keypoint to be from the high-confusion risk class.

This gives us an indication of valid parameters. From this, we can plot the inlier ratio as a function of p. A first example is shown with figure 7 for the chess images with the SURF detector and ORB descriptor. Again we can observe the expected behavior: by removing keypoints leading to confusion, the average inlier ratio is increased.

. Now, let's plot the same evaluation for our four descriptors and three detectors on the document text images, with 2nn filtering. This is given in figure 10. An interesting ascertainment is the fact that the descriptors are not equal on the behavior of confusion reduction when we apply them with different detectors. For example, BRISK shares the same behavior with all detectors used: the first keypoint subsets extracted is always above the non-filtering approach but increasing p

leads to converge the ratios towards the non-filtering value. BRIEF seems to give average results. This could be explained by the fact that it was the first modern binary descriptor; now a bit outdated, its simple mechanism of random sampling the pixel pairs might proves to be less discriminant on text document images without spots that particularly stand out against the others. At last, results associated with the ORB detectors give almost always poorer results, even when comparing non-filtering ratios: since keypoints are ordered with a harris corner measure, it is not well suited for text document images with sharp angles and high contrast, thus loosing discriminative power of locations. Only BRISK manages to benefit from confusion reduction which could imply the high discriminative power of this descriptor.

. Finally, to better understand the impact of μ choice, figure 8 shows us what happens when we increase this parameter: we can notice the inlier ratio curve shifts as the number of points kept rises slower.

From our evaluations, it seems that usable values should be included between 0.20 and 0.35 but of course the final choice might depends on the context.

Data: I : image input

Data: p : probability confusion tolerated

Data: D : descriptor dimension

Data: σ: average variance of (real-valued) descriptor's feature vectors

μ: (binary) feature vector bit-flip probability

Data: $C_{th} \leftarrow$ findThreshold(p, $\sigma|\mu$, D) *(b)*

Result: χ : keypoint subset returned

$K \leftarrow$ keypoint set detected

$U \leftarrow$ associated feature vectors

for $u_i \in U$ **do**
  $c_i \leftarrow$ KDE(u_i, U) *(a)*

end

for $k_i \in K$ **do**
  if $c_i < C_{th}$ **then**
    Add k_i to χ
  end

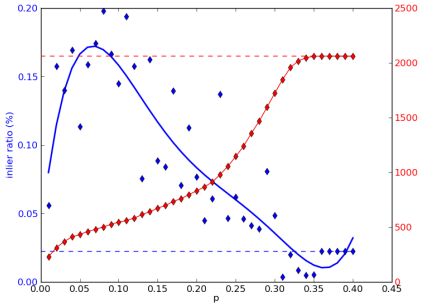
end

return χ

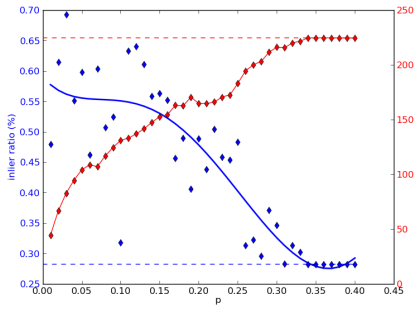
Algorithm 1: CORE algorithm.

Table 1: Comparisons of the results (percentage, number of good matches/total matches) for three different approaches: first column plain matching SIFT, second column SIFT with the 2NN filter ($d = 0.8$) and last column SIFT with both CORE ($p = 0.1$) and 2NN filter ($d = 0.8$).

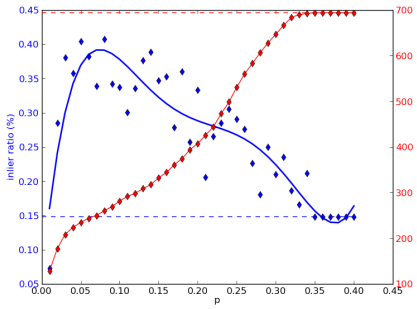
couple	unfiltered		2NN		CORE + 2NN	
object0014	23.89%	*322 / 1348*	70.68%	*258 / 365*	81.82%	*153 / 187*
object0008	20.00%	*336 / 1680*	52.71%	*204 / 387*	66.51%	*143 / 215*
object0039	26.78%	*448 / 1673*	66.24%	*310 / 468*	67.37%	*159 / 236*
object0110	24.58%	*222 / 903*	57.29%	*165 / 288*	69.34%	*95 / 137*
object0164	25.16%	*685 / 2723*	65.66%	*545 / 830*	71.88%	*317 / 441*
object0170	41.61%	*928 / 2230*	80.25%	*760 / 947*	87.83%	*469 / 534*
object0181	32.35%	*645 / 1994*	74.77%	*495 / 662*	81.69%	*290 / 355*
object0192	18.75%	*486 / 2592*	64.78%	*309 / 477*	73.93%	*241 / 326*
object0106	25.06%	*505 / 2015*	74.71%	*325 / 435*	77.42%	*216 / 279*
chess01	15.92%	*225 / 1413*	47.49%	*142 / 299*	84.48%	*49 / 58*
chess02	10.72%	*182 / 1698*	35.98%	*127 / 353*	86.44%	*51 / 59*



Plain RANSAC

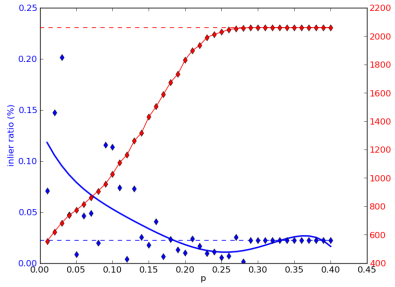


RANSAC with 2nn filtering

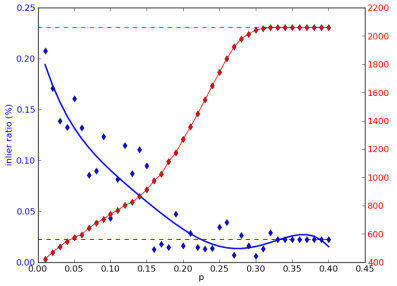


RANSAC with cross-check

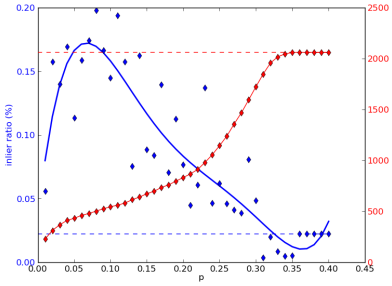
Figure 7: Evolution of inlier ratio (blue) when increasing p with $\mu = 0.30$, with SURF detector and ORB descriptor on chess images. Numbers of matches are shown in red, dashed lines are the respective values for non-filtering approach.



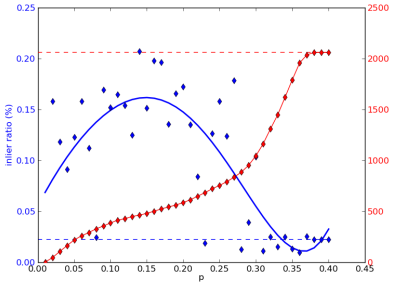
$\mu = 0.20$



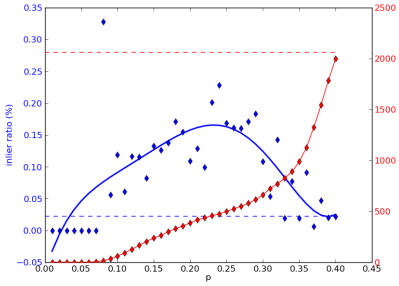
$\mu = 0.25$



$\mu = 0.30$



$\mu = 0.35$



$\mu = 0.40$

Figure 8: Evolution of inlier ratio when increasing the bit-switching probability, μ, with SURF detector and ORB descriptor on chess images.

Figure 9: Inlier ratio plots (blue) as a function of μ with SURF detector. Row labels indicate chosen descriptors, line labels are RANSAC approaches. Numbers of matches are shown in red, dashed lines are the respective values for non-filtering approach.

$p = 0.05$

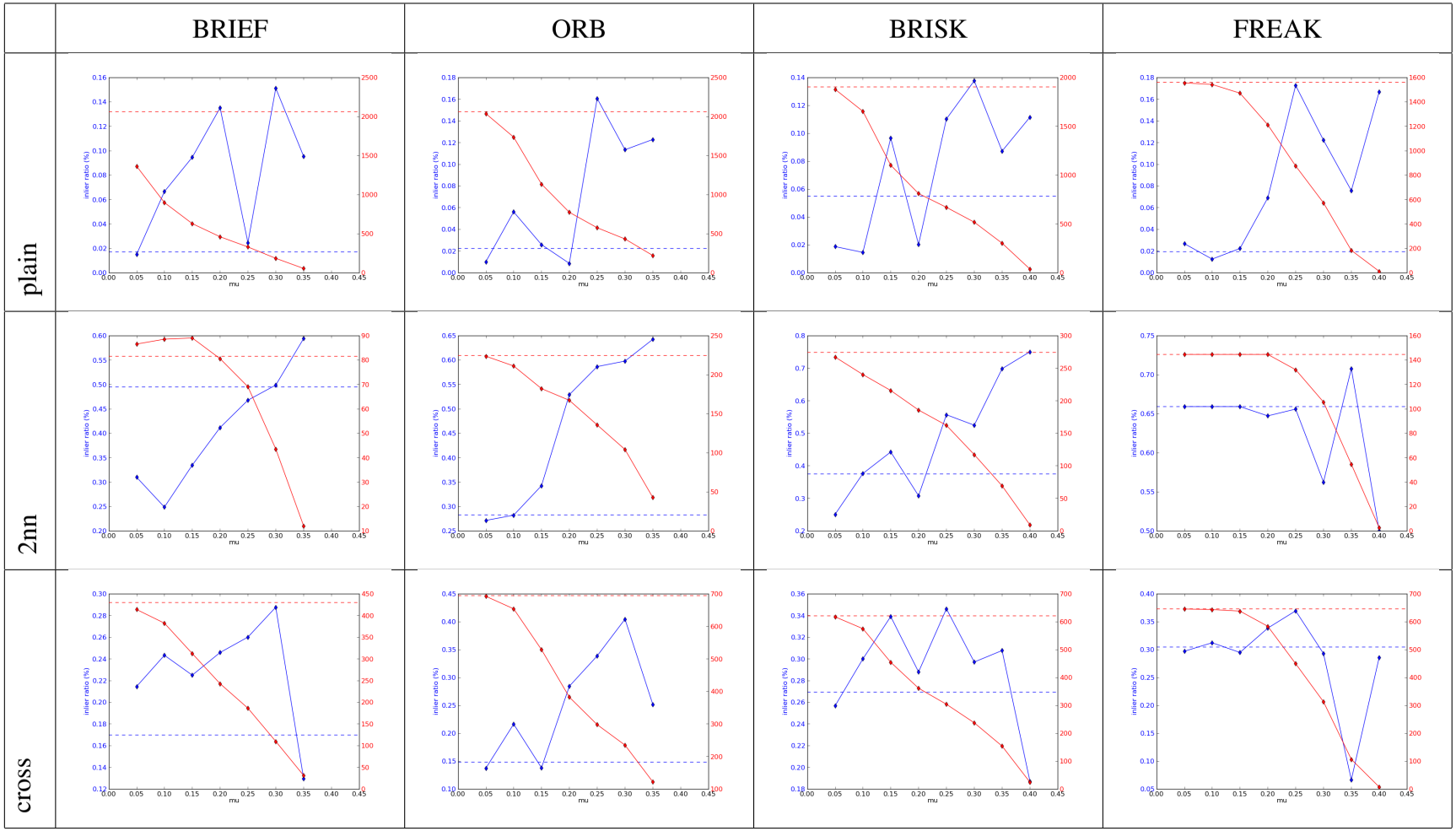
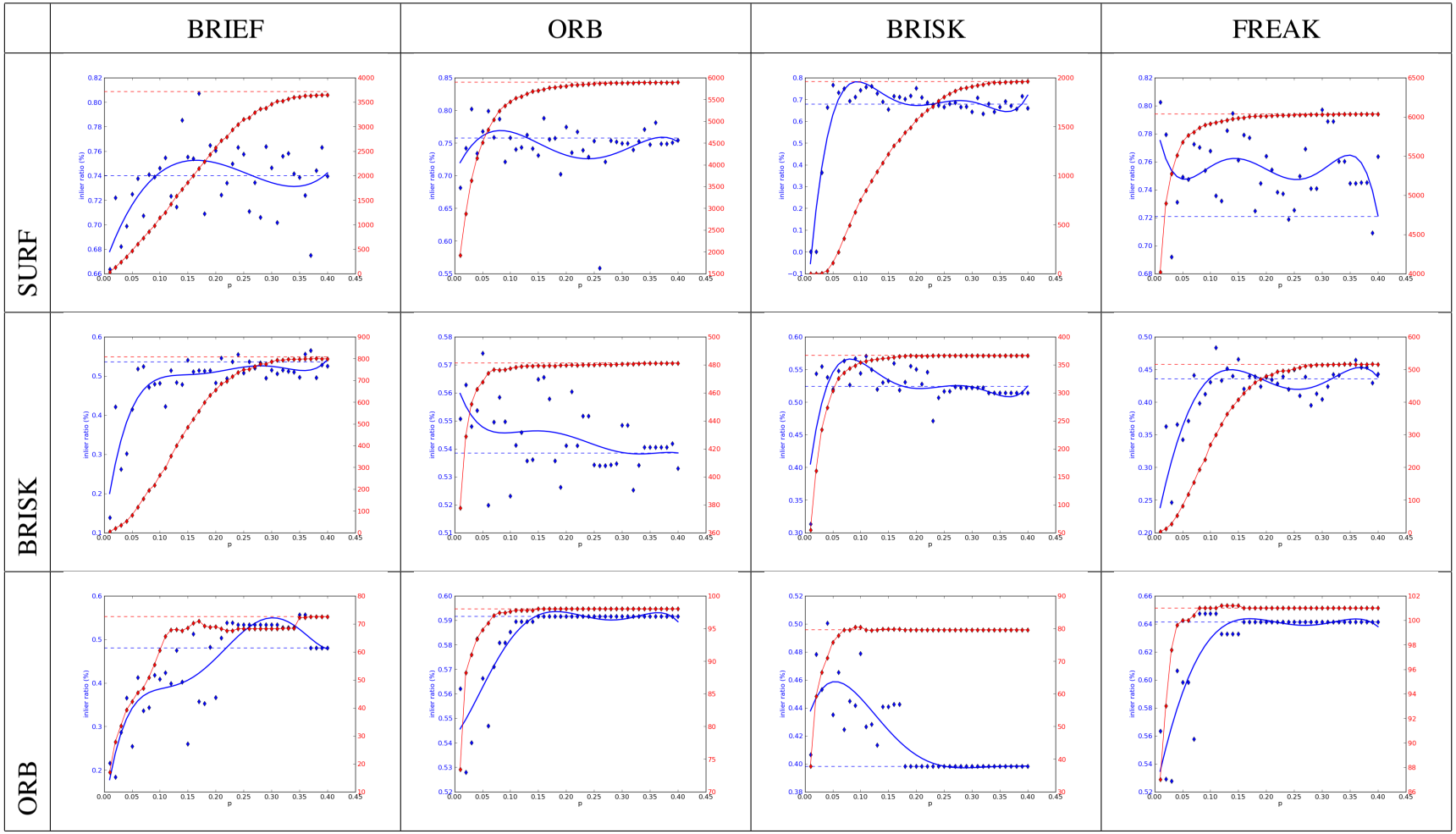


Figure 10: Inlier ratio plots (blue) as a function of p With 2NN filtering. Row labels indicate chosen descriptors, line labels are detectors. Number of matches are shown in red, dashed lines are the respective values for non-filtering approach.



3. Conclusions

We presented the CORE algorithm, a pre-processing filter which extracts from a feature vector set a smaller subset less subject to confusion by removing highly similar keypoints thanks to a probability approach. Results showed that subsets extracted are more discriminant and our approach can be combined with post-processing ones.

The algorithm can be applied on feature points and binary descriptors and it is better used on high-confusion context with lots of repetitive visual patterns.

ACKNOWLEDGMENTS

This work was financially supported by the French region Provence-Alpes-Côte d'Azur (PACA).

References

D. Lowe, Object recognition from local scale-invariant features, in: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Vol. 2, 1999, pp. 1150–1157 vol.2. doi:10.1109/ICCV.1999.790410.

K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (10) (2005) 1615–1630. doi:10.1109/TPAMI.2005.188.

C. Wu, SiftGPU: A GPU implementation of scale invariant feature transform (SIFT), http://cs.unc.edu/ ccwu/siftgpu (2007).

H. Bay, T. Tuytelaars, L. Gool, Surf: Speeded up robust features, in: Computer Vision - ECCV 2006, Vol. 3951 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 404–417.

J.-M. Morel, G. Yu, Asift: A new framework for fully affine invariant image comparison, SIAM J. Img. Sci. 2 (2) (2009) 438–469. doi:10.1137/080732730.
URL http://dx.doi.org/10.1137/080732730

M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: Computer Vision - ECCV 2010, Vol. 6314 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 778–792.

T. Ojala, M. Pietikainen, D. Harwood, Performance evaluation of texture measures with classification based on kullback discrimination of distributions, in: Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on, Vol. 1, 1994, pp. 582–585 vol.1. doi:10.1109/ICPR.1994.576366.

E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 2564–2571. doi:10.1109/ICCV.2011.6126544.
URL http://dx.doi.org/10.1109/ICCV.2011.6126544

S. Leutenegger, M. Chli, R. Y. Siegwart, Brisk: Binary robust invariant scalable keypoints, in: Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, IEEE Computer Society, Washington, DC, USA,

2011, pp. 2548–2555. doi:10.1109/ICCV.2011.6126542.
URL http://dx.doi.org/10.1109/ICCV.2011.6126542

R. Ortiz, Freak: Fast retina keypoint, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 510–517.
URL http://dl.acm.org/citation.cfm?id=2354409.2354903

T. Trzcinski, V. Lepetit, Efficient Discriminative Projections for Compact Binary Descriptors, in: European Conference on Computer Vision, 2012.

M. C. T. Trzcinski, V. Lepetit, Learning Image Descriptors with Boosting, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).

V. Balntas, L. Tang, K. Mikolajczyk, Bold - binary online learned descriptor for efficient image matching, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

G. Levi, T. Hassner, LATCH: learned arrangements of three patch codes, in: Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016.
URL http://www.openu.ac.il/home/hassner/projects/LATCH

E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision, Vol. 1, 2006, pp. 430–443.

C. Harris, M. Stephens, A combined corner and edge detector, in: In Proc. of Fourth Alvey Vision Conference, 1988, pp. 147–151.

R. Deriche, Z. Zhang, Q.-T. Luong, O. Faugeras, Robust recovery of the epipolar geometry for an uncalibrated stereo rig, in: Proceedings of the Third European Conference on Computer Vision (Vol. 1), ECCV '94, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994, pp. 567–576.
URL http://dl.acm.org/citation.cfm?id=189359.189599

P. H. S. Torr, D. W. Murray, Outlier detection and motion segmentation, 1995, pp. 432–443.

J. Zhao, J. Ma, J. Tian, J. Ma, D. Zhang, A robust method for vector field learning with application to mismatch removing, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 2977–2984. doi:10.1109/CVPR.2011.5995336.

J. Ma, J. Zhao, J. Tian, A. L. Yuille, Z. Tu, Robust point matching via vector field consensus, IEEE Transactions on Image Processing 23 (4) (2014) 1706–1721. doi:10.1109/TIP.2014.2307478.
URL http://dx.doi.org/10.1109/TIP.2014.2307478

M. A. Fischler, R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395. doi:10.1145/358669.358692.
URL http://doi.acm.org/10.1145/358669.358692

P. H. S. Torr, A. Zisserman, Mlesac: A new robust estimator with application to estimating image geometry, Comput. Vis. Image Underst. 78 (1) (2000) 138–156. doi:10.1006/cviu.1999.0832.
URL http://dx.doi.org/10.1006/cviu.1999.0832

T. Sattler, B. Leibe, L. Kobbelt, Scramsac: Improving ransac's efficiency with a spatial consistency filter., in: ICCV, IEEE, 2009, pp. 2090–2097.

S. Pang, J. Xue, Q. Tian, N. Zheng, Exploiting local linear geometric structure for identifying correct matches, Computer Vision and Image Understanding 128 (0) (2014) 51 – 64. doi:http://dx.doi.org/10.1016/j.cviu.2014.06.006.
URL http://www.sciencedirect.com/science/article/pii/S1077314214001337

X. Li, Z. Hu, Rejecting mismatches by correspondence function, Int. J. Comput. Vision 89 (1) (2010) 1–17.

G. Loy, J.-O. Eklundh, Detecting symmetry and symmetric constellations of features, in: Proceedings of the 9th European Conference on Computer Vision - Volume Part II, ECCV'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 508–521.

S. Lee, R. Collins, Y. Liu, Rotation symmetry group detection via frequency analysis of frieze-expansions, in: Proceedings of CVPR 2008, 2008.

Y. Liu, R. Collins, Y. Tsin, A computational model for periodic pattern perception based on frieze and wallpaper groups, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (3) (2004) 354–371. doi:10.1109/TPAMI.2004.1262332.

N. Jiang, P. Tan, L. F. Cheong, Multi-view repetitive structure detection, in: ICCV, IEEE, 2011, pp. 535–542.

M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, L. J. Guibas, Discovering structural regularity in 3d geometry, in: ACM SIGGRAPH 2008 Papers, SIGGRAPH '08, ACM, New York, NY, USA, 2008, pp. 43:1–43:11. doi:10.1145/1399504.1360642.
URL http://doi.acm.org/10.1145/1399504.1360642

E. N. Mortensen, H. Deng, L. Shapiro, A sift descriptor with global context, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 184–190. doi:10.1109/CVPR.2005.45.
URL http://dx.doi.org/10.1109/CVPR.2005.45

S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522. doi:10.1109/34.993558.
URL http://dx.doi.org/10.1109/34.993558

S. J. Mok, K. Jung, D. W. Ko, S. H. Lee, B.-U. Choi, Serp: Surf enhancer for repeated pattern, in: Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part II, ISVC'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 578–587.
URL http://dl.acm.org/citation.cfm?id=2045195.2045259

P. Martins, P. Carvalho, C. Gatta, Context aware keypoint extraction for robust image representation, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2012, pp. 100.1–100.12. doi:http://dx.doi.org/10.5244/C.26.100.

E. Parzen, On estimation of a probability density function and mode, The Annals of Mathematical Statistics 33 (3) (1962) pp. 1065–1076.
URL http://www.jstor.org/stable/2237880

J. Rabin, Y. Gousseau, J. Delon, A contrario matching of local descriptors (2007).

J. Rabin, J. Delon, Y. Gousseau, A statistical approach to the matching of local features, SIAM Journal on Imaging Sciences (2009) 958.

J. G. Skellam, The frequency distribution of the difference between two poisson variates belonging to different populations, J. Royal Statist. Soc. 109 (1946) 296.