Noname manuscript No.
(will be inserted by the editor)

Evaluation Method, Dataset size or Dataset Content:
How to Evaluate Algorithms for Image Matching?

Nadia Kanwal · Erkan Bostanci · Adrian F. Clark

Received: date / Accepted: date

Abstract Most vision papers have to include some evaluation work in order to demonstrate that the algorithm proposed is an improvement on existing ones. Generally, these evaluation results are presented in tabular or graphical forms. Neither of these is ideal because there is no indication as to whether any performance differences are statistically significant. Moreover, the size and nature of the dataset used for evaluation will obviously have a bearing on the results, and neither of these factors are usually discussed. This paper evaluates the effectiveness of commonly-used performance characterization metrics for image feature detection and description for matching problems and explores the use use of statistical tests such as McNemar's test and ANOVA as better alternatives.

Keywords Performance characterization · Feature matching · Homography

1 Introduction

Vision research has developed a substantial number of algorithms for tasks such as image matching, segmentation, stitching, tracking and navigation. One should not expect all of these algorithms to be equally accurate and reliable, yet deciding which one is best-suited to a particular problem can be difficult. However, statisticians have done a great service to vision researchers and

N. Kanwal
Lahore College for Women University, Lahore, Pakistan
E-mail: nadia.kanwal@lcwu.edu.pk

E. Bostanci
Computer Engineering Department, Ankara University, Turkey
E-mail: ebostanci@ankara.edu.tr

A. F. Clark
School of Computer Science & Electronic Engineering, University of Essex, Colchester, UK
E-mail: alien@essex.ac.uk

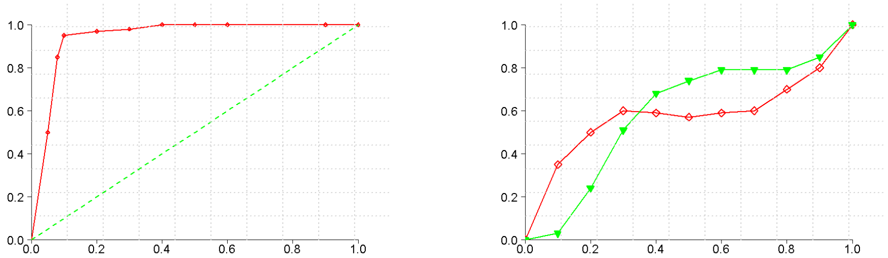
developers by defining a number of performance characterization measures. Novertheless, the selection of an appropriate performance measure is again crucial and demands an in-depth understanding of the domain (such as computer vision here) and data (images here) used for performance evaluation.

There are a number of performance measures already in widespread use. The best-known is arguably the ROC (Reciever Operating Characteristic) curve [1], a visual form of performance comparison in which the curve on a plot is used to represent an algorithm's performance, as illustrated in Figure 1. However, there are other visual performance metrics. These include precision–recall [2] and sensitivity–specificity [3] graphs, which were introduced because calculating the accuracy of algorithms alone was found to be misleading [4] in machine learning applications. In the computer vision domain, it has become the norm to present a comparison of a newly-proposed algorithm with results from existing methods using an appropriate performance evaluation measure. These metrics characterize different aspects of algorithms' performances. As a consequence, they can produce different rank orderings of algorithms, as seen in for example [5–7]. This paper reviews the strengths and weaknesses of existing performance evaluation measures specifically for assessing feature detection and description algorithms to match pairs of images, including ROC and precision–recall curves, F-measure *etc.*, and explores the use of statistical tests such as McNemar's test and ANOVA as more principled alternatives.

Performance evaluation studies in other domains regularly use statistical hypothesis tests, such as the χ^2 test, t-test, McNemar's test and ANOVA [8–12], though few of them focus on the quantity of data needed and their variability. Ensuring the dataset is large enough and exhibits enough variation is as important for vision research as any other mathematical or statistical discipline if the results are to be generalized. Even sophisticated and statistically-reliable evaluation techniques may produce misleading results if the sample size is not sufficiently large.

In vision research, algorithms are generally tested on a number of images, though the amount of image data employed is rarely large in the statistical sense. This study examines the behaviour of algorithms for an image matching problem using different numbers of images with different content. This is done by dividing a large database into many small subsets and ascertaining whether they produce similar rankings of algorithms as on the whole database; as the image content is the same in all images, one might expect the results to be consistent.

To avoid confusion, the term 'database' in this paper refers to a collection of images which is intended for evaluation purposes, while 'dataset' is one component of a database, typically images of the same scene. This paper uses two widely-used databases of different numbers of images to ascertain whether the performance differences calculated for small numbers of images reflect the general trends of algorithms. Although the findings are strictly applicable only to the problem of image matching, the processes employed in obtaining them can be applied to many problems in the computer vision domain.



(a) A typical ROC curve. The full line indicates the performance of an algorithm as a tuning parameter is varied, while the dotted line indicates what would be expected from an algorithm choosing outcomes randomly

(b) When ROC curves cross, one needs to choose which algorithm to use carefully

Fig. 1: Receiver Operating Characteristic (ROC) curves

The remainder of this paper is organized as follows. Section 2 introduces the way that vision evaluation studies are generally performed and describes measures that commonly are used to measure or represent performance. Section 3 introduces the null hypothesis testing framework that can be employed to assess performance differences of vision algorithms. Two statistical tests, McNemar's test and ANOVA, are also introduced in this section. Section 4 presents McNemar's test as an alternate to other performance metrics. Section 5 introduces the homography testing framework used to analyse the performances of feature extraction algorithms (also known as feature operators) using McNemar's test and ANOVA. Section 6 describes the databases of images employed and an evaluation of the performances of feature operators on them. Section 7 goes on to explore the interplay between image content and dataset size. Finally, Section 8 concludes the discussion by presenting some rules of thumb about selecting an appropriate dataset size and a proper evaluation framework for obtaining statistically-valid performance comparisons of multiple algorithms.

2 Performance Evaluation Measures: Reliability and Statistical Significance

Vision algorithms are generally assessed according to whether they have succeeded or failed, a true or false (T/F) result, on a series of test images. (A separate set of training images is normally available for problems that involve manual tuning or machine learning.)

In the absence of well-defined benchmarks, the performance of an algorithm is usually compared with those of other algorithms and outcomes are counted in the form of a confusion matrix.

		Actual Outcome	
		T	*F*
Predicted Outcome	*T*	True Positive *TP*	False Positive *FP*
	F	False Negative *FN*	True Negative *TN*

There are several metrics in widespread use that characterize the performance of algorithms using these values, including the *true positive rate* (TPR), *false positive rate* (FPR), *accuracy*, *precision*, *recall*, *sensitivity* and *specificity*. Table 1 gives definitions of these metrics. These metrics are usually shown graphically as in ROC curves, precision–recall curves *etc.*

An ROC curve shows the performance of an algorithm by plotting TPR versus FPR, as illustrated in Figure 1a. It is important to appreciate what is plotted here: each point on the curve summarizes the performance of an algorithm with a specific set of tuning parameters; hence, the curve shows how an algorithm's performance varies as a tuning parameter changes. An algorithm whose performance is close to the top-left corner of an ROC curve is performing better than one whose curve lies further away. In practice, ROC curves often cross as illustrated in Figure 1b; then one has to be careful about the settings of the tuning parameters of algorithms. In an attempt to identify an overall better algorithm when ROC curves cross, several researchers calculate the *area under the curve* (AUC). However, this is not necessarily reliable [13, 14]. These concerns are also applicable to the other performance measures discussed below.

Precision–recall curves (recall on the y-axis against precision on the x-axis) are somewhat analogous to ROC curves, though the top right corner indicates good performance. (Many researchers plot recall against $1-$precision to have a similar orientation as ROC curves [7].) Ideally, precision $\approx$ recall ≈ 1 represents good performance; however, recall can be easily maximized at the expense of precision and *vice versa*. Hence, for ranking an algorithm, one often combines precision and recall into the so-called *F-measure* (see Table 1).

Similarly, sensitivity–specificity graphs are most commonly used in behavioural sciences and are closely related to ROC curves [15]. Their appearance can be similar to an ROC curve if sensitivity is plotted against $1-$ specificity [16]. It is also interesting to see an algorithm's performance using simple measures such as *false positive ratio* (FP_r) against *true positive ratio* (TP_r). Lastly, accuracy is probably the most popular method for translating confusion matrix data into a single numeric performance measure.

Table 1 shows how the confusion matrix values are affected as an algorithm is tuned. The algorithm in question is concerned with matching an image using feature correspondences, though that is not important in this example; rather,

Table 1: Quantitative measures for performance assessment and how they are may vary as an algorithm is tuned. Set A represents original output of an algorithm, Set B the result of parameter tuning from down-sampling of negative examples, and Set C the result of tuning parameters that resulted in a uniform increase in all examples. The values of individual performance metrics are based on the output of this notional algorithm to show if these are invariant to the distribution of values in the confusion matrix.

	Set	A	B	C
	TP	3361	3361	3371
	TN	2370	198	2380
	FP	1294	101	1304
	FN	375	375	385
Performance metric	Description	Results		
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	0.7744	0.8820	0.7730
Precision	$\frac{TP}{TP+FP}$	0.7220	0.9708	0.7210
Recall	$\frac{TP}{TP+FN}$	0.8996	0.8996	0.8974
TPR	$\frac{TP}{TP+FP}$	0.7220	0.9708	0.7210
FPR	$\frac{FP}{TN+FP}$	0.3532	0.3378	0.3540
F-measure	$2\frac{Precision \times Recall}{Precision+Recall}$	0.8011	0.9339	0.7997
Sensitivity	$\frac{TP}{TP+FN}$	0.8996	0.8996	0.8975
Specificity	$\frac{TN}{TN+FP}$	0.6468	0.6622	0.6460
TP_r	$\frac{TP}{N}$	0.4541	0.8330	0.4531
FP_r	$\frac{FP}{N}$	0.1749	0.0250	0.1752

it is how the confusion matrix values change as the algorithm's tuning parameter is changed. Similar effects can be observed while almost any algorithm's tuning parameters are varied. Changing the value of a tuning parameter may, for example, convert a false positive into a true positive, affecting two cells of the confusion matrix. [17] considered these kinds of events in detail, identifying for example that if the data in the confusion matrix change proportionally, the ROC curve is unaffected. Table 1 shows the confusion matrix values from the algorithm with different tuning parameter settings A, B and C and the corresponding derived measures. Set C in Table 1 has 40 more points than Set A, 10 in each of the four categories. Because there are different counts in each of TP *etc*, the actual performance is different yet the TPR is unchanged. This is clearly undesirable. Similarly, recall is invariant to this type of change, as are both sensitivity and specificity. Hence, these curves have some shortcomings for assessing the performances of vision algorithms.

With so many performance metrics available, it is illuminating to discover whether they all yield consistent results. If they do not, one could argue that they are actually measuring something other than performance, or are measuring different aspects of performance. To that end, an experiment was performed using a pair of images drawn from the well-established Oxford Graffiti database[1], hereafter referred to as the 'small' database. The algorithm being assessed identifies characteristic features in images in the database and then attempts to match them, so that specific points in one image are known to relate to specific points in another. The need to perform this type of matching is an important enabling technology for computer vision applications such as panorama stitching, depth-from-stereo, tracking, segmentation, object identification, and so on. More precisely, a local image feature detection and description algorithm (SIFT [18] in this case) is used to identify interest points in both images independently, then these interest points are matched using descriptors of the interest points [19]. The correct and false matches are categorized into four possible outcomes based on the descriptors' similarity, namely:

- TP: obtained when an algorithm's outcome is a correct match;
- FP: obtained when an algorithm reports a result but that result is an incorrect match;
- TN: obtained when an algorithm reports a failure when a point in the first image is not matched because there is no corresponding point in the second image;
- FN: obtained when an algorithm reports a failure when the corresponding point exists in the second image but was not matched.

Results were collected for three different matching thresholds, $\tau = 0, 0.7$ and 1: a threshold of zero yields no matched points; for a threshold of 0.7, all points will be matched for which the descriptor difference is less than 0.7; a threshold of 1 means that all points are matched. Table 2 presents TP *etc* for the three thresholds; this is equivalent to three confusion matrices. It also shows the corresponding measures calculated from the data for these thresholds. The results are also plotted in Figure 2 using the curves commonly encountered in the literature. The shaded area in each graph represents the region where the outcomes are consistent with good performance. Let us consider an algorithm to be good if, in each graph, all three points appear in the shaded region. According to this criterion, only two of the plots (the accuracy and specificity–sensitivity graphs) classify this algorithm as good. But does this reflect the algorithm's true performance?

The ambiguity in the accuracy graph is obvious because it says that algorithm has same accuracy of 0.4 when all of the outcomes are negatively ($\tau = 0$) or positively ($\tau = 1$) classified! Similarly, the ROC, precision–recall, TP_r–FP_r and F graphs rate this algorithm at their lowest positions of zero at $\tau = 0$, when all outcomes fall in negative classes, overlooking the fact that many of these negative results are *true* negatives. Although some measures do highlight poor performance, they do not do so consistently: for example, the high specificity scores suggests that the algorithm would be good at identifying negative outcomes correctly but ignores the large number of false negative outcomes for thresholds of 0 and 0.7. Similarly, the high true positive rate with a lower false positive rate in the ROC curve may rank this an algorithm as having good

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/

Table 2: Performance metrics calculated for three different matching thresholds

Threshold	**0.0**	**0.7**	**1.0**
TP	0	375	743
FP	0	114	1048
TN	727	640	0
FN	1064	662	0

Measure	**0.0**	**0.7**	**1.0**	**Measure**	**0.0**	**0.7**	**1.0**
FPR	0.0000	0.1512	1.0000	Specificity	1.0000	0.8488	0.0000
TPR	0.0000	0.7669	0.4149	Sensitivity	0.0000	0.3616	1.0000
Precision	0.0000	0.3616	1.0000	FP_r	0.0000	0.0637	0.5851
Recall	0.0000	0.7669	0.4149	TP_r	0.0000	0.2094	0.4149
Accuracy	0.4059	0.5667	0.4149	F-Measure	0.0000	0.4915	0.5864

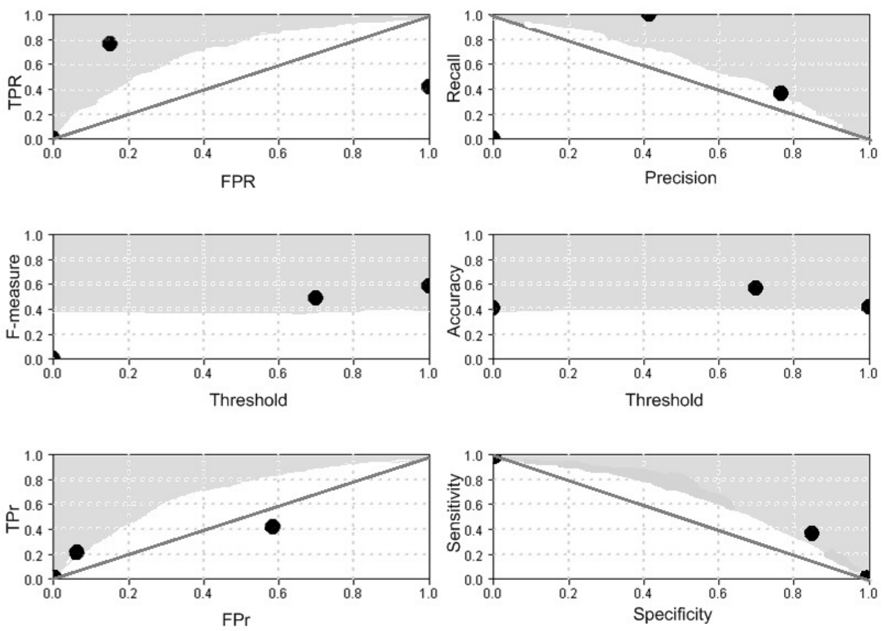
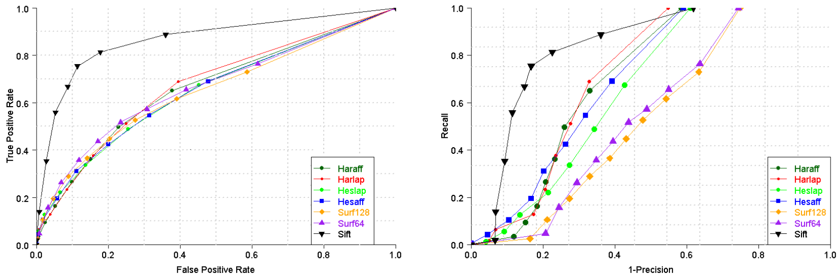


Fig. 2: Plots of performance evaluation measures. Diagonal lines on the first and last rows of graphs indicate the expected performance of a random algorithm. Performance above these lines correspond to good performance, indicated by a shaded area. Similarly, the middle row shows F-measure and accuracy graphs and the shaded area corresponds to good performance (> 40% here).

performance for threshold 0.7, overlooking the large number of false negatives. The same is the case with the precision–recall curve for thresholds of 0.7 and 1.0. Interestingly, with lower TP_r for thresholds 0.7 and 1.0, the TP_r–FP_r graph seems more representative of the algorithms' actual performances.

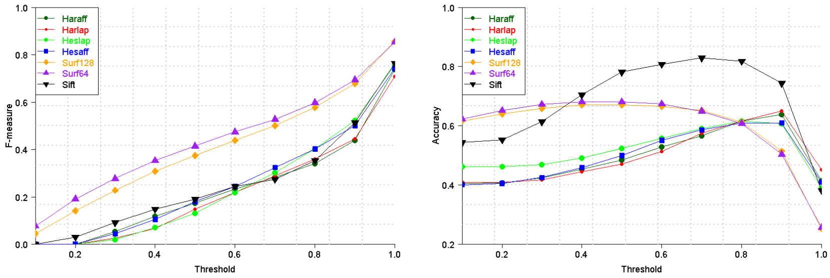
These results highlight some of the weakness of existing performance characterization methods for assessing an algorithm's performance. However, evaluating one algorithm's performance in isolation is usually not required, and hence these methods have been widely used in the literature to compare algorithms' performances and produce rankings.

2.1 Comparing the Performances of Several Algorithms



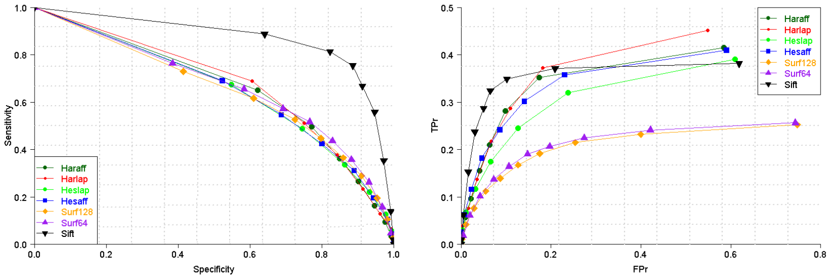
(a) (b)

Fig. 3: ROC and precision–recall plots for matching Graffiti images 1 and 2



(a) (b)

Fig. 4: F-measure and Accuracy plots for matching Graffiti images 1 and 2



(a) (b)

Fig. 5: Sensitivity-Specificity and TP_r-FP_r plots for matching Graffiti images 1 and 2

To understand the behaviour of performance evaluation measures for comparing several algorithms, a number of feature operators have been used to match features in the same image data. These are SIFT [18], SURF-64 [6], SURF-128 [6], Harris-Affine-GLOH (Haraff) [19], Harris-Laplace-GLOH (Harlap) [19], Hessian-Affine-GLOH (Hesaff) [19] and Hessian-Laplace-GLOH (Heslap) [19]. All seven feature operators were evaluated using for matching Graffiti images 1 and 2.

Regrettably, one cannot simply run the operators on the images and determine whether features were found in the right places, principally because the different feature operators detect different types of features — for example, Harris-based operators tend to identify corner features in an image while SIFT avoids them. Hence, the approach that has been taken is to use feature matches found between the two images to calculate the homography matrix, the transformation of one image onto the other [20]. Ground truth homography matrices are provided with the databases used here.

The simplest way to compare two homography matrices is to see how closely they project points. Therefore, a specific number of points from one image were projected using the estimated homography and then compared with the true projection of the point using the homography supplied with the database. This process was repeated for several nearest neighbour matching thresholds and the resultant graphs are shown in Figures 3, 4 and 5; the results presented here are essentially the same as those in Table 2 but include a greater number of matching threshold values.

As mentioned before, the hope is that these graphs would show some similarity in the ranking of algorithms based on their performance, determined by the relationships of the curves in the graphs. This is not the case: for example, there are obvious discrepancies in the results presented by the ROC and precision–recall curves. According to Figure 3b, both versions of SURF employed exhibit poor performance but in Figure 3a, all algorithms but SIFT have similar performance. Similarly, when both FN and TN become zero (at a threshold of unity), the ROC curve shows that all algorithms have same TPR and FPR and hence should be considered similar; but this is not the case for the precision–recall and TP_r–FP_r curves. Figure 4b shows a completely different behaviour, where the accuracies of SURF-64 and SURF-128 are significantly better than all other methods except SIFT. This may be happening because, by increasing the threshold, all negative outcomes shift to positive ones, and accuracy and ROC are invariant to this change (a consequence of the combination of the confusion matrix values they use). As mentioned earlier, the sensitivity–specificity graph in Figure 5a is closely related to the ROC curve and hence shares similar properties. Similarly, Figure 4a shows SURF-64 and SURF-128 being dominant as they have the highest F-measures, better than SIFT. Although the precision–recall and TP_r–FP_r ratio graphs mostly agree, the question asked earlier still remains valid: do they characterize performance well? Indeed, one might conclude that any algorithm can be presented as performing better than the others by intelligently selecting the most suitable performance measure.

Notwithstanding the above, the major drawback of all these graphical methods is that even if they carry any statistical significance, it is not shown. Even error bars do not necessarily indicate that performances are necessarily different in the statistical sense. In any case, these curves may well overlap each other, making it difficult to identify which algorithm is better overall.

Due to these biased performance characterizations of algorithms by different graphical evaluation methods, the authors contend that the research community should be looking for other reliable and statistically valid evaluation techniques. The remainder of the paper explores this.

3 Null Hypothesis Testing

An hypothesis is a way of describing a theory about data. In many cases, an hypothesis can be proven to be right or wrong using evidence. A methodology has been developed over the last few decades by the statistics research community that allows for evidence-based decisions to be made about performance. One starts with a so-called *null hypothesis* that (say) a newly developed algorithm is no better than an existing one. The formal definition of this null hypothesis will be

H_o: there is no difference in performance between the two algorithms

One can also propose an alternative hypothesis:

H_1: the two algorithms have different performances

By gathering the results of a trial employing the two algorithms on the same data, one can amass evidence as to which hypothesis is better. One assumes that the null hypothesis H_o is correct unless the evidence suggests that it cannot be; hence, null hypothesis testing is inherently conservative. A number

Table 3: Truth table for McNemar's test

	Algorithm A Failed	Algorithm A Succeeded
Algorithm B Failed	N_{ff}	N_{sf}
Algorithm B Succeeded	N_{fs}	N_{ss}

of tests can be used for null hypothesis testing, including the χ^2 test, the t-test, McNemar's test, ANOVA, and so on — the test that is selected is based on known properties of the data being used for evaluation. The resulting test statistics are compared with some critical value selected for an arbitrary level, α, which is often used as a cut-off between a statistically significant and a statistically insignificant result. A statistically significant result rejects the null hypothesis while a statistically insignificant result indicates that there is not enough evidence to reject the null hypothesis.

In this work, two statistical tests are used, McNemar's test and ANOVA. As we shall see, McNemar's test works by exploring where one treatment succeeded and the other failed, ensuring that well-understood binomial statistics apply; indeed, it is sometimes described as a form of the statistical sign test for categorical data. The test is non-parametric but statistically 'weak' in that it requires a larger amount of evidence to indicate dissimilarity of performance than other, statistically 'stronger' tests. The second test that will be used is ANOVA ("analysis of variance"), which is perhaps best thought of as a generalisation of the t-test to many variates. ANOVA is statistically stronger than McNemar's test but imposes some requirements on the data, principally that they are Normally distributed. To be able to employ ANOVA, one needs to ensure that these requirements are met. However, when ANOVA can be employed, less data are required for it to ascertain whether performance differences are significant.

3.1 McNemar's Test

McNemar's test has been widely used in medical research [8–12]; however, it has not been fully explored for comparing the performances of vision algorithms. Therefore, this study explores the use of McNemar's test in the null hypothesis framework to ascertain whether it produces more reliable rankings than the graphical methods rejected in the previous section.

McNemar's test is a non-parametric evaluation metric introduced by Quinn McNemar in 1947 [21]. To compare Algorithm A with Algorithm B, a null hypothesis can be formed by assuming that there is no statistical difference between their performances. One then assesses whether the evidence obtained from testing does or does not support that hypothesis. Given a dataset for which the ground truth is known, one applies both algorithms to each member of the dataset in turn, recording successes and failures in a kind of 'truth table' for this pair of algorithms as shown in Table 3. In the table, N_{sf} is the number of tests for which algorithm A succeeded and algorithm B failed, and so on.

When all the tests have been performed, the values of N_{sf} *etc* are used to calculate the so-called *Z-score* (or just Z):

$$Z = \frac{|N_{sf} - N_{fs}| - 1}{\sqrt{N_{sf} + N_{fs}}} \tag{1}$$

This expression takes into account the cases where one algorithm succeeds and the other fails and is also normalised by the number of these differences. This is in sharp contrast to many evaluations currently performed in computer vision, which largely focus on where algorithms succeed. One can see that Z has similarities to the popular χ^2 test.

If Algorithm A and Algorithm B give similar results, then $Z \approx 0$; as their results diverge, Z increases. To assess whether Z indicates a statistically significant result, one normally does so in the context of a particular level of significance. In computer vision, it is common to use a one-in-twenty level ($\alpha = 0.05$), which means that the particular results might be obtained purely by random fluctuations in the data one time in twenty. For the value of Z to be reasonably reliable, one needs $N_{sf} + N_{fs} \gtrsim 20$ to achieve this one-in-twenty criterion [22].

McNemar's test is suitable for use on pairs of algorithms only; when there are more than two algorithms being compared in a pair-wise manner, as here, then it is necessary to introduce a correction when interpreting the results; this matter is discussed further in section 4. However, the need to make these corrections makes ANOVA an attractive alternative when the data obey a Normal distribution.

3.2 ANOVA

ANOVA can be used to perform multiple (more than two) comparisons at the same time without increasing the Type-I error (false rejection of null hypothesis). It can also be used for null hypothesis testing, provided the data are Normally distributed data, for example as one would find in psychological research [23].

ANOVA assesses whether two or more groups exhibit a statistically-significant difference in their means (μ). The null hypothesis is

$$H_o : \mu_1 = \mu_2 = \mu_3 = ... = \mu_n \tag{2}$$

where n is the number of independent groups under comparison. ANOVA can show that there are at least two groups whose means differ significantly. There are different variants of ANOVA test, based on the number of factors that vary. So-called "one-way" ANOVA is used when one needs to compare data means grouped under a single category; two-way ANOVA is used to

Table 4: ANOVA test calculations for k groups and n data instances per group. SS is the Sum of Squares, df the number of degrees of freedom, MS the Mean Square and F the ratio of two mean square values

Source of Variation	*SS*	*df*	*MS*	*F*
Between Groups	SS_b	k-1	$MS_b = \frac{SS_b}{k-1}$	$\frac{MS_b}{MS_w}$
Within Groups	SS_w	n-k	$MS_w = \frac{SS_w}{n-k}$	

compare population means based on two factors or categories, and so on. Before applying ANOVA, there are some conditions for the data that need to be checked:

- groups must be independent;
- data in each group must be Normally distributed;
- there is homogeneity of variance.

The homogeneity of variance criterion means the variances of the groups under analysis should be similar, which can be ascertained using Hartley's F_{max} test [24]. This calculates the ratio of the maximum and minimum group variances, F_{max}, and if this ratio is less than a critical value (obtained from a table), the groups are assumed to have similar variances. However, if the groups' variances do not show homogeneity, then some mathematical treatment is required to prepare the data for ANOVA: this treatment can be for example calculating the natural logarithm of the data or taking its square root. (To avoid 0-based arithmetic errors, adding 1 prior to calculation is acceptable.) Moreover, the independence and normality of the data can be checked by calculating mean, median and mode of the data for each group. An equivalent mean, median and mode suggests that the data are Normally distributed, though a formal test can be used to confirm this.

ANOVA compares groups by calculating the mean square difference between and within groups as shown in Table 4, where SS is sum of square differences, given by

$$SS = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{3}$$

SS is divided by the number of degrees of freedom (df) both between and within groups. Commonly, the F-test is used in conjunction with the variance for comparing groups of total deviation using MS (see Table 4) between and within groups. F is compared with F_{crit}, whose value depends on the significance level chosen α and can be determined from tables [25]. If $F \geq F_{crit}$ or if the probability of error $P \leq \alpha$, then the null hypothesis should be rejected, showing at least two of the groups' means have statistically significant differences; however, the test does not indicate which mean or means differ, a limitation of the technique.

The most important distinction between McNemars test and ANOVA is that the former compares outcomes (success/failure) whereas the latter compares a continuous measure. One can convert a continuous measure into a

Table 5: Mapping of TP *etc* when comparing the results from an algorithms with the ground truth

	Ground truth Failed	Ground truth Succeeded
Algorithm Failed	0	FP+FN
Algorithm Succeeded	0	TP+TN

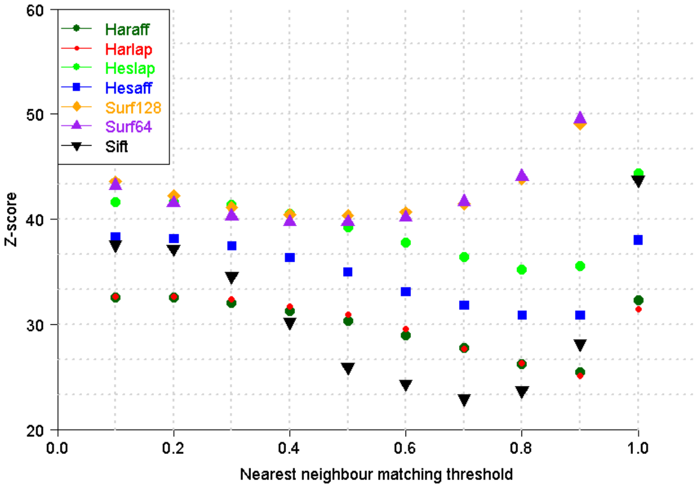


Fig. 6: McNemar's test for ROC-like analysis. Algorithms' performances are compared against the ground truth, so a Z-score closer to zero indicates better performance. Z-scores are shown for different threshold values in a similar way to ROC curves.

discrete outcome using a threshold but not *vice versa*. Unlike ANOVA, McNemars test can answer two questions: whether the two samples are statistically different; and which one of them is better — both with a given confidence level.

4 McNemar's Test for ROC-Like Analysis

Although McNemar's test is normally used to compare the performances of two algorithms, if one uses the correct values as one 'algorithm', the resulting Z can be regarded as a performance measure, with a smaller Z indicating better performance. Table 5 shows how the numbers of true and false positives contribute to the sums needed for calculating Z.

McNemar's test is for the analysis of paired data, and hence to compare more than two samples one needs to perform McNemar's test multiple times. According to statistical theory, multiple two-sample tests tend to increase the probability of Type-I errors. However, corrections can be applied to reduce Type-I errors, the simplest of which is the Bonferroni correction [26]. This is done by adjusting the significance level α; see [27] for a detailed discussion. As the number of algorithms under comparison in this study is seven ($A = 7$) so to achieve the overall 5% error rate alluded to above, we determine the per-test error rate α_c by dividing α by number of algorithms under test so that $\frac{\alpha}{A} = 0.00714$ and $1 - \alpha_c = 0.9928$. From tables, the corresponding $Z_{\text{crit}} = 2.5$. Hence, for these multiple comparisons, $Z < 2.5$ indicates that there is no significant performance difference between algorithms.

There are some concerns over the use of any correction [28]. For example, Bonferroni corrections control only the probability of false positives and come at the cost of increasing the probability of false negatives; it may therefore be considered as being too conservative to control the family-wise error rate [29]. There are some other corrections suggested in the literature, such as the Benjamini & Hochberg [30] and Benjamini & Yekutieli [31] corrections, which control the expected proportion of false discoveries amongst the rejected hypothesis, a less rigid condition than the Bonferroni correction. However, using different corrections can yield different results, with one correction causing the hypothesis to be rejected while another causes it to be accepted [32]; hence, applying the most conservative correction is considered safest approach. Hence, in this work the Bonferroni correction is used.

The Z-scores for the algorithms' outcomes when compared with the ground truth for each threshold are presented in Figure 6. A Z-score closer to zero shows that an algorithm performed more similarly to the ground truth and hence exhibits better performance. The Z-score, PR (Figure 3b) and TP_r–FP_r graphs (Figure 5b) show somewhat similar rankings *i.e.* with SIFT performing best, closely followed by Harlap. However, the TP_r–FP_r graph has more similarity with the Z-scores. It was observed earlier that no two performance measures agree with each other and that is a good reason for questioning their reliability. As the TP_r–FP_r graph gives somewhat similar results to Figure 6, one could say that there are two measures which are consistent and are therefore more reliable than the others; however, this is not a strong argument.

5 Homography Testing

The general approach to measuring the performances of algorithms by projecting points using a homography matrix calculated from feature matches was described in Section 2. The spacing of these points to be projected is important: if the points selected when calculating the homography matrix are not evenly distributed over the image, then the homography tends to represent the transformation of only that part of the image and therefore may not project all points correctly [33, 34]. Therefore, to test a homography matrix, equally-

Table 6: Z-scores calculated between SIFT and SURF-128 for different numbers of points selected for homography testing between an image pair is shown in the first column. A Z-score of 0 means the two algorithms exhibited similar performance.

Number of Points	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
Bark 1-2	3.75	3.75	3.75	4.25	4.25	4.25	4.25	4.25	4.25	4.25
Bark 1-3	0	0	0	0	4.8	4.8	4.8	4.8	4.8	4.8
Bark 1-4	0	0	0	1.15	1.5	1.5	1.5	1.5	1.5	1.5
Bark 1-5	0	0	0	0	0	0	4.59	11	11.53	11.53
Graffiti 1-2	0	0	0	0	0	0	6.56	11.96	15.59	16.19
Graffiti 1-3	8.89	8.89	8.89	8.89	11.66	15.36	18.33	20.88	23.07	23.07
Graffiti 1-4	9.9	14.07	17.26	19.95	18.85	18.85	15.37	15.37	15.37	15.37
Graffiti 1-5	0	0	0	0	0	0	0	0	0	0

spaced points are used to avoid any skew towards the points selected for the calculation of the homography.

5.1 How Many Points Are Required To Produce Consistent Results?

The objective here is to establish the number of points that need to be projected from one image to another using a calculated homography in order for it to produce consistent results. This is done by starting with a small number of regularly-spaced points for projection, then increasing the number of points. When enough points are projected, the statistical significance of the result will not change as further points are added; the results become consistent. If consistency were not reached then clearly the method would be inappropriate for assessing performance.

Table 6 presents results for the Graffiti and Bark datasets from the small database. Both of these datasets contain images with complex transformations: the Graffiti images have been taken from different viewpoints while the Bark images are zoomed and rotated. McNemar's test is applied to find the number of correct and incorrect projections for two algorithms, SIFT and SURF-128, and the Z-scores between the sets of projected points are presented in Table 6. Shaded cells in the table indicate the point at which a significant result ($Z > Z_{\text{crit}}$) is obtained, highlighting the number of points required to obtain consistent results for the particular image data. It can be seen that, for some images, 100 points are enough to establish the performance differences between operators, such as for Bark images 1 and 4, where the result remained insignificant even for 1000 points ($Z < 1.96$). Similarly, for Graffiti image pairs 1-3, 1-4 and 1-5, the evaluation shows a similar trend for different numbers of points. However, for the rest of the images, at least 700 points are required to obtain consistent results. From this, we conclude that for homography testing the number of points should be greater than 700.

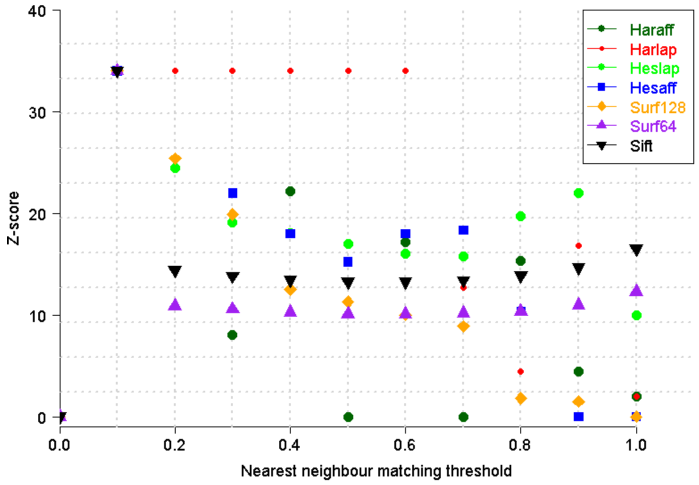


Fig. 7: Z-scores between algorithms and ground truth homographies for Graffiti images 1 and 2. A Z-score of 35 (added manually) denotes the cases where homography calculation was not possible due to there being fewer than 4 matched points.

Table 7: Z-scores between feature operatorsfea for matching Graffiti images 1 and 2. To find the better algorithm, follow the arrowhead direction in each pairwise comparison.

	Haraff	**Harlap**	**Hesaff**	**Heslap**	**SURF-64**	**SURF-128**	**Score**
SIFT	↑ 8.89	← 8.89	← 12.92	← 15.97	← 5.40	← 2.70	5
Haraff		← 12.65	← 15.75	← 18.33	← 8.60	← 5.00	6
Harlap			← 9.27	← 13.19	← 6.70	0.50	3
Hesaff				← 9.27	1.10	← 9.10	3
Heslap					0	↑ 7.20	0
SURF-64						↑ 6.90	1
SURF-128						-	2

5.2 Homography Testing Using McNemar's Test

As described earlier, the list of matched points obtained from each algorithm is used to calculate an estimated homography (H_e), which is then compared with the ground truth homography (H_{gt}). For McNemar's test, some 1000 equally-spaced points were selected from a reference image to be projected onto a test image using both homography matrices. If the Euclidean distance between two projections of a point is less than some threshold (5 in this case), then it is a success; otherwise, it is a failure.

Table 5 presents the Z-scores plotted in Figure 7. There is a limitation of this test: if the number of matched points between two images is less than 4

Table 8: Single factor ANOVA test summary

Groups	***Count***	***Sum***	***Mean***	***Variance***
Haraff	1000	576.24	0.576	0.001
Harlap	1000	1005.32	1.005	0.107
Hesaff	1000	1093.45	1.093	0.161
Heslap	1000	1099.35	1.099	0.402
SIFT	1000	903.79	0.904	0.123
SURF-64	1000	883.40	0.883	0.072
SURF-128	1000	1109.96	1.110	0.412
Source of Variation		***SS***	***df***	***MS***
Between Groups		217.74	6	36.29
Within Groups		1276.86	6993	0.18
F	**P**			F_{crit}
198.75	1.1×10^{-234}			**2.10**

then homography matrix estimation is not tractable. These cases can be seen in the table with Z-scores of 35. However, this limitation does not affect the overall test results because if there are less than 4 matched points, the algorithm's performance cannot be predicted anyway. To accept the null hypothesis, the Z-score should be less than the $Z_{\text{crit}} = 2.50$ for $\alpha = 0.007$ (two-tailed test); however, all scores are significantly higher than this, making it safe to reject the null hypothesis. For an algorithm to have better performance than others, it needs to show low Z-scores for different thresholds.

It is interesting to see that the Z-scores given in Figure 7 for a threshold of 0.7 vary consistently with the Z-scores between algorithms given in Table 7. In Figure 7, Haraff shows highest score of being better than all other algorithms. The distance between Harlap and SIFT in Figure 7 shows that their difference in the performance is significant, with SIFT showing better performance (lower Z-score). This result is supported by $Z = 8.89$ given in the pairwise comparison. It is evident that if we have ground truth available then comparing algorithms' performances with ground truth using McNemar's test can give a figure of merit and therefore pairwise comparison is not required. However, in the absence of ground truth data, pairwise comparison gives a reliable and statistically-significant ranking. This is not possible using any graphical evaluation method.

5.3 Homography Testing using ANOVA

To compare the performances of the feature operators under study using ANOVA, the same Graffiti images have been used using the same general approach as with McNemar's test. As previously discussed, some 1000 equally spaced points are projected using both ground truth and estimated homography matrices. If a point P_i is projected using both homography matrices, then $P_e = H_e \times P_i$ and $P_t = H_{gt} \times P_i$ are the projections of that point using the estimated and ground truth homography matrices respectively. Let $d = |P_t - P_e|$ be the difference in their projected positions; this will be close to

Table 9: Small database of images

Dataset	**Transformation**	**Number of images**
Bikes	blur	6
Trees		6
Graffiti	viewpoint change	6
Wall		6
Bark	zoom + rotation	6
Boat		6
UBC	JPEG compression	6
Leuven	illumination	6

zero if both homography matrices represent similar transformation and large otherwise. Hence, d is used to calculate sum of square difference for ANOVA:

$$SS = \sum_{i=1}^{n} (d_i - \bar{d})^2 \tag{4}$$

Before applying ANOVA, the data are checked for basic homogeneity of variances. The distances of false matches make the data variance too high and non-homogeneous, so some data treatment is required. Here, the square roots of data have been used to make variances homogeneous, after which ANOVA is applied and the result shown in Table 8.

The result based on $F = 198.75 \gg F_{crit} = 2.10$ and $P = 1.1 \times 10^{-234} \ll \alpha = 0.05$ suggests that the null hypothesis should be rejected and shows statistically significant differences in the performances of interest point operators, in agreement with the results obtained using McNemar's test. The difference between the two tests is that the former used a distance threshold to determine success and failure, but the latter used a numerical measure.

The next section explores discrepancies in evaluation results as a consequence of database content, by comparing the performances on one well-established database with another of similar size. The effects of database size are also explored by employing subsets of a larger database.

6 The Effect of Dataset Size

For evaluating the performance of image feature matching, the most widely used database of images, the small database alluded to above, was introduced in [19]. This database[2] comprises several datasets of real images with different geometric and photometric transformations; it was made publicly-available and many subsequent studies have also used it [5–7,35–54]. This work employs it too; it is summarized in Table 9.

To assess whether the amount of data affects the relative performances of interest operators, this work also employs a second database[3]. This was

[2] `http://www.robots.ox.ac.uk/~vgg/research/affine/`

[3] `http://www.featurespace.org/`

Table 10: Large database of images

Dataset	Transformation	Number of images
Asterix	zoom	16
BIP		8
Crolle		7
East-Park	zoom + rotation	10
East-South		9
Ensimag		10
Laptop		21
Resid		10
Laptop_rs	rotation	13
Mars		18
Monet		18
NewYork		35
VanGogh		16

collected by the same researchers as devised the small database, perhaps implicitly indicating that they believe it is really too small. This larger database contains 191 images in 13 datasets (Table 10). Different datasets encompass geometric and photometric transformations that include zoom, rotation, viewpoint change, blurring, change in illumination and JPEG compression. All images in each dataset are planar scenes or taken with a fixed camera position, so each image pair is related by a homography matrix which is supplied along with the imagery as 'ground truth.'

6.1 How Many Image Pairs?

Having established that more than 700 points need to projected between a pair of images to obtain a consistent result from a test, we are now able to ask how many image pairs are required to produce consistent results. The same general approach as in previous section is adopted, *i.e.* starting with a small number of image pairs and increasing the number; again the aim is to find a point at which results become consistent (either remaining statistically significant or insignificant). This establishes the minimum number of images that is required in a dataset. Again, this is done using McNemar's test for two algorithms, SIFT and SURF-128.

Five images from the each of the New York, Laptop, Mars and Asterix datasets of the large database were selected as these contain the largest number of images (35, 21, 18 and 16 respectively). A subset of 5 image pairs is used as starting point because this is the size of the datasets in the small database. If the number of images do not affect the performance results, then the Z-scores of small subsets should be similar to the result from the whole dataset. However, the results presented in Table 11 show that if the performance difference between two algorithms is insignificant for 5 pairs of images (set 4 of New York dataset in the Table 11), it subsequently becomes signifi-

Table 11: Z-scores calculated between SIFT and SURF-128 to identify the minimum number of image pairs required for performance evaluation. Z-scores less than $Z_{\text{crit}} = 1.96$ are considered insignificant. Z-scores with an asterisk denotes the case where SURF-128 outperformed SIFT; in all other cases SIFT outperformed SURF-128.

New York dataset (35 images)

Number of Image Pairs	5	10	15	17	34
Set 1	12.62	7.00	7.52	4.92	18.84
Set 2	*2.89	3.69	15.83	21.42	
Set 3	3.41	18.29			
Set 4	1.67				
Set 5	12.17				

Laptop dataset (21 images)

Number of Image Pairs	5	10	15	20
Set 1	0.00	0.00	11.40	11.40
Set 2	0.00	11.40		
Set 3	0.00			
Set 4	11.36			

Monet dataset (18 images)

Number of Image Pairs	5	10	15	17
Set 1	26.27	26.53	36.91	40.45
Set 2	3.47	30.50		
Set 3	25.61			

Asterix dataset (16 images)

Number of Image Pairs	5	10	15
Set 1	3.34	1.66	7.48
Set 2	0.99	6.89	
Set 3	12.90		

cant when the number of image pairs is increased. Similarly, set 2 of the same dataset shows that SURF-128 is significantly better than SIFT; however, this result does not occur when the number of image pairs were increased in the other sets. Similarly, there is no performance difference between SIFT and SURF-128 for three sets of the Laptop dataset with 5 image pairs, but this changes for the sets containing 15 or more image pairs. The same evaluation differences can be seen in the Monet and Asterix datasets. From these results, a rule of thumb can tentatively be established that at least 15 image pairs are required to obtain consistent performance evaluation results.

6.2 Results Using The Small Database

Results for all eight datasets of the small database are presented in Table 12, grouped according to the image transformations involved. An easy inspection method is to follow the scores generated for each algorithm in the last column for each dataset. None of these algorithms appears to be best for matching images with all kind of transformations: if an algorithm is good at matching images with rotation it fails to match images with scale difference and so on. Of

Table 12: Z-scores of feature operators for the small image database. To find the better algorithm, follow the arrowhead direction in a pairwise comparison.

Operators	SURF-64	SURF-128	Haraff	Harlap	Hesaff	Heslap	Score
Blurring (Bikes + Trees)							
SIFT	↑ 3.31	← 10.25	← 40.32	← 28.87	← 38.55	← 9.46	5
SURF-64		↑ 3.99	← 41.30	← 30.51	← 35.98	← 13.84	5
SURF-128			← 41.72	← 33.75	← 43.18	← 15.04	5
Haraff				↑ 13.50	1.09	↑ 33.86	0
Harlap					← 10.84	↑ 18.84	2
Hesaff						↑ 32.64	0
Heslap							3
Viewpoint change (Wall +Graffiti)							
SIFT	↑ 4.20	0.79	↑ 26.98	← 5.08	↑ 14.58	← 18.58	2
SURF-64		↑ 2.83	↑ 19.78	← 9.32	↑ 10.61	← 22.66	3
SURF-128			↑ 19.11	2.39	↑ 12.48	← 19.84	2
Haraff				← 17.20	← 9.50	← 29.69	6
Harlap					↑ 3.47	← 12.53	1
Hesaff						← 26.67	5
Heslap							0
Zoom+Rotation (Bark + Boat)							
SIFT	← 29.88	← 30.13	← 22.08	1.65	← 31.10	← 23.85	5
SURF-64		← 4.07	↑ 6.18	↑ 24.02	1.63	↑ 6.05	1
SURF-128			↑ 15.00	↑ 28.33	2.23	↑ 13.29	0
Haraff				↑ 22.63	← 7.35	0.45	3
Harlap					← 26.56	← 24.56	5
Hesaff						↑ 8.18	0
Heslap							3
JPEG Compression (UBC)							
SIFT	← 14.73	↑ 9.22	← 8.74	↑ 8.72	← 9.33	← 2.47	4
SURF-64		↑ 18.65	↑ 4.67	↑ 17.18	↑ 11.31	↑ 14.46	0
SURF-128			← 14.10	← 4.73	← 12.90	← 9.59	6
Haraff				↑ 14.11	1.96	↑ 8.00	1
Harlap					← 12.85	← 9.17	5
Hesaff						↑ 8.89	1
Heslap							3
Change in Illumination (Leuven)							
SIFT	← 5.62	← 8.05	↑ 11.27	← 32.36	← 19.60	← 16.87	5
SURF-64		← 3.69	↑ 14.38	← 31.97	← 17.76	← 9.36	4
SURF-128			↑ 15.25	← 31.26	← 16.85	← 7.11	3
Haraff				← 36.15	← 25.01	← 22.60	6
Harlap					↑ 21.80	↑ 23.82	0
Hesaff						↑ 8.29	1
Heslap							2

Table 13: Rankings of feature operators, the sum of arrowheads pointing to an operator. The table is sorted by total score.

Operators	Blur	View Point Change	Zoom + Rotation	JPEG compression	Change in illumination	Overall Score
SIFT	5	2	5	4	5	21
SURF-128	5	2	0	6	3	16
Haraff	0	6	3	1	6	16
SURF-64	5	3	1	0	4	13
Harlap	2	1	5	5	0	13
Heslap	3	0	3	3	2	11
Hesaff	0	5	0	1	1	7

Table 14: Performance analysis using ANOVA for the small database of images

Source of Variation	Between Groups	Within Groups	Groups	Mean	Variance
Zoom + Rotation (Bark and Boat)			SIFT	1.50	1.14
Sum of Square (SS)	25973.8	3991675	SURF-64	2.40	13.62
Degree of Freedom (df)	6	69993	Haraff	2.49	14.74
Mean Square (MS)	4328.97	57.03	SURF-128	2.56	5.19
F	75.91		Heslap	2.62	23.15
P	6.8×10^{-95}		Harlap	2.87	48.50
F_{crit}	2.10		Hesaff	3.72	292.87
Blur (Bikes and Trees)			SURF-128	1.25	0.95
Sum of Square (SS)	4183.34	123537	SURF-64	1.28	1.04
Degree of Freedom (df)	6	69993	Heslap	1.42	1.59
Mean Square (MS)	697.22	1.76	SIFT	1.43	1.18
F	395.03		Harlap	1.64	1.89
P	0		Haraff	1.82	2.67
F_{crit}	2.10		Hesaff	1.92	3.03
Change in view point (Graffiti and wall)			Harlap	1.88	8.07
Sum of Square (SS)	118817	1826445	Hesaff	2.52	8.75
Degree of Freedom (df)	6	69994	Haraff	2.60	8.84
Mean Square (MS)	19802.8	26.09	SURF-64	3.40	17.12
F	758.89		Heslap	4.34	36.20
P	0		SURF-128	4.85	52.70
F_{crit}	2.10		SIFT	5.75	50.99
Change in illumination (Leuven)			Haraff	1.84	3.70
Sum of Square (SS)	49404	436862	SURF-64	1.92	3.48
Degree of Freedom (df)	6	35063	SURF-128	2.03	3.70
Mean Square (MS)	8234	12.46	SIFT	2.11	4.45
F	660.87		Hesaff	2.27	3.97
P	0		Heslap	2.28	4.24
F_{crit}	2.10		Harlap	5.44	63.67
JPEG compression (UBC)			Haraff	1.84	3.70
Sum of Square (SS)	49404	436862	SURF-64	1.92	3.48
Degree of Freedom (df)	6	35063	SURF-128	2.03	3.70
Mean Square (MS)	8234	12.46	SIFT	2.11	4.45
F	660.87		Hesaff	2.27	3.97
P	0		Heslap	2.28	4.24
F_{crit}	2.10		Harlap	5.44	63.67

course, we need to bear in mind that the number of image pairs — 10 for first three sets and 5 for the last two — are not sufficient to draw any statistically valid conclusion according to the rule of thumb established earlier. According to these results, SIFT is robust for matching all images except for viewpoint change. Similarly, Harris-affine with GLOH descriptor appears to be a strong combination of detector and descriptor for matching images under viewpoint change and change in illumination. SURF-128 showed, unexpectedly, to be the worst algorithms for matching zoomed and rotated images, a contradiction of [6].

The overall ranking is shown by the order of algorithms in Table 13, according to which SIFT, SURF-128 and Harris-affine show statistically better performance when compared with other algorithms. This can also be seen in the ANOVA analysis given in Table 14. Hesaff performed better only for one type of images, *i.e.* matching images with different viewpoints. Exact similarity in McNemar's test and ANOVA results is not observed, perhaps because of the mathematical treatment applied to the data prior to ANOVA to cajole them to be Normally distributed.

Table 15: The Z-scores of feature operators for three different transformations

Operators	**SURF-64**	**SURF-128**	**Haraff**	**Harlap**	**Hesaff**	**Heslap**	**Score**
Zoom (Asterix, BIP, Crolle)							
SIFT	←5.99	←3.42	←26.63	←25.961	←50.05	←48.61	6
SURF-64		↑5.23	←20.48	←20.25	←45.18	←45.03	4
SURF-128			←24.78	←22.59	←45.20	←42.56	5
Haraff				2.02	←35.24	←44.54	2
Harlap					←35.85	←42.13	2
Hesaff						←26.68	1
Heslap						-	0
Rotation (East Park, East South, Ensimag, Laptop, Resid)							
SIFT	←36.19	←46.09	←5.36	←8.15	←52.08	←55.38	6
SURF-64		←17.68	↑29.35	↑25.08	←28.22	←28.79	2
SURF-128			↑41.77	↑34.80	←18.95	←21.10	2
Haraff				←4.69	←52.87	←52.41	5
Harlap					←49.51	←47.73	4
Hesaff						←3.24	1
Heslap						-	0
Zoom + Rotation (Laptop_rs, Mars, Monet, New York, VanGogh)							
SIFT	←10.66	←11.68	←67.07	←58.86	←64.57	←88.36	6
SURF-64		↑7.99	←55.40	←47.89	←51.72	←78.24	4
SURF-128			←61.58	←53.12	←56.35	←81.95	5
Haraff				↑11.21	↑3.68	←38.77	1
Harlap					←5.58	←44.29	3
Hesaff						←40.28	2
Heslap						-	0

Table 16: Rankings of feature operators for the larger database. The table is sorted by the scores of algorithms, so the topmost one has the highest rank in the pool.

	Zoom	**Zoom + Rotation**	**Rotation**	**Overall Score**
SIFT	6	6	6	**18**
SURF-128	5	2	5	**12**
SURF-64	4	2	4	**10**
Harlap	2	4	3	**9**
Haraff	2	5	1	**8**
Hesaff	1	1	2	**4**
Heslap	0	0	0	**0**

6.3 Results Using The Large Database

These experiments allow us to ascertain whether the small database contains enough images to characterize the performances of algorithms: if differences are obtained using a larger database, we should be concerned that there are not enough. The larger database comprises 191 images in 13 datasets (Table 10), and it was used in exactly the same way as described in the previous section.

In order to demonstrate an algorithm's behaviour for a particular transformation between image pairs, a summary of these results is presented in

Table 17: ANOVA test results for datasets in the large database, grouped according to images with the same transformation

Zoom			**Mean based groups' ranking**		
Source of Variation	***Between Groups***	***Within Groups***	***Groups***	***Mean***	***Variance***
			SIFT	6.32	88.88
Sum of Square (SS)	1704064	9.67×10^8	SURF-64	10.10	461.51
Degree of Freedom (df)	6	209991	Haraff	12.22	474.49
Mean Square (MS)	284010.69	4605.29	Heslap	12.55	23715.91
F	61.67		SURF-128	12.58	1983.18
P	9.07×10^{-77}		Hesaff	14.78	795.56
F_{crit}	2.10		Harlap	15.53	4718.77
Rotation					
			SIFT	3.11	52.95
Sum of Square (SS)	198418.2	97346697	Haraff	3.27	63.67
Degree of Freedom (df)	6	671996	SURF-64	3.52	154.15
Mean Square (MS)	33069.70	144.86	SURF-128	3.70	147.36
F	228.28		Heslap	4.11	104.45
P	1.8×10^{-292}		Harlap	4.43	141.58
F_{crit}	2.10		Hesaff	4.66	349.89
Zoom + Rotation					
			SIFT	1.56	1.28
Sum of Square (SS)	120086	12789590	SURF-128	1.75	1.92
Degree of Freedom (df)	6	412993	SURF-64	1.82	4.22
Mean Square (MS)	20014.34	30.97	Harlap	2.40	13.36
F	646.29		Haraff	2.42	92.74
P	0		Hesaff	2.93	73.47
F_{crit}	2.10		Heslap	3.04	29.78

Table 15. This table identifies performance differences more clearly than ROC or precision–recall curves and has the advantage of associating a statistical confidence with each comparison. The Z-scores and directions of the arrows show that SIFT's detector and descriptor are effective in identifying stable features under geometric transformations. The performance of SURF-128 closely follows that of SIFT but there is a significant difference between their performances, evident by Z-scores such as 3.42, 11.68 and 46.09 for zoomed images, zoomed-and-rotated images, and images with only rotation respectively.

Table 16 gives a summarized characterization of the performances of all algorithms by collecting their scores from Table 15. A comparison of the rankings produced for the large and small databases shows broadly similar characterization, because only SURF-64 and Haraff operators have changed their positions in the table. Of course, one needs to keep in mind the difference in the image transformation in both databases, which can be a critique to the comparison of these results as being unfair.

To see a comparison between similar transformations, Table 18 presents a ranking of algorithms for those image datasets featuring both zoom and rotation. Again, the order of algorithms highlights the relative performance on different amount of data. Unfortunately none of the algorithms share similar positions in the tables. Interestingly, Heslap with GLOH is performing better when there are smaller numbers of image pairs, while Hesaff secured last position; this is not the case for the larger dataset. Having this level of dissimilarity in results suggests that the size of the database used has a significant effect

Table 18: Ranking of algorithms based on sample size (number of images with zoom and rotation) from Table 13 and 16

Ranking based on 10 image pairs	**Score**	**Ranking based on 59 image pairs**	**Score**
SIFT	5	**SIFT**	6
Harlap	5	**Haraff**	5
Haraff	3	**Harlap**	4
Heslap	3	**SURF-64**	2
SURF-64	1	**SURF-128**	2
SURF-128	0	**Hesaff**	1
Hesaff	0	**Heslap**	0

on the ranking — and this means that existing evaluations based around the small database need to be treated with some caution.

To confirm these results are not an artefact of the use of McNemar's test, the same general procedure has also been carried out with ANOVA, and the mean performances of these operators can be compared by comparing the last block of Table 17 and first block of Table 14. Both show different ranking of algorithms for the larger and smaller databases of images with same transformation zoom+rotation. For the large database SURF-128 is second best but for the small database its performance degrades and it is ranked fourth.

7 The Effect of Dataset Content

The use of different sample sizes for performance analysis revealed statistically significant performance differences of algorithms. However, it does not show whether these results will be different if the images are changed; in other words, does image content play a role in favour of any operator? To explore this effect, the larger datasets of images are divided into smaller subsets of fifteen image pairs each and McNemar's test and ANOVA have both been used to study the behaviour of the feature operators.

One need to keep in mind that a different amount of transformation has applied to each image in a dataset; the images in the Mars, Monet and New York datasets are rotated at different angles compared to the first image which is used to match with them. All of these operators are sensitive to these geometric transformations and may perform differently for different amounts of transformation. Theoretically speaking, a sufficiently large sample size should overcome this problem and one should be able to observe the general behaviour of algorithms.

Let us examine the results generated for different subsets of the four datasets Laptop, Mars, Monet and New York. All of these sets have more

Table 19: Performance comparison of SIFT and SURF with other operators for subsets of large dataset, where each subset contains 15 image pairs. The result of each subset can be compared with the whole dataset result given at the bottom of each set in bold.

Subsets	Operator	SURF-64	SURF-128	Haraff	Harlap	Hesaff	Heslap
laptop		0	0	← 12.962	← 3.75	← 5.66	← 22.98
laptop	**SIFT**	← 11.18	← 11.40	← 59.254	← 46.658	← 19.053	← 62.282
laptop		← **11.18**	← **11.40**	← **59.25**	← **46.66**	← **19.08**	← **62.28**
Mars		0	0	0	0	0	← 2.67
Mars	**SIFT**	0	0	0	0	0	← 1.789
Mars		**0**	**0**	**0**	**0**	**0**	← **2.67**
Monet		← 13.47	← 36.91	1.15	← 27.695	← 61.68	← 55.60
Monet	**SIFT**	← 18.71	← 37.336	1.15	← 27.166	← 58.489	← 59.523
Monet		← **20.27**	← **40.45**	**1.154**	← **27.69**	← **61.75**	← **60.93**
New york		← 13.526	← 7.5202	↑ 13.55	↑ 22.875	↑ 9.0387	↑ 11.172
New york		0.8157	← 7.5884	↑ 10.484	↑ 14.404	↑ 7.603	↑ 10.929
New york	**SIFT**	← 13.599	← 22.101	← 3.7275	1.072	← 10.291	← 16.614
New york		← **16.88**	← **18.83**	↑ **12.30**	↑ **22.87**	↑ **5.59**	**0.017**
laptop			0	← 13.417	← 5.0709	← 5.4801	← 21.52
laptop	**SURF-64**		0.47	← 56.903	← 44.812	← 11.386	← 60.202
laptop			**0.468**	← **56.90**	← **44.81**	← **11.49**	← **60.20**
Mars			0	0	0	0	1.79
Mars	**SURF-64**		0	0	0	0	0.71
Mars			**0**	**0**	**0**	**0**	**1.789**
Monet			← 33.211	↑ 12.123	← 17.124	← 60.22	← 53.359
Monet	**SURF-64**		← 29.519	↑ 18.6	← 9.5549	← 52.601	← 53.371
Monet			← **31.43**	↑ **20.19**	← **7.76**	← **54.77**	← **54.18**
New york			↑ 9.79	↑ 22.92	↑ 29.50	↑ 19.13	↑ 20.62
New york			← 10.332	↑ 5.9465	↑ 12.491	↑ 6.7651	↑ 8.7899
New york	**SURF-64**		← 11.77	↑ 9.19	↑ 12.61	2.17	← 4.55
New york			**1.36**	↑ **24.23**	↑ **33.61**	↑ **18.07**	↑ **12.17**
laptop				← 10.536	← 4.7246	← 4.5873	← 22.383
laptop	**SURF-128**			← 57.318	← 45.35	← 10.90	← 60.856
laptop				← **57.31**	← **45.35**	← **10.94**	← **59.19**
Mars				0	0	0	← 1.79
Mars	**SURF-128**			0	0	0	0
Mars				**0**	**0**	**0**	← **1.78**
Monet				↑ 36.824	↑ 15.531	← 47.605	← 31.688
Monet	**SURF-128**			↑ 36.715	↑ 16.671	← 40.042	← 38.258
Monet				↑ **40.01**	↑ **20.70**	← **41.59**	← **36.02**
New york				↑ 20.193	↑ 23.563	↑ 12.863	↑ 15.231
New york				↑ 15.602	↑ 18.242	↑ 14.504	↑ 13.707
New york	**SURF-128**			↑ 19.015	↑ 18.776	↑ 8.9711	0.23
New york				↑ **29.78**	↑ **33.30**	↑ **19.67**	↑ **11.67**

than 15 images and allow subsets of 15 image pairs to be selected. The performances of all operators are compared for these datasets. McNemar's test results are presented in Tables 19 and 20 and show Z-scores between pairs of feature operators for each subset and for the whole dataset (at the bottom of each set in highlighted and bold).

The results for the subsets from Laptop, Mars and Monet are consistent with the whole dataset, showing that the appropriate evaluation framework with sufficient dataset size can predict the behaviour of the algorithms. However, for the New York dataset, the better performing operator changes for different subsets. This appears to be principally because of the varying amounts of transformation alluded to above: the first subset contains images with less rotation, while the last subset has images with rotation up to 360°. For these kind of data, the subset size needs to be large to accommodate the maximum variation in transformation.

ANOVA results for whole New York dataset, given in Table 21, mostly agree with subsets results shown in Table 22 in showing statistically significant

Table 20: Performance comparison of Harris- and Hessian-based operators for subsets of large dataset, where each subset contains 15 image pairs. The result of each subset can be compared with the whole dataset result given at the bottom of each set in bold.

Subsets	Operator	Harlap	Hesaff	Heslap
laptop		↑ 9.3991	↑ 8.9745	← 14.425
laptop	**Haraff**	↑ 25.017	↑ 53.802	← 7.9418
laptop		**↑ 25.02**	**↑ 53.75**	**← 7.94**
Mars		0	0	1.1547
Mars	**Haraff**	0	0	0.7071
Mars		**0**	**0**	**1.15**
Monet		← 26.42	← 62.081	← 55.426
Monet	**Haraff**	← 26.42	← 62.177	← 60.133
Monet		**← 26.42**	**← 62.17**	**← 60.75**
New york		↑ 12.556	← 2.6052	0.4138
New york		↑ 7.9206	1.1107	↑ 3.8999
New york	**Haraff**	↑ 4.9357	← 8.9805	← 13.802
New york		**↑ 14.81**	**← 4.67**	**← 8.92**
laptop			0.8571	← 19.372
laptop	**Harlap**		↑ 39.068	← 32.098
laptop			**↑ 39.03**	**← 32.09**
Mars			0	← 3.6148
Mars	**Harlap**		0	← 2.6667
Mars			**0**	**← 3.61**
Monet			← 54.483	← 43.723
Monet	**Harlap**		← 50.938	← 48.516
Monet			**← 54.50**	**← 49.90**
New york			← 11.514	← 8.3461
New york			← 4.0682	0.596
New york	**Harlap**		← 10.99	← 15.37
New york			**← 14.42**	**← 17.93**
laptop				← 22.38
laptop	**Hesaff**			← 59.425
laptop				**← 59.40**
Mars				1.5
Mars	**Hesaff**			0
Mars				**1.5**
Monet				↑ 14.051
Monet	**Hesaff**			↑ 4.1503
Monet				**↑ 2.99**
New york				0.9843
New york	**Hesaff**			↑ 4.6988
New york				← 4.2632
New york				**← 5.39**

Table 21: ANOVA test for New York dataset as a whole

Groups	*Mean*	*Variance*
SURF-64	2.53	2.28
Harlap	2.56	2.57
SURF-128	2.60	2.43
Hesaff	2.61	2.51
Haraff	2.63	2.56
Heslap	2.64	2.81
SIFT	2.64	2.61
Source of Variation	*Between Groups*	*Within Groups*
Sum of Squares (SS)	474.40	791252.1
Degree of Freedom (df)	6	311619
Mean Square (MS)	79.07	2.54
F	31.14	
P	1.23×10^{-37}	
F_{crit}	2.10	

Table 22: ANOVA test for subsets of the New York dataset, each subset containing fifteen image pairs

New York Dataset	**Subset (Image1-16)**		**Subset Image13-27)**			**Subset (Image21-35)**		
Groups	*Mean*	*Variance*	*Groups*	*Mean*	*Variance*	*Groups*	*Mean*	*Variance*
Harlap	2.34	2.16	**SURF-64**	2.78	2.63	**SURF-64**	2.40	1.96
SURF-64	2.41	2.13	**Heslap**	2.79	3.02	**Harlap**	2.42	2.09
Haraff	2.44	2.26	**Hesaff**	2.83	2.85	**SURF-128**	2.46	2.02
Hesaff	2.44	2.24	**SURF-128**	2.85	2.74	**Haraff**	2.51	2.25
SIFT	2.48	2.33	**Harlap**	2.91	3.27	**SIFT**	2.52	2.36
Heslap	2.48	2.68	**SIFT**	2.92	3.01	**Hesaff**	2.56	2.36
SURF-128	2.48	2.41	**Haraff**	2.93	3.03	**Heslap**	2.64	2.69
Source of Variation	*Between Groups*	*Within Groups*		*Between Groups*	*Within Groups*		*Between Groups*	*Within Groups*
Sum of Squares (SS)	256.17	243110.6		360.13	308371.8		617.53	228337.8
Degree of Freedom (df)	6	104993		6	104993		6	101619
Mean Square (MS)	42.69	2.32		60.02	2.94		102.92	2.25
F	18.44			20.44			45.80	
P	1.54×10^{-21}			4.75×10^{-24}			2.41×10^{-56}	
F_{crit}	2.099			2.099			2.0989	

performance differences of feature operators ($F >> F_{\text{crit}}$) but do not yield similar rankings based on low means and variances. The major problem with ANOVA is that the data are required to be Normally distributed and the variances homogeneous; the data under analysis do not obey these rules and, even though they have been transformed by calculating its square root — the most effective of the standard transformations for these data — they do not fit a Normal distribution well. This transformation makes these rankings unreliable and so the rankings produced by McNemar's test are considered more trustworthy.

The underlying concept of the this study was to highlight the importance of the amount of data for comparison and an appropriate testing framework. Furthermore, the performance metrics discussed and critically analysed here are not specific to image matching problem but can be and have been used for comparing several other image processing algorithms. These include image stitching, tracking, navigation, augmented reality, visual SLAM and many more. Therefore, suggested evaluation framework and data centric rules can be easily applied to other domains where the analysis is done using a number of features, images, frames of a video, recognition of a number of objects or classification tasks *etc.* in order to provide a comprehensive benchmark.

8 Conclusions

Performance characterization is a sensitive problem and therefore needs to be dealt with carefully. Graphical methods for evaluation appear to be unreliable and sometimes misleading. The use of statistically reliable methods is more rigorous. To explore this, McNemar's test has been used to carry out a statistically-valid examination of the performances of vision algorithms. It is also important to note that McNemar's test can be used to carry out tasks such as ranking algorithms based on their performance, in which case comparison between algorithms and ground truth has been established as being sufficient. As McNemar's test alone can be criticised because it involves assigning an arbitrary threshold for distinguish success from failure, a companion study using ANOVA has been carried out to see whether a similar characterization is obtained. Although the testing procedures for the tests are slightly different, the results are broadly similar.

Unlike previous studies, this research takes account of the size of the dataset used in making comparisons. Contrary to previous evaluation studies [19,55], in which overlapping precision–recall curves made it difficult to determine which algorithms outperformed others, the results presented here are not only statistically reliable but also clearly indicative of differences in the performances of algorithms for the same set of data. Table 18 reflects the changes in ranking when the evaluations were performed on datasets with different sample sizes. Therefore, one needs to be very careful in drawing conclusions when the amount of data is not sufficiently large.

The paper has attempted to establish some valuable rules of thumb regarding data size when evaluating the performances of vision algorithms. The homography testing framework proposed and used for the evaluation of feature operators should use a minimum of 700 points. Similarly, it has been established that 5 images pairs are not sufficient; at least 15 image pairs are needed for statistically-valid results to be obtained.

Using these rules, a number of feature operators have been characterized based on their performances and the results are compared with the standard dataset of 5 to 10 image pairs widely used for this purpose. The results show that the SIFT detector and descriptor are more distinctive and robust for matching under different image transformations and give consistent performance regardless of the type of images. Conversely, the Harris-based detector combined with the GLOH descriptor (which is an extended form of SIFT descriptor) gives good performance only when there is a significant viewpoint change in images. It should therefore be most useful when used in conjunction with another reliable feature descriptor, such as SIFT or SURF.

References

1. Lee B Lusted. Signal detectability and medical decision-making. *Science*, 171(3977):1217–1219, 1971.
2. Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.
3. Robert S Galen and S Raymond Gambino. *Beyond normality: the predictive value and efficiency of medical diagnoses*. Wiley New York, 1975.
4. Foster J Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *ICML*, volume 98, pages 445–453, 1998.
5. Nico Cornelis and Luc Van Gool. Fast scale invariant feature detection and matching on programmable graphics hardware. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.
6. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.
7. T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.
8. N. Uemura, S. Okamoto, S. Yamamoto, N. Matsumura, S. Yamaguchi, M. Yamakido, K. Taniyama, N. Sasaki, and R.J. Schlemper. Helicobacter pylori infection and the development of gastric cancer. *New England Journal of Medicine*, 345(11):784, 2001.
9. R. Frothingham. Rates of torsades de pointes associated with ciprofloxacin, ofloxacin, levofloxacin, gatifloxacin, and moxifloxacin. *Pharmacotherapy*, 21(12):1468–1472, 2001.
10. Valerie L Durkalski, Yuko Y Palesch, Stuart R Lipsitz, and Philip F Rust. Analysis of clustered matched-pair data. *Statistics in medicine*, 22(15):2417–2428, 2003.
11. Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601. AUAI Press, 2004.
12. Mithat Gönen, Katherine S Panageas, and Steven M Larson. Statistical issues in analysis of diagnostic imaging experiments with multiple observations per patient1. *Radiology*, 221(3):763–767, 2001.
13. Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2):145–151, 2008.
14. Blaise Hanczar, Jianping Hua, Chao Sima, John Weinstein, Michael Bittner, and Edward R Dougherty. Small-sample precision of roc-related estimates. *Bioinformatics*, 26(6):822–830, 2010.
15. Nadia Kanwal. *Motion Tracking in Video using the Best Feature Extraction Technique*. Grin Publishing, Germany, 2009.
16. Karthikeyan Sakthivel and Rengarajan Narayanan. An automated detection of glaucoma using histogram features. *International journal of ophthalmology*, 8(1):194, 2015.
17. Lutz Hamel. Model assessment with roc curves. *The Encyclopedia of Data Warehousing and Mining, Idea Group Publishers,*, 2008.
18. D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
19. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1615–1630, 2005.
20. Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.
21. Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.
22. Sitanshu Sekhar Kar and Archana Ramalingam. Is 30 the magic number? issues in sample size estimation. *National Journal of Community Medicine*, 4(1), 2013.
23. Michael W Vasey and Julian F Thayer. The continuing problem of false positives in repeated measures anova in psychophysiology: A multivariate solution. *Psychophysiology*, 24(4):479–486, 1987.

24. Herman O Hartley. The use of range in analysis of variance. *Biometrika*, 37(3/4):271–280, 1950.
25. E Peaeson and H Haetlet. Biometrika tables for statisticians. *Biometrika Trust*, 1976.
26. H. Abdi. *Bonferroni and Šidák corrections for multiple comparisons.* Sage, Thousand Oaks, CA, 2007.
27. Adrian F Clark and Christine Clark. Performance characterization in computer vision a tutorial, 1999.
28. Thomas V. Perneger. What's wrong with bonferroni adjustments. *British Medical Journal*, 316:1236–1238, 1998.
29. Yosef Hochberg and Yoav Benjamini. More powerful procedures for multiple significance testing. *Statistics in medicine*, 9(7):811–818, 1990.
30. Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.
31. Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.
32. Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
33. Shoaib Ehsan, Nadia Kanwal, Adrian F Clark, and Klaus D McDonald-Maier. Measuring the coverage of interest point detectors. In *Image Analysis and Recognition*, pages 253–261. Springer, 2011.
34. E. Bostanci, N. Kanwal, and A.F. Clark. Spatial statistics of image features for performance comparison. *Image Processing, IEEE Transactions on*, 23(1):153–162, Jan 2014.
35. Hongli Deng, Eric N Mortensen, Linda Shapiro, and Thomas G Dietterich. Reinforcement matching using region context. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 11–11. IEEE, 2006.
36. C. Valgren and A. Lilienthal. SIFT, SURF and seasons: Long-term outdoor localization using local features. In *Proceedings of the European Conference on Mobile Robots (ECMR)*, pages 253–258. Citeseer, 2007.
37. Dermot Kerr, Sonya Coleman, and Bryan Scotney. Fesid: Finite element scale invariant detector. In *Image Analysis and Processing–ICIAP 2009*, pages 72–81. Springer, 2009.
38. Ruan Lakemond, Clinton Fookes, and Sridha Sridharan. Affine adaptation of local image features using the hessian matrix. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 496–501. IEEE, 2009.
39. Anil K Jain, Jung E Lee, and Rong Jin. Graffiti-id: Matching and retrieval of graffiti images. In *Proceedings of the First ACM workshop on Multimedia in forensics*, pages 1–6. ACM, 2009.
40. Matthew Toews and W Wells. Sift-rank: Ordinal description for invariant feature correspondence. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 172–177. IEEE, 2009.
41. M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.
42. Xiaojie Guo and Xiaochun Cao. Triangle-constraint for finding more good features. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1393–1396. IEEE, 2010.
43. Sidnei Alves de Araújo and Hae Yong Kim. Color-ciratefi: A color-based rst-invariant template matching algorithm. In *IWSSIP-17th International Conference on Systems, Signals and Image Processing*, 2010.
44. NV Medathati and Jayanthi Sivaswamy. Local descriptor based on texture of projections. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 398–404. ACM, 2010.
45. Mesut Guney and Nafiz Arica. Maximally stable texture regions. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4549–4552. IEEE, 2010.
46. Petros Kapsalas and S Kollias. Affine morphological shape stable boundary regions (ssbr) for image representation. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3381–3384. IEEE, 2011.

47. Taha H Rassem and Bee Ee Khoo. New color image histogram-based detectors. In *Visual Informatics: Sustaining Research and Innovations*, pages 151–163. Springer, 2011.
48. Arathi Issac, C Shunmuga Velayutham, et al. Saddlesurf: A saddle based interest point detector. In *Mathematical Modelling and Scientific Computation*, pages 413–420. Springer, 2012.
49. P Martins, C Gatta, and P Carvalho. Feature-driven maximally stable extremal regions. In *VISAPP (1)*, pages 490–497, 2012.
50. Jingneng Liu and Guihua Zeng. Description of interest regions with oriented local self-similarity. *Optics Communications*, 285(10):2549–2557, 2012.
51. Xiang Yang Wang, Pan Pan Niu, Hong Ying Yang, and Li Li Chen. Affine invariant image watermarking using intensity probability density-based harris laplace detector. *Journal of Visual Communication and Image Representation*, 23(6):892–907, 2012.
52. Felix von Hundelshausen and Rahul Sukthankar. D-nets: Beyond patch-based image descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2941–2948. IEEE, 2012.
53. Muhua Zhang, Yuxiong Zhang, and Jianrong Wang. Eliminating false matches using geometric context. In *Contemporary Research on E-business Technology and Strategy*, pages 325–334. Springer, 2012.
54. Qidan Zhu, Xue Liu, Chengtao Cai, and Qingchen Liu. Image local invariant feature description fusing multiple information. In *Fifth International Conference on Machine Vision (ICMV 12)*, pages 87830E–87830E. International Society for Optics and Photonics, 2013.
55. Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.