Evaluation of Features Detectors and Descriptors based on 3D objects

Pierre Moreels and Pietro Perona

California Institute of Technology, Pasadena CA91125, USA

Abstract

We explore the performance of a number of popular feature detectors and descriptors in matching 3D object features across viewpoints and lighting conditions. To this end we design a method, based on intersecting epipolar constraints, for providing ground truth correspondence automatically. These correspondences are based purely on geometric information, and do not rely on the choice of a specific feature appearance descriptor. We test detector-descriptor combinations on a database of 100 *objects viewed from* 144 *calibrated viewpoints under three different lighting conditions. We find that the combination of Hessian-affine feature finder and SIFT features is most robust to viewpoint change. Harris-affine combined with SIFT and Hessian-affine combined with shape context descriptors were best respectively for lighting change and change in camera focal length. We also find that no detector-descriptor combination performs well with viewpoint changes of more than 25-30°.*

1 Introduction

Detecting and matching specific features across different images has been shown to be useful for a diverse set of visual tasks including stereoscopic vision [34, 19], vision-based simultaneous localization and mapping (SLAM) for autonomous vehicles [30, 18], mosaicking images [4] and recognizing objects [28, 18]. This operation typically involves three distinct steps. First a ‘feature detector’ identifies a set of image locations presenting rich visual information and whose spatial location is well defined. The

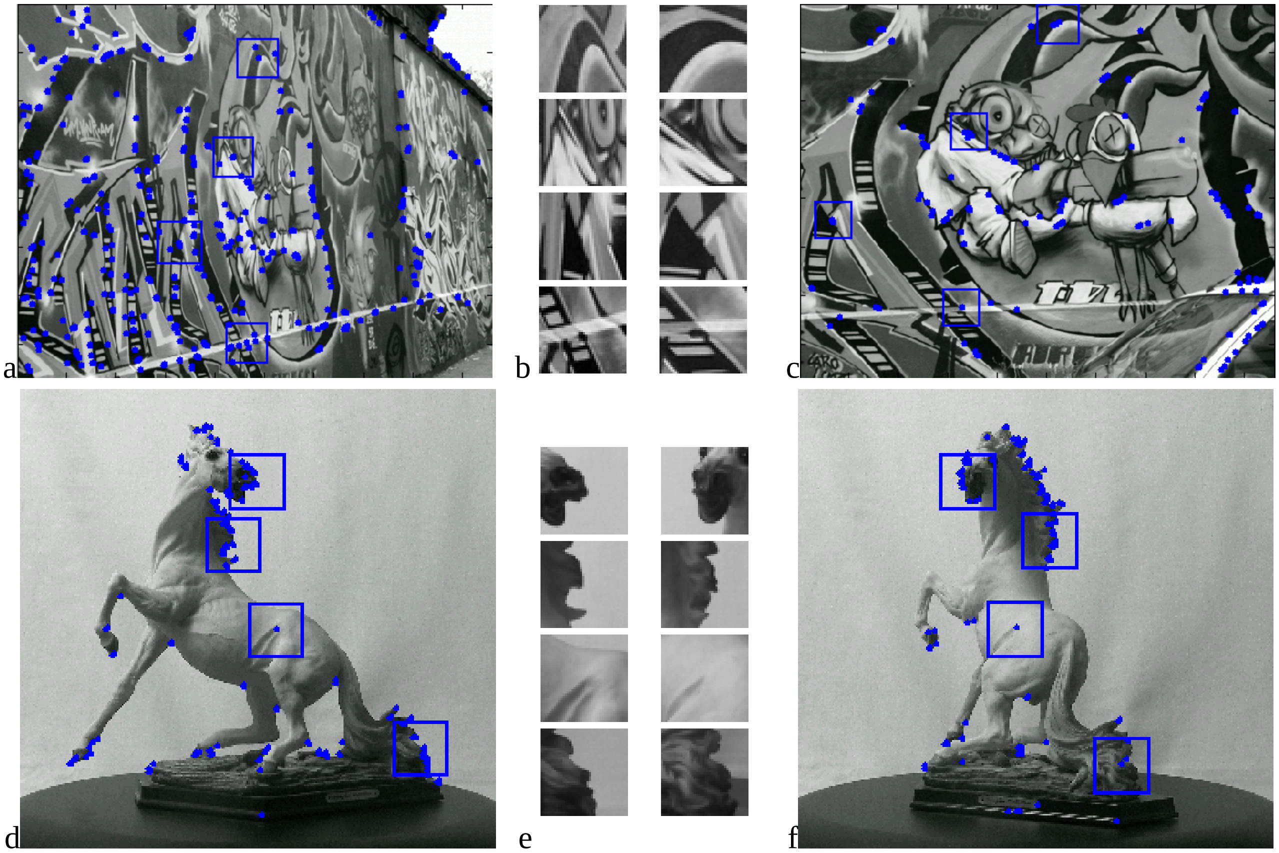


Figure 1: (Top row) Large ($\approx 50^\circ$) viewpoint change for a flat scene. Many interest points can be matched after the transformation. The appearance change is modeled by an affine transformation. Panel b. shows four 40x40 patches before and after viewpoint change - images courtesy of K.Mikolajczyk (Bottom row) Similar 50° viewpoint change for a 3D scene. Many visually salient features are associated with locations where the 3D surface is irregular or near boundaries. The local geometric structure of the image around these features varies rapidly with viewing direction changes, which makes matching features more challenging because of occlusion and changes in appearance. In particular, the appearance of the patches shown in panel e. varies significantly with the change in viewpoint. This change is difficult to model.

spatial extent or 'scale' of the feature may also be identified in this first step, as well as the local shape near the detected location [20, 19, 34, 35]. The second step is 'description': a vector characterizing local visual appearance is computed from the image near the nominal location of the feature. 'Matching' is the third step: a given feature is associated with one or more features in other images. Important aspects of matching are metrics and criteria to decide whether two features should be associated, and data structures and algorithms for matching efficiently.

The ideal system will be able to detect a large number of meaningful features in the typical image, and will match them reliably across different views of the same scene / object. Critical issues in detection, description and matching are robustness with respect to viewpoint and lighting changes, the number of features detected in a typical image, the frequency of false alarms and mismatches, and the computational cost of each step. Different applications weigh these requirements differently. For example, viewpoint changes more significantly in object recognition, SLAM and wide-baseline stereo than in image mosaicking, while the frequency of false matches may be more critical in object recognition, where thousands of potentially matching images are considered, rather than in wide-baseline stereo and mosaicing where only few images are present.

A number of feature detectors [19, 10, 1, 12, 20, 6], feature descriptors [18, 9, 2, 13] and feature matchers [28, 18, 5, 25] have been proposed in the literature. They can be variously combined and concatenated to produce different systems. Which combination should be used in a given application? A couple of studies explore this question. Schmid [28] characterized and compared the performance of several features detectors. Recently, Mikolajczik and Schmid [23] focused primarily on the descriptor stage. For a chosen detector, the performance of a number of descriptors was assessed. These evaluations of interest point operators and feature descriptors, have relied on the use of images of flat scenes, or in some cases synthetic images. The reason is that in these special cases the transformation between pairs of images can be computed easily, which is convenient to establish ground truth.

However, the relative performance of various detectors can change when switching from planar scenes to 3D images (see Figs., 1, 16 and [8]). Features detected in an image are generated in part by surface markings, and in part by the geometric shape of the object. The former are often associated with smooth surfaces, they are usually located far from object boundaries and have been shown to have a high stability across viewpoints [28, 23]. Their deformation may be modeled by an affine transformation, hence the development of affine-invariant detectors [17, 20, 27, 34, 35]. The latter are associated with high surface curvature and are located near edges, corners and folds of the object. Due to self-occlusion and complexity of local shape, these features have a much lower stability with respect to viewpoint change. It is difficult to model their deformation without a full 3D model of the shape.

The present study generalizes the analyses in [28, 13, 23] to 3D scenes [1]. We evaluate the performance of feature detectors and descriptors for images of 3D objects viewed under different viewpoint, lighting and scale conditions. To this effect, we collected a database of 100 objects viewed from 144 different

[1] An early version of this work was presented in [24]

calibrated viewpoints under 3 lighting conditions. We also developed a practical and accurate method for establishing automatically ground truth in images of 3D scenes. Unlike [8] ground truth is established using geometric constraints only, so that the feature/descriptor evaluation is not biased by the choice of a specific descriptor and appearance-based matches. Besides, our method is fully automated, so that the evaluation can be performed on a large-scale database, rather than on a handful of images as in [23, 8].

Another novel aspect is the use of a metric for accepting/rejecting feature matches due to D.Lowe [18]; it is based on the ratio of the distance of a given feature from its best match vs the distance to the second best match. This metric has been shown to perform better than the traditional distance-to-best-match.

Section 2 presents the previous work on evaluation of features detectors and descriptors. In section 3 we describe the geometrical considerations which allow us to construct automatically a ground truth for our experiments. Section 4 presents our laboratory setup and the database of images we collected. Section 5 describes the decision process used in order to assess performances of detectors and descriptors. Section 6 presents the experiments. Section 7 contains our conclusions.

2 Previous work

The first extensive study of features stability depending on the feature detector being used, was performed by Schmid & Mohr [29]. The database consisted of images of drawings and paintings photographed from a number of viewpoints. The authors extracted and matched interest points across pairs of views. The different views were generated by rotating and moving the camera as well as by varying the illumination. Since all scenes were planar, the transformation between two images taken from different viewpoints was a homography. Ground truth, i.e. the homography between pairs of views, was computed from a grid of artificial points projected onto the paintings. The authors measured the performance by the repeatability rate, i.e. the percentage of locations selected as features in two images.

Mikolajczyk et al. [21] performed a similar study of affine-invariant features detectors. This time, most images of the database consisted of natural scenes. However, the scenes were either planar (e.g. graffiti on a wall), or viewed from a large distance, such that the scene appeared flat. Therefore the authors could model the ground truth transformation between a pair of views with a homography as was previously done in [29]. This ground truth homography was computed using manually selected correspondences, followed by an automatic computation of the residual homography.

Note that the performance criterion used in both of these studies is well defined only when a small number of features is detected in each image. If the number of interest points is arbitrary, one could indeed consider a trivial interest point operator that selects every point in the image to be a new feature. The performance of this detector would be excellent in terms of stability of the features location. In particular for planar images such as considered by [21, 29], this detector would reach 100% stability. This perfect stability still holds if the detector selects a dense grid of points in the image. This argument illustrates the necessity of including the descriptor stage in performance evaluation.

Fraundorfer $\&$ Bischof [8] compared local detectors on real-world scenes. Ground truth was established in triplets of views. Correspondences were first identified between grids of points sampled densely in two close views: matches were obtained by nearest neighbor search in appearance space. The coordinates of pairs of matching points in the first two images, were transferred on the third image via the trifocal tensor. The test scenes used for detector evaluation were piecewise flat (building, office space).

Mikolajczyk $\&$ Schmid [23] provided a complementary study where the focus was not anymore on the detector stage but on the descriptor, i.e. a vector characterizing the local appearance at each detected location. Two interest points were considered a good match if their appearance descriptors was closer than a threshold t in appearance space. Matches that were accepted were compared to ground truth to determine if they were true matches or false alarms. Ground truth was computed as in their previous study [21]. By varying the acceptance threshold t, the authors generated recall-precision curves to compare the descriptors. If the value of t is small, the user is very strict in accepting a match based on appearance, which leads to a high precision but a poor recall. If t is high, all candidate correspondences are accepted regardless of their appearance. Correct matches are accepted (high recall), as well as lots of false positives, leading to lower precision.

Ke $\&$ Sukthankar [13] used a similar setup to test their PCA-SIFT descriptor against SIFT. Test features were indexed into a database, the resulting matches were accepted based on a threshold t on quality of the appearance match. Ground truth was provided by labeled images, or by using synthetic data. The threshold t was varied to obtain recall-precision curves.

A recent study by Mikolajczyk et al. [22] compared detectors and descriptors when they are integrated in the framework of the full recognition system from [14]. They assessed the performance from the performance of the overall system. The integration within a complete recognition method has the advantage of computing directly the bottom line performance in recognition. However, the scores might depend heavily on the architecture of the recognition system and may not be generalized to other applications such as large baseline stereo, SLAM and mosaicking.

3 Ground truth

In order to evaluate a particular detector-descriptor combination we need to calculate the probability that a feature extracted in a given image, can be matched to the corresponding feature in an image of the same object/scene viewed from a different viewpoint. For this to succeed, the feature's physical location must be visible in both images, the feature detector must detect it in both cases with minimal positional variation, and the descriptor of the features must be sufficiently close. To compute this probability we must have a ground truth telling us if any tentative match between two features is correct or not. Conversely, whenever a feature is detected in one image, we must be able to tell whether in the corresponding location in another image a feature was detected and matched.

We establish ground truth by using epipolar constraints between triplets of calibrated views of the objects (this is equivalent to using the trifocal tensor [32, 11]). The motivation comes from stereoscopic imagery: if the position of a point is identified in two calibrated images of a same scene, the position in 3D space of the physical point may be computed, and its location may be predicted in any additional calibrated image of the same scene.

We distinguish between a *reference* view (A in Fig.2 $\&$ Fig.3), a *test* view C, and an *auxiliary* view B. Given one *reference feature* f^A in the reference image, any feature in C that matches the reference feature must satisfy the constraint of belonging to the corresponding reference epipolar line l^{AC}. This excludes most potential matches but not all of them (in our experiments, typically 5-10 features remain out of 500-1000 features in image C). We make the test more stringent by imposing a second constraint. In the auxiliary image B, an epipolar line l^{AB} is associated to the reference feature f^A. Again, f^A has typically 5-10 potential matches along l^{AB}, each of which in turn generates an 'auxiliary' epipolar line $l^{BC}_{1...10}$ in C. The intersection of the primary (l^{AC}) and auxiliary ($l^{BC}_{1...10}$) epipolar lines in C identify a number of small matching regions, in which only zero or one features are typically detected. As we will make clear later, when a matching feature is found, this indicates with overwhelming probability that it is the correct match.

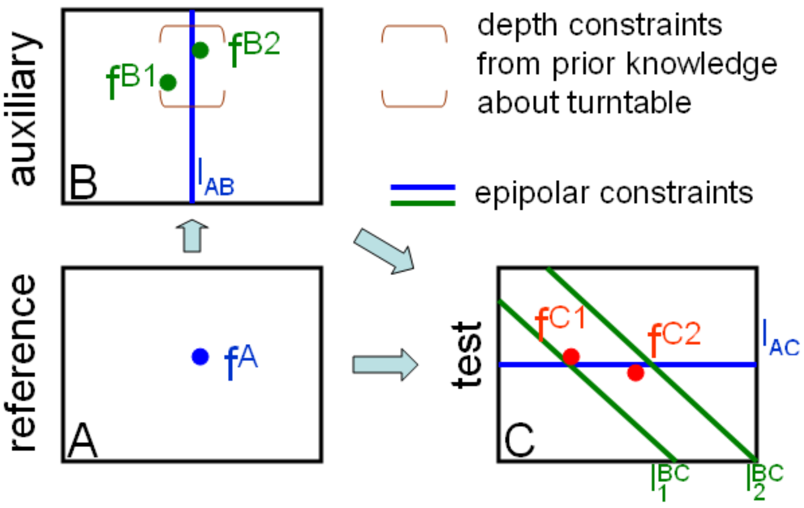


Figure 2: Diagram explaining the geometry of our three-cameras arrangement and of the triple epipolar constraint.

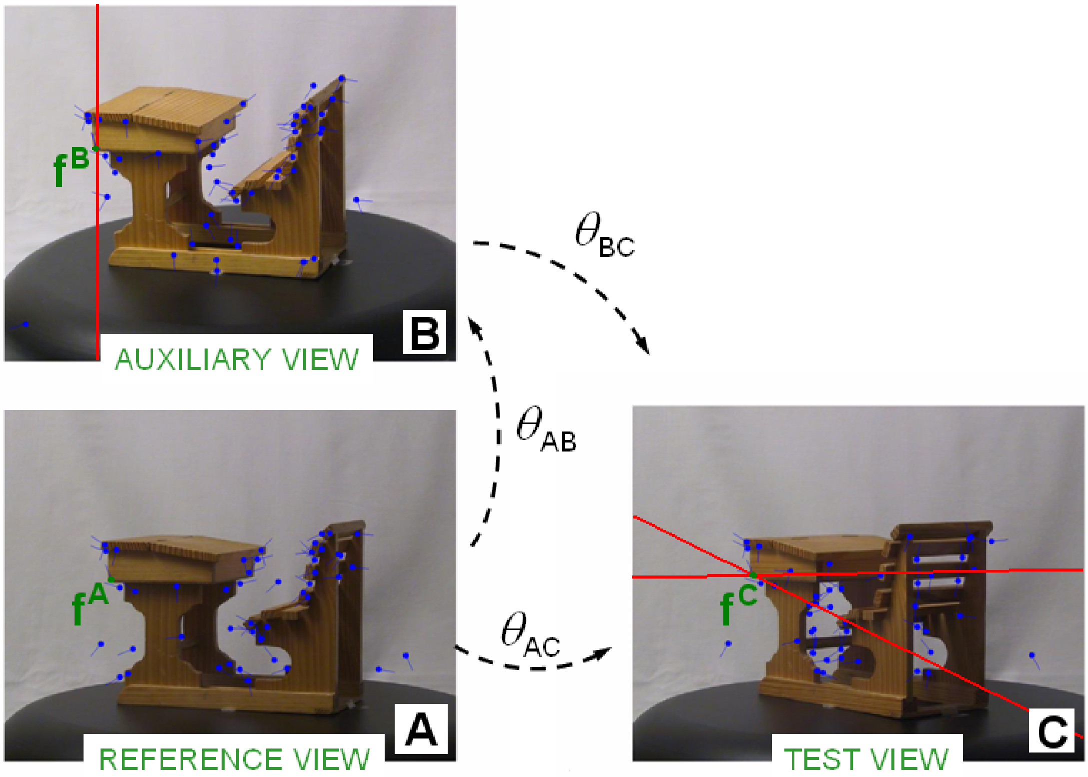


Figure 3: Example of matching process for one feature.

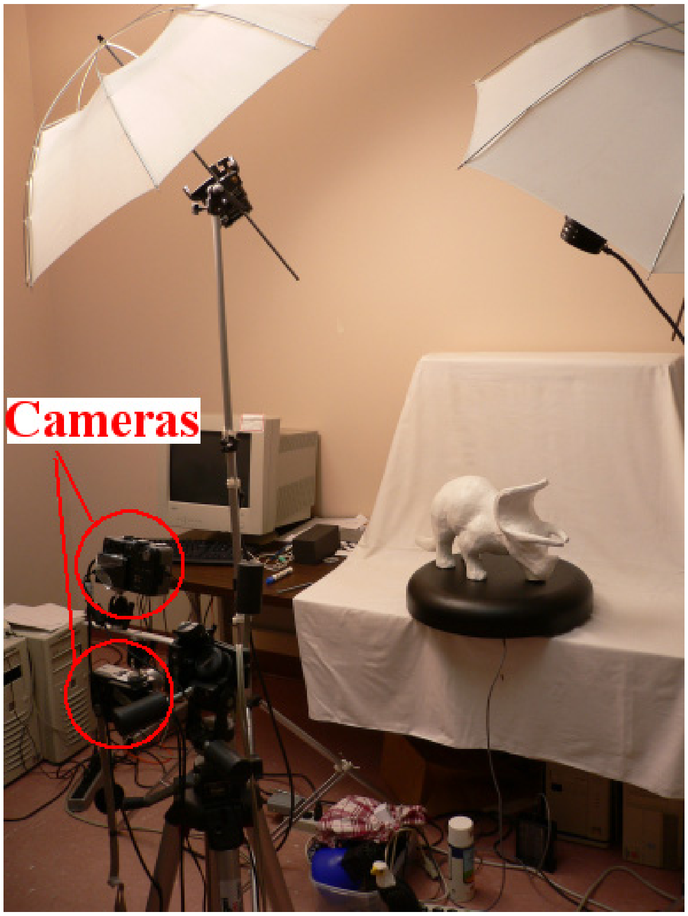


Figure 4: Photograph of our laboratory setup. Each object was placed on a computer-controlled turntable which can be rotated with $1/50$ degree resolution and 10^{-5} degree accuracy. Two computer-controlled cameras imaged the object. The cameras were located 10° apart with respect to the object. The resolution of each camera is 3Mpixels. In addition to a neon tube on the ceiling, two photographic spotlights with diffusers are alternatively used to create 3 lighting conditions.

Note that the geometry of our acquisition system (Fig.2 $\&$ Fig.3) does not allow the degenerate case where the reference point is on the trifocal plane. In this case, both epipolar constraints are superimposed and their intersection is not defined. In this case, the triangle (reference camera/auxiliary camera/test camera) would be a degenerate triangle.

The benefit of using the double epipolar constraint in the test image is that any correspondence - or lack thereof - may be validated with extremely low error margins. The cost is that only a fraction (50-70%) of the reference features have a correspondence in the auxiliary image, thus limiting the number of features triplets that can be formed.

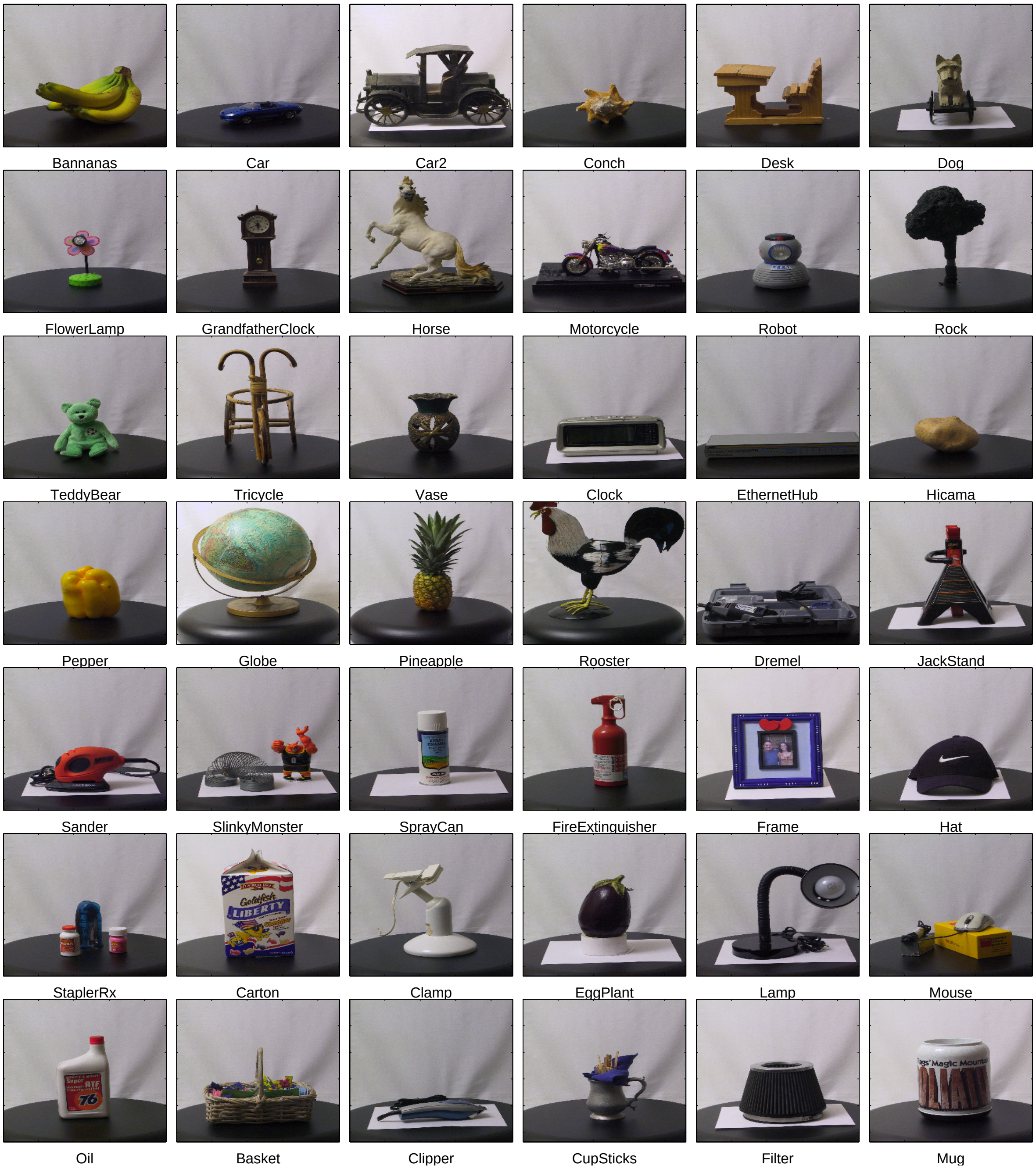
4 Experimental setup

4.1 Photographic setup and database

Our acquisition system consists of 2 cameras taking images of objects on a motorized turntable (see Fig.4). We used inexpensive off-the-shelf Canon Powershot G1 cameras with a 3 MPixels resolution. The highest focal length available on the cameras - 14.6mm - was used in order to minimize distortion (0.5% pincushion distortion with the 14.6mm focal length). A change in viewpoint is performed by the rotation of the turntable. The lower camera takes the reference view and the top camera the auxiliary view, then the turntable is rotated and the same camera takes the test view. Each acquisition was repeated with 3 different lighting conditions obtained with a combination of photographic spotlights and diffusers. The images were converted to gray-scale using Matlab (keeps luminance, eliminates hue and saturation).

The baseline of our stereo rig, or distance between the reference camera and the auxiliary camera, is a trade-off parameter between repeatability and accuracy. On one hand, we would like to set these cameras very close to each other, in order to have a high feature stability (also called repeatability rate) between the reference view and the auxiliary view. On the other hand, if the baseline is small the epipolar lines intersect in the test view C with a very shallow angle, which lowers the accuracy in the computation of the intersection. We chose an angle of $10°$ between reference camera and auxiliary camera; with this choice, the intersection angle between both epipolar lines varied between $65°$ and $6°$ when the rotation of the test view varied between $5°$ and $60°$.

The database consisted of 100 different objects. Fig.5-6-7 show some examples from this database. The objects were chosen to include both heavily textured objects (pineapple, globe) and objects with a more homogenous surface (bananas, horse). The only constraint on the objects' identity concerned their size. They had to be small enough to fit on the turntable (40 cm diameter), but needed to be large enough so that their image would generate a significant number of features. Aside from these constraints, the objects were selected randomly. Most objects were 3-dimensional, with folds and self-occlusions, which are a major cause of features instability in real-world scenes. A few piecewise-flat objects (e.g. box of cereals, bottle of motor oil) were also present.



Bannanas Car Car2 Conch Desk Dog

FlowerLamp GrandfatherClock Horse Motorcycle Robot Rock

TeddyBear Tricycle Vase Clock EthernetHub Hicama

Pepper Globe Pineapple Rooster Dremel JackStand

Sander SlinkyMonster SprayCan FireExtinguisher Frame Hat

StaplerRx Carton Clamp EggPlant Lamp Mouse

Oil Basket Clipper CupSticks Filter Mug

Figure 5: Our calibrated database consists of photographs of 100 objects which were imaged in three lighting conditions: diffuse lighting, light from left and light from right. Two people chose objects from the set they met in their daily life. The objects had to fit on the turntable and within the camera's field of view. The range of shape statistics was explored, ranging from wireframe-type objects (Tricycle) to irregular 3D objects (Car2, Desk). Textured objects (Pineapple, Globe) were included as well as homogenous ones (Hicama, Pepper). A few piecewise flat objects were imaged as well (Carton, Oil can). Each object was photographed by two cameras located above each over, 10° apart. 42 objects from the database are displayed.

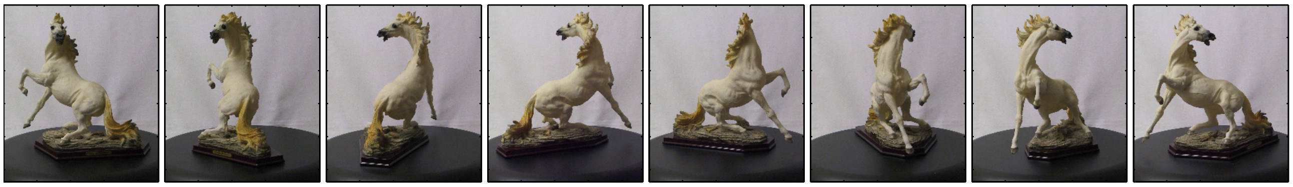


Figure 6: Each object was rotated with $5°$ increments and photographed at each orientation with both cameras and three lighting conditions for a total of $72 \times 2 \times 3 = 432$ photographs per object. Eight such photographs (taken every $45°$)are shown for one of our objects.

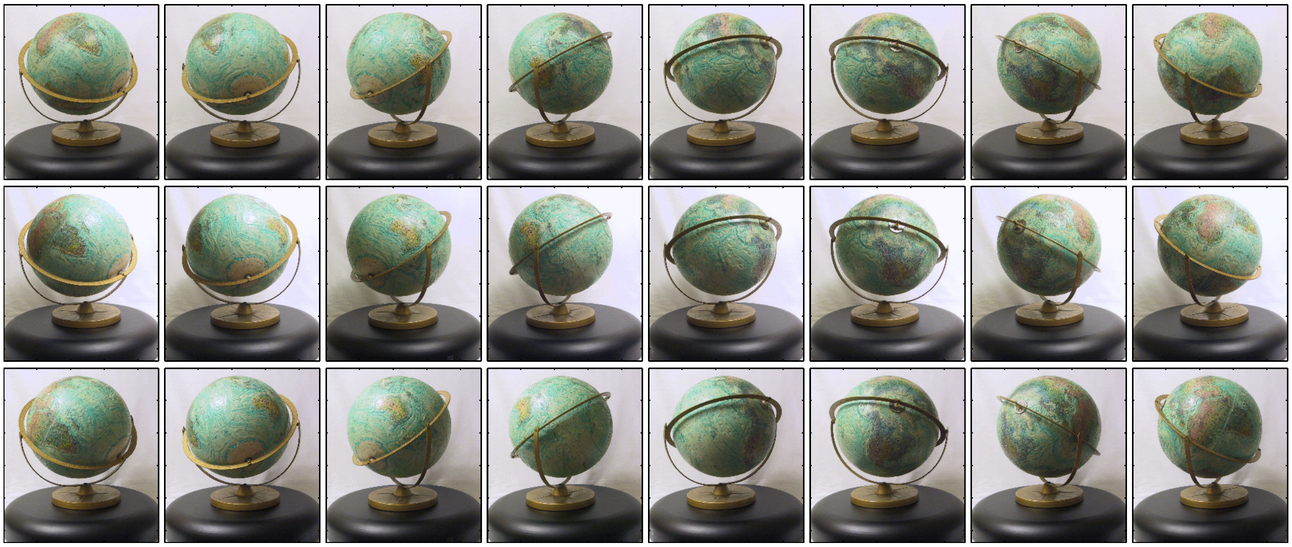


Figure 7: Three lighting conditions were generated by turning on a spotlight (with diffuser) located on the left hand side of the object, then a spotlight located on the right hand side, then both. This figure shows 8 photographs for each lighting condition.

4.2 Calibration

The calibration images were acquired using a checkerboard pattern. The corners of the checkerboard were first identified by the Harris interest point operator, then both cameras were automatically calibrated using the calibration routines in Intel's Open CV library, including the estimation of the radial distortion [3], which was used to map features locations to their exact perspective projection.

The uncertainty on the position of the epipolar lines was estimated by Monte Carlo perturbations of the calibration patterns. Hartley & Zisserman [11] showed that the envelope of the epipolar lines obtained when the fundamental matrix varies around its mean value, is a hyperbola. Rather than computing this curve analytically, we computed it by Monte Carlo simulation. The calibration patterns were perturbed by a random amount between 0 and 3 pixels (uniform distribution). This perturbation was performed by shifting the position of the corners in the checkerboard pattern after detection. This quantity was chosen so that it would produce a reprojection error on the grid's corners that was comparable to the one observed during calibration. The perturbation was followed by the calibration optimization.

For each point f^A of the reference image, the Monte-Carlo process leads to a bundle of epipolar lines in the test image, whose envelope is the hyperbola of interest. From our Monte Carlo simulation we found that the width between the two branches of the hyperbola varied between 3 and 5 pixels. The area inside the hyperbola defines the region allowed for detection of a match to f^A (a similar condition holds between reference and auxiliary images, and between auxiliary and test images).

4.3 Detectors and descriptors

4.3.1 Detectors

A large number of the traditional feature detectors follow the same general scheme. In a first step a *saliency map* is computed, which is a local function of the image. The saliency is a measure of the local contrast or local information content in the image. Patches with a high contrast (typically corners or highly textured areas) are expected to be detected and localized reliably between different images of the scene, therefore the local maxima of the saliency map are selected as features. This process is repeated after subsampling iteratively the image, to provide a multi-scale detector. In order to provide some invariance to noise, only local maxima that exceed a given threshold are selected.

- The Forstner detector [7] relies on first order derivatives of the image intensities. It is based on the second order moment matrix (also called squared gradient matrix)

$$\mu = \begin{bmatrix} L_x^2 & L_x \cdot L_y \\ L_x \cdot L_y & L_y^2 \end{bmatrix} \text{ where } L_x = \frac{\partial I}{\partial x} \text{ and } L_y = \frac{\partial I}{\partial y} \tag{1}$$

and selects as features the local maxima of the function $det(\mu)/tr(\mu)$. The second order moment matrix is a local measure of the variation of the gradient image. It is usually integrated over a small window in order to obtain robustness to noise and to make it a matrix of rank 2.

Several other features detectors use the second order moment matrix as well. The popular Harris detector [10] selects as features the extrema of the saliency map defined by $det(\mu) - 0.04 \cdot tr^2(\mu)$. The Lucas-Tomasi-Kanade feature detector [16, 33] averages μ over a small window around each pixel, and selects as features the points that maximize the smallest eigenvalue of the resulting matrix. The motivation for these 3 detectors, is to select points where the image intensity has a high variability both in the x and the y directions.

- The Hessian detector [1] is a second order filter. The saliency measure is here the negative determinant of the matrix of second order derivatives.

- Affine-invariant versions of the previous two detectors have been developed and used by [17, 20, 27, 34, 35]. The second order moment matrix is used as an estimation of the parameters of the local shape around the detected point. The goal is to deform the shape of the detected region so that it is invariant to affine transformations. The affine rectification process is an iterative warping method that reduces the image's local second-order moment matrix at the detected feature location, to have identical eigenvalues.

- The difference-of-Gaussians detector [6, 15] selects scale-space extrema of the image filtered by a difference of Gaussians. Note that the difference-of-Gaussians filter can be considered as an approximation of a Laplacian filter, i.e. a second-order derivative-based filter.

- The Kadir-Brady detector [12] selects locations where the local entropy has a maximum over scale and where the intensity probability density function varies fastest.

- MSER features [19] are based on a watershed flooding [37] process performed on the image intensities. The authors look at the rate of expansion of the segmented regions, as the flooding process is performed. Features are selected at locations of slowest expansion of the catchment basins . This carries the idea of stability to perturbations, since the regions are virtually unchanged over a range of values of the 'flooding level'.

Regarding speed, the detectors based Gaussian filters and their derivatives (Harris, Hessian, Difference-of-Gaussians) are fastest, they can easily be implemented very efficiently using the recursive filters introduced in [36]. The detection process typically takes 1s or less for a 3GHz machine on a 1024x768 image. If one uses the affine rectification process, computation is more expensive, similar detection takes of the order of 5 seconds. The most expensive detector is the Kadir-Brady detector, which takes of the order of 1 minute on a 800x600 image.

4.3.2 Descriptors

The role of the descriptor is to characterize the local image appearance around the location identified by the feature detector. Invariance to noise is usually obtained by low-pass filtering. Partial invariance to lighting conditions is obtained by considering image derivatives instead of the raw greylevels.
- SIFT features [18] are computed from gradient information. Invariance to orientation is obtained by evaluating a main orientation for each feature and rotating the local image according to this orientation prior to the computation of the descriptor. Local appearance is then described by histograms of gradients, which provides a degree of robustness to translation errors.
- PCA-SIFT [13] computes a primary orientation similarly to SIFT. Local patches are then projected onto a lower-dimensional space by using PCA analysis.
- Steerable filters descriptor [9] are generated by applying banks of oriented Gaussian derivative filters to an image. This achieves invariance to in-plane rotation without having to choose a preferred feature orientation, but at the expense of having an overcomplete representation of the image. Scale invariance is achieved by using various filter sizes.
- Differential invariants [28] combine local derivatives of the intensity image (up to 3rd order derivative) into quantities which are invariant with respect to rotation.
- The shape context descriptor [2] is based on edges. Edges are extracted with the Canny filter, their location and orientation are then quantized into histograms using log-polar coordinates.

5 Performance evaluation

5.1 Matching criteria

The performance of the different combinations of detectors and descriptors was evaluated on a feature matching problem. Each feature f^C from a test image C was appearance-matched against a large database of features. The nearest neighbor in this database was selected and tentatively matched to the feature. The database contained both features from a reference image A of the same object ($10^2 - 10^3$ features depending on the detector and on the image), as well as a significantly larger number (10^5) of features from unrelated images. Using this large database replicates the matching process in object/class recognition applications, where incorrect pairs can arise from matching features to wrong images.

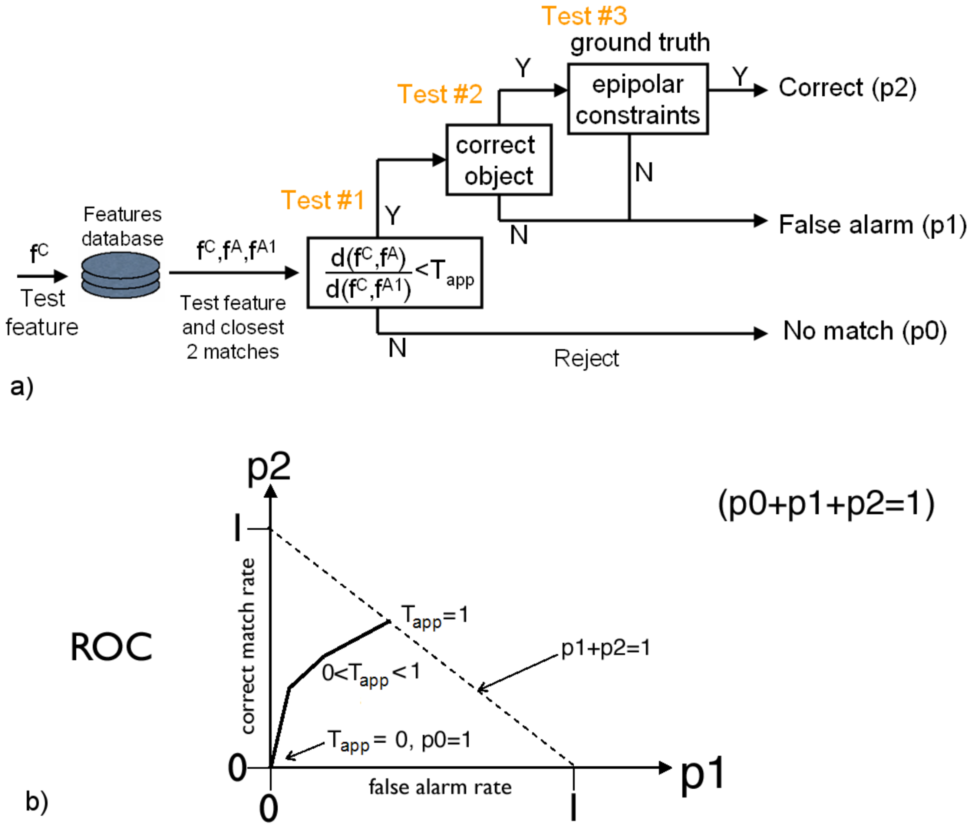


Figure 8: (Panel a) Diagram showing the process used to classify feature triplets. (Panel b) Conceptual shape of the ROC trading off false alarm rate with detection rate. The threshold T_{app} on distance ratios (sec.5.2) is bounded by $[0, 1]$ cannot take values larger than 1 and the ROC is bounded by the curve $p1 + p2 = 1$.

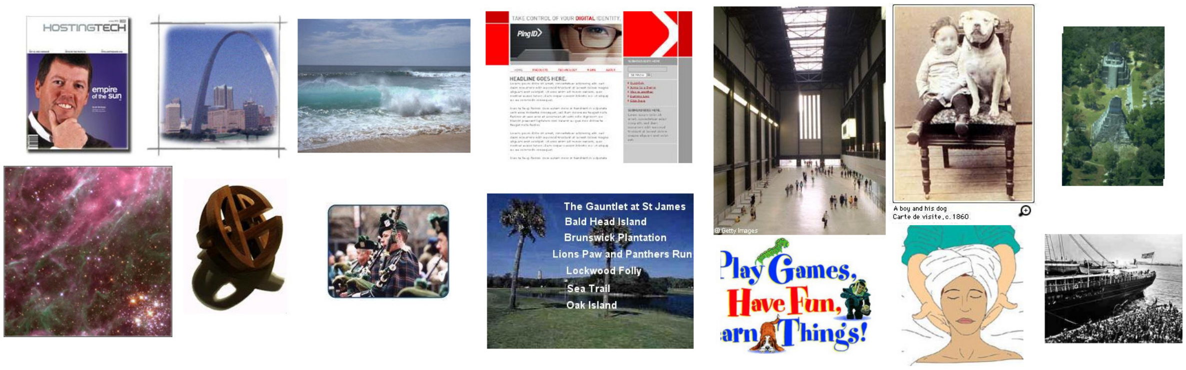


Figure 9: A few examples of the 535 irrelevant images that were used to load the feature database. They were obtained from Google by typing ‘things’. 10^5 features detected in these images were selected at random and included in our database.

The diagram in Fig.8-a shows the decision strategy. Starting from feature f^C from the test image C, a candidate match to f^C is proposed by selecting the most similar amongst the whole database of features. The search is performed in appearance space. The feature returned by the search is accepted or rejected ($Test\#1$) based on the distance metric ratio that will be described in sec.5.2. The candidate match is accepted only if the ratio lies below a user-defined threshold T_{app}.

If the candidate match is accepted based on this appearance test, the next stages aim at validating this match. $Test\#2$ checks the identity of the image from which the proposed match is coming. If it comes from the image of an unrelated object, the proposed match cannot correspond to the same physical point. The match is rejected as a false alarm.

$Test\#3$ validates the proposed match based on geometry. The test starts from the proposed match f^A in the reference image, it uses the epipolar constraints described in sec.3 and tries to build a triplet (initial feature - auxiliary feature - proposed match) that verifies all epipolar conditions (one constraint in the auxiliary image and two constraints in the test image). As mentioned in sec.3, typically only zero or one features from the test image verify all epipolar constraints generated by a given feature from the reference image. If this feature from the test image is precisely our test feature f^C, the proposed match is declared validated and is accepted. In the alternative this is a false alarm.

In case no feature was found along the epipolar line in the auxiliary image B, the initial point f^C is discarded and doesn't contribute to any statistics, since our inability to establish a triple match is not caused by a poor performance of the detector on the target image C.

Note that this method doesn't guarantee the absence of false alarms. False alarms can arise if an incorrect auxiliary feature is used during the geometric validation - as we will see, they are very few. However, our method offers the important advantage of being purely geometric. Any system involving appearance vectors as an additional constraint would be dependent on the underlying descriptor and bias our evaluation.

In order to evaluate the fraction of incorrect correspondences established and accepted by our geometric system, 2 experts examined visually the triplets accepted by the system and classified them into correct and incorrect matches. 3000 matches were selected randomly from the accepted triplets and were visually classified, results are reported in Fig.10. The users also classified matches obtained by a simpler method that would use only two images of the object (reference and test view) and a single epipolar

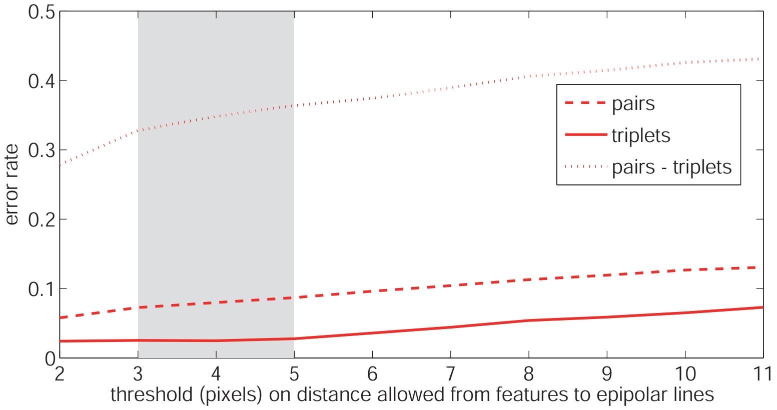


Figure 10: Operator-assisted validation of our automated ground truth. A sample of 3000 pairs and triplets was randomly selected from the set of automatically computed tentative feature matches. Two experts classified each pair and triplet by hand as to whether it was correct or not. The fraction of wrong triplets is displayed as a function of the maximum distance allowed to the epipolar line (curve 'triplets'). Our experiments were conducted using adaptive thresholds of 3-5 pixels (gray-shaded zone, see section 4.2), which as the plot shows yields 2% of incorrect triplets. A method based on a single epipolar line constraint ('pairs') would have entailed a rate of wrong correspondences three times higher. In particular, the rate of wrong correspondences is very high for features that could be matched in two images but not in all 3 images ('pairs $-$ triplets').

constraint: in this case the geometric validation consists of checking whether or not the test feature lies on the epipolar line generated by the proposed match in the test view. The fraction of incorrect matches is displayed as a function of the threshold on the maximum distance in pixels allowed between features and epipolar lines. We also display the error rate for features that were successfully matched according to the 2-views method, but failed according to the 3-views method. The method using 3 views yields a significantly better performance: when the threshold on acceptable distances to epipolar lines varies between 3 and 5 pixels (see sec.4.2), the error rate of the 3-views method is 2%, while the error rate of the 2-views method is three times higher at 6%.

5.2 Distance measure in appearance space

In order to decide on acceptance or rejection of a candidate match ($Test\#1$ in Fig.8), we need a metric on appearance space. Instead of using directly the Euclidean or Mahalanobis distance in appearance as in [23, 13], we use the distance ratio introduced by Lowe [18].

The proposed measure compares the distances in appearance of the query point to its best and second best matches. In Fig.8 the query feature and its best and second best matches are denoted by f^C, f^A

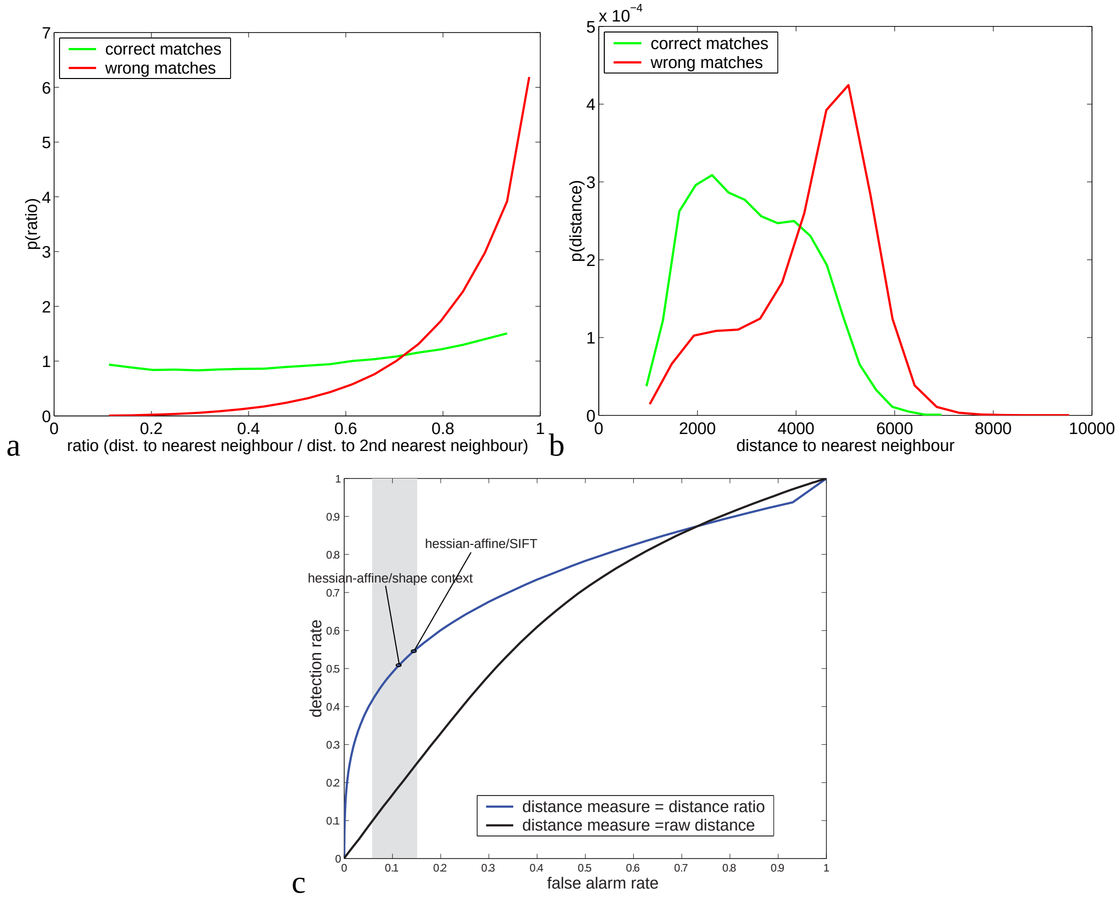


Figure 11: Panel a: Sample pdf of the distance ratio between best and second best match for correct correspondences (green) and false alarms (red). These curves are analogous to the ones in Fig.11 of Lowe [18]. Lowe's correct-match density is peaked around 0.4 while ours is flat – this may be due to the fact that we use 3D objects, while D.Lowe uses flat images with added noise. Panel b:. Distributions obtained using the distance to best match. Panel c: comparative ROC curves obtained from the distance ratio distributions in a. and the raw distance distributions in b. The distance ratio clearly performs better.

and f^{A1} respectively. The criterion used is the ratio of these two distances, i.e. $\frac{d(f^C,f^A)}{d(f^C,f^{A1})}$. This ratio characterizes how distinctive a given feature is, and avoids ambiguous matches. A low value means that the best match performs significantly better than its best contender, and is thus likely to be a reliable match. A high value of the distance ratio is obtained when the features points are clustered in a tight group in appearance space. Those features are not distinctive enough relatively to each other. In order to avoid a false alarm it is safer to reject the match.

The distance ratio is a convenient measure for our study, since the range of values it can take is always $[0, 1]$ no matter what the choice of descriptor is.

Fig.11-a shows the resulting distribution of distance ratios conditioning on correct or incorrect matches. The distance ratios statistics were collected during the experiments in sec.6. Correct matches and false alarms were identified using the process described in 5.1. Fig.11-b shows the distributions of 'raw distance to nearest neighbor' conditioning on correct or incorrect matches. Since distances depend on the chosen descriptor, the descriptor chosen here was SIFT.

Fig.11-c motivates further the use of the distance ratio by comparing it to raw distance on a classification task. We computed ROC curves on the classification problem 'correct vs. incorrect match', based on the conditional distributions from Fig.11-a and Fig.11-b. The parameter being varied to generate the ROC is the threshold T_{app} which decides if a match is correct or incorrect. Fig.11-c displays the results. Depending on the descriptor, the operating point chosen for the comparisons in sec.6 leads to value of T_{app} between 0.56 and 0.70. In the ROC curves from Fig.11-c, these values are highlighted by a shaded area. In this operating region, the distance ratios clearly outperform raw distances by a factor 3 to 5 in terms of detection rate.

5.3 Detection and false alarm rates

As seen in the previous section and Fig.8, the system can have 3 outcomes. In the first case, the match is rejected based on appearance (probability p_0). In the second case, the match is accepted based on distance in appearance space, but the geometry constraints are not verified and ground truth rules the match as incorrect : this is a false alarm (probability p_1). In the third alternative, the match verifies both appearance and geometric conditions, this is a correct detection (probability p_2). These probabilities verify $p_0 + p_1 + p_2 = 1$. The false alarm rate is further normalized by the number of database features (10^5). This additional normalization was an arbitrary choice, motivated by the dependency of the false alarm rate on the size of the database: the larger the database, the higher the risk of obtaining an incorrect match during the appearance-based indexing described in sec.5.1. Detection rate and false alarm rate can be written as

$$false\_alarm\_rate = \frac{\#false\ alarms}{\#attempted\ matches \cdot \#database} \tag{2}$$

$$detection\_rate = \frac{\#detections}{\#attempted\ matches} \tag{3}$$

By varying the threshold T_{app} on the quality of the appearance match, we obtain a ROC curve (Fig.8-b). Note that the detection rate does not necessarily reach 1 when T_{app} is lowered to zero since some features

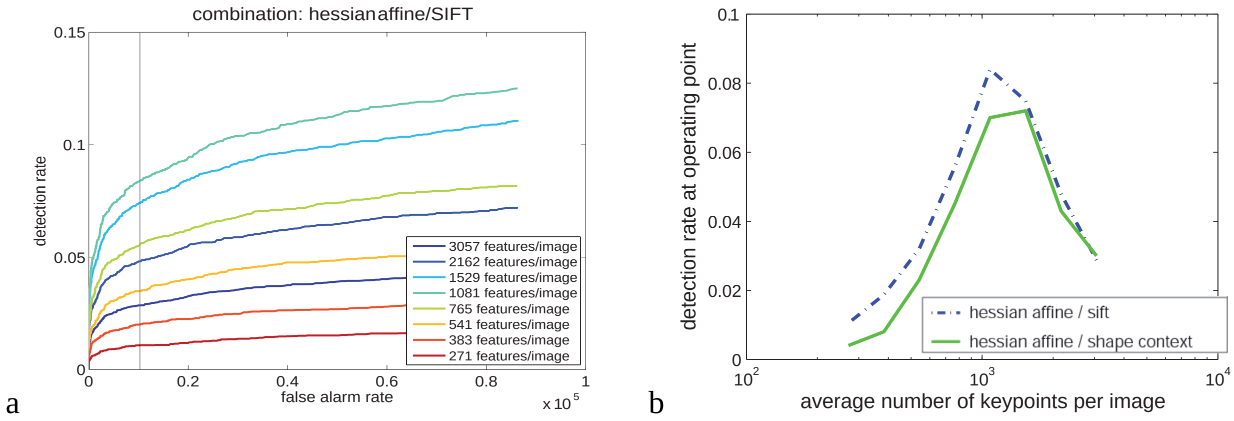


Figure 12: Panel a: ROCs obtained when varying the threshold T_{det} on minimum saliency that a region has to satisfy in order to be declared a feature. The legend shows the average number of features detected per image. The operating point is displayed by a vertical line. ROCs are displayed for the hessian-affine/SIFT combination. Panel b: performance at the operating point, as a function of the average number of features per image. Results are displayed for the two combinations that performed best in sec.6.1

will fail Tests$\#2\&3$ on object identity and on geometry.

5.4 Number of detected features

For the detectors based on extrema of a saliency map, the threshold T_{det} that determines the minimum saliency necessary for a region to be considered as a feature, is an important parameter. If many features are accepted, the distinctiveness of each of them might be reduced, as the appearance descriptor of one feature will be similar to the appearance of a feature located only a few pixels away. This causes false alarms during appearance-based indexing of features in the database. Conversely, if T_{det} is set to a high value and only few highly salient regions are accepted as features, missed detections will occur when a region has been detected in one image but didn't make it to the threshold level in the second image. In order to use each detector/descriptor combination at its optimal performance level, we performed the matching process described in sec.5.1 with a range of values of T_{det}. These values were chosen such that the number of features would vary from ≈ 270 features (one feature every 2200 pixels on the objects), up to ≈ 3000 features (one feature every 200 pixels on the object), with increments by a factor of $\sqrt{2}$ in the number of features. Similarly to sec.6, we choose the operating point at the false alarm rate 10^{-6}. As expected, the detection rate for this operating point first increases, then decreases when the number of features is increased. Fig.12-a shows the ROC curves obtained for the combination hessian-affine/SIFT.

The operating point is indicated by a vertical line. Fig.12-b shows at this operating point, the detection rate as a function of the number of features detected per image, for the two combinations that performed best in sec.6: hessian-affine/SIFT and hessian-affine/shape context. In the experiments from sec.6, the value of T_{det} corresponding to the highest detection rate was chosen for the various detectors/descriptors.

6 Results and Discussion

6.1 Viewpoint change

Fig.13 shows the detection results when the viewing angle was varied and lighting/scale was held constant. Panels a-h display results when varying the feature detector for a given image descriptor. Panels a-d display the ROC curves obtained by varying the threshold T_{app} in the first step of the matching process (threshold on distinctiveness of the features' appearance). The number of features tested is displayed in the legend. Panels e-h show the detection rate as a function of the viewing angle for a fixed false alarm rate of 10^{-6} was chosen (one false alarm every 10 attempts - this is displayed by a gray line in the ROC curves from Fig.13-15). This false alarm rate corresponds to different distance ratio thresholds for each detector / descriptor combination. Those thresholds varied between 0.56 and 0.70 (a bit lower than the 0.8 value chosen by Lowe in [18]). Fig.14a-b summarize for each descriptor, the detector that performed best.

The Hessian-affine and difference-of-Gaussians detectors performed consistently best with all descriptors. While the absolute performance of the various detectors varies when they are coupled with different descriptors, their rankings vary very little. The combination of Hessian-affine with SIFT and shape context obtained the best overall score, with SIFT slightly ahead. In our graphs the false alarm rate was normalized by the size of the database (10^5) so that the maximum false alarm rate was 10^{-5}. The PCA-SIFT descriptor is only combined with difference-of-gaussians, as was done in [13]. PCA-SIFT didn't seem to outperform SIFT as would be expected from [13].

Note that the Difference-of-Gaussians detector performed consistently almost as well as Hessian-affine. The Difference-of-Gaussians is simpler and faster, this motivates its use in fast recognition systems such as [18].

In the stability curves, the fraction of stable features doesn't reach 1 when $\theta = 0°$. This is due to

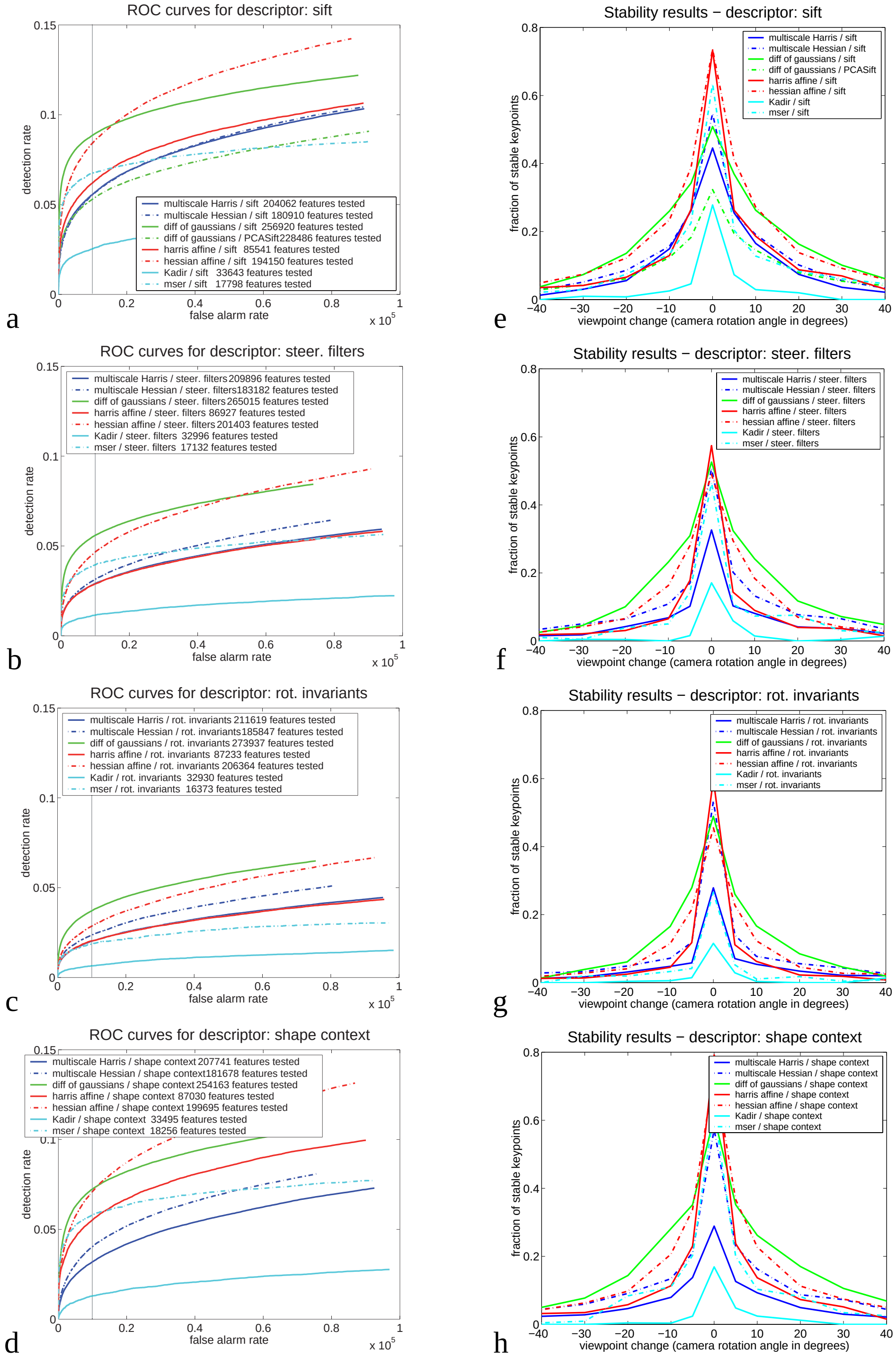


Figure 13: Performance for viewpoint change - each panel a-d shows the ROC curves for a given descriptor when varying the detector. Panels e-h show the corresponding stability rates as a function of the rotation angle. The 0° value is computed by matching features extracted from different images taken from the same location. The operating point chosen for the stability curves on the right hand side is highlighted by a vertical line in the ROCs.

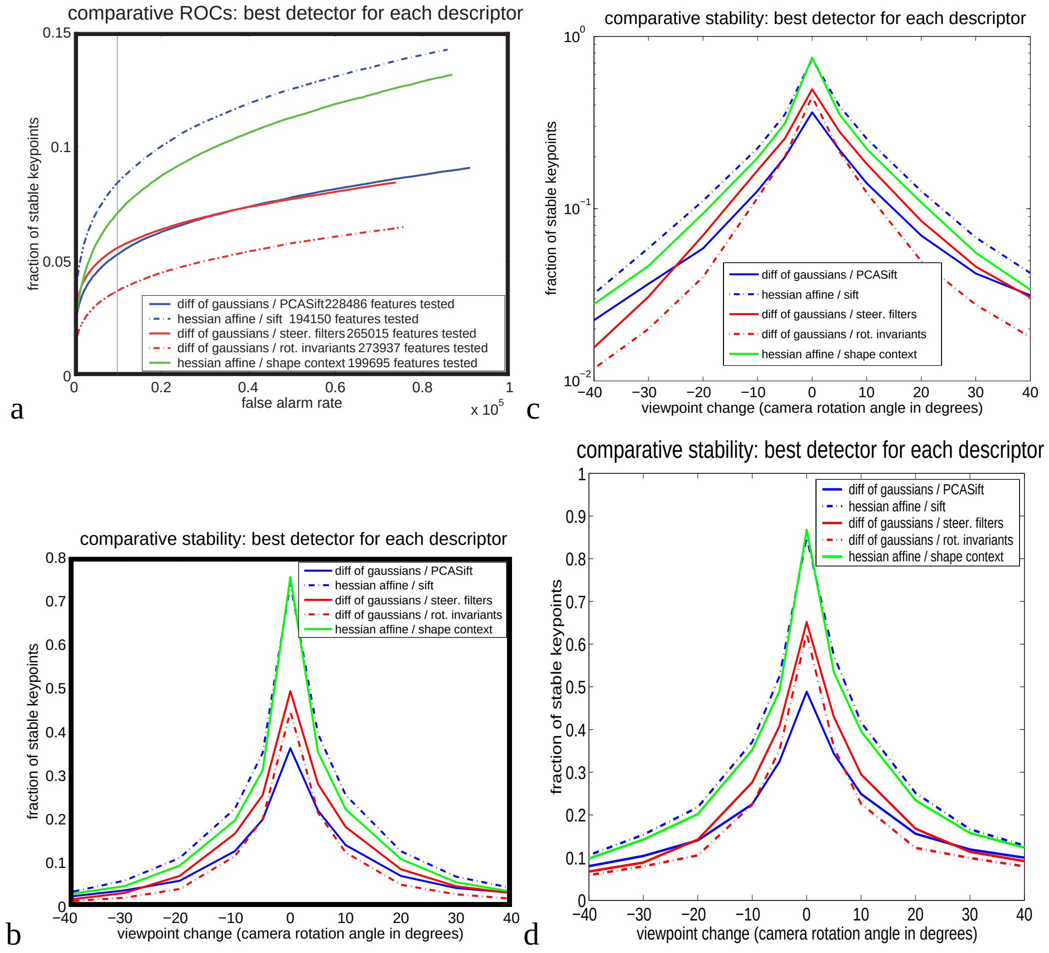


Figure 14: Summary of performance for viewpoint change - Panels a-b show the combination of each descriptor with the detector that performed best for that descriptor. Panel c. displays the stability results on a semi-log scale. Panel d is similar to panel b, but the database used for the search tree contained only the features extracted from the correct image (easier task which mimicks wide-baseline stereo).

several factors: first, triplets can be identified only when the match to the auxiliary image succeeds (see section 3). The $10°$ viewpoint change between reference and auxiliary image prevents a number of features from being identified in both images.

Another reason lies in the tree search. The use of a tree that contains both the correct image and a large number of unrelated images replicates the matching process used in recognition applications. However, since some features have low distinctiveness, the correct image doesn't collect all the matches. In order to evaluate the detection drop due to the search tree, the experiment was run again with a search tree that contained only the features from the correct image. Fig.14-c shows the stability results, the performance

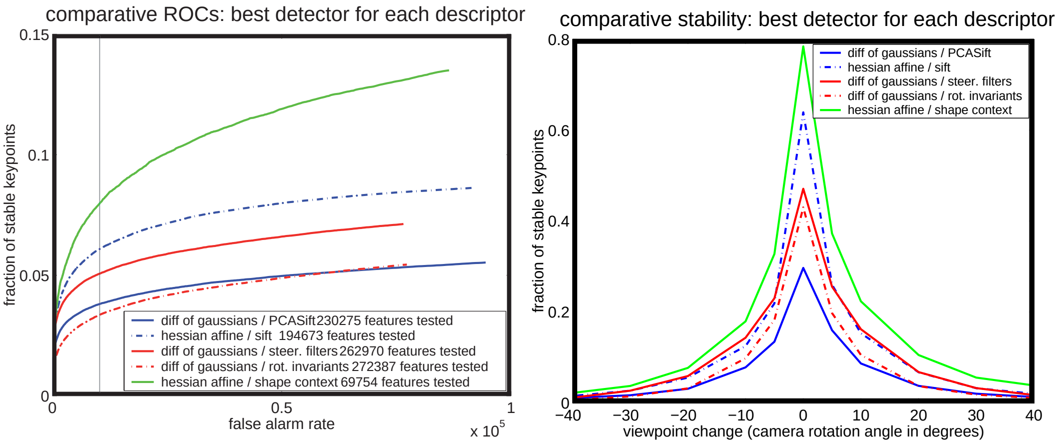


Figure 15: Results for viewpoint change, using the Mahalanobis distance instead of the Euclidean distance on appearance vectors.

is $10-15\%$ higher.

A third reason is the noise present in the camera. On repeated images taken from the same viewpoint, this noise causes $5-10\%$ of the features to be unstable.

Another observation concerns the dramatic drop in number of matched features with viewpoint change. For a viewpoint change of 30° the detection rate was below 5%.

Fig.15 shows the results ('summary' panel only) when the Euclidean distance on appearance descriptors is replaced by the Mahalanobis distance. Most relative performances were not modified. Hessian-affine performed again best, while shape context and SIFT were the best descriptors. In this case, shape context outperformed SIFT.

6.2 Normalization

As mentioned above, the matching performance between images A and C is affected by the inability to find a match in the auxiliary image B. One could want to normalize out this loss in order to get 'pure' stability results between A and C.

Let's denote by $p(\theta)$ and $p(\theta_1,\theta_2)$ the probabilities that given a reference feature, a match will respectively exist in one view of the same scene taken from a viewpoint θ degrees apart (for pairs), and in two views taken from viewpoints θ_1 and θ_2 apart from the reference image (triplets). If we assume independence between the matching processes from A to B and from A to C, we can decompose $p(\theta_{AB},\theta_{AC})$ into $p(\theta_{AB},\theta_{AC}) = p(\theta_{AB})p_{f^A}(\theta_{AC})$ and normalize by $p(\theta_{AB}) = p(10^\circ)$ to obtain absolute performance figures between A and C.

Unfortunately, it seems that the matching processes from A to B and from A to C cannot be considered to be independent. First, Fig.10 shows a different behavior between features that were successfully matched between A, B and C, and the features that were matched between A and C, but for which the match $A - B$ failed. In the latter case, the fraction of incorrect matches is much higher. Another hint comes from the stability results from Fig.14-c. Note that all combinations detectors/descriptors show a comparable performance of $6 - 10\%$ when the rotation is $40°$. If we were to normalize by $p(10°)$, the combination that performs worst at $0°$ (i.e. difference-of-Gaussians/PCA-SIFT) would by far perform best at $40°$. It seems very unlikely that a combination that performs poorly in easy conditions, would outperform all others when matching becomes more difficult. Therefore we believe that matches between A and B and between A and C are not independent. In order to avoid any inconsistency, we did not normalize the stability results. Our system is only collecting the most stable features, those that were not only stable between A and C, but were successfully matched into triplets.

6.3 Flat vs. 3D scenes

As mentioned in sec.1, one important motivation for the present study is the difference in terms of stability between texture-generated features extracted from images of flat scenes, and geometry-generated features from 3D scenes. In order to illustrate this stability difference, we performed the same study as in Sec.6.1, on one hand with 2 images of piecewise flat objects (box of cookies, can of motor oil), on the other hand on two objects with a more irregular surface (toy car and dog). Results are displayed in Fig.16. As expected, the stability is significantly higher for features extracted from the flat scenes. Note that the stability curves are not as symmetrical with respect to the $0°$ value as the curves in Fig.13-14. This is due to the fact that here the results are only averaged over a small number of objects.

One interesting result was that the relative performance of the various combinations detector/descriptor was modified between flat and 3D objects. Panels e-f display stability results respectively for rotations of $10°$ and $40°$. The fractions of stable features from flat scenes is displayed on the x axis, for 3D scenes it is on the y axis. All combinations lie below the diagonal $x = y$ since stability is lower for 3D scenes. Some changes in relative performances are highlighted. For example, for flat scenes MSER/SIFT and MSER/shape context performed best, while their performance was only average for 3D scenes. Con-

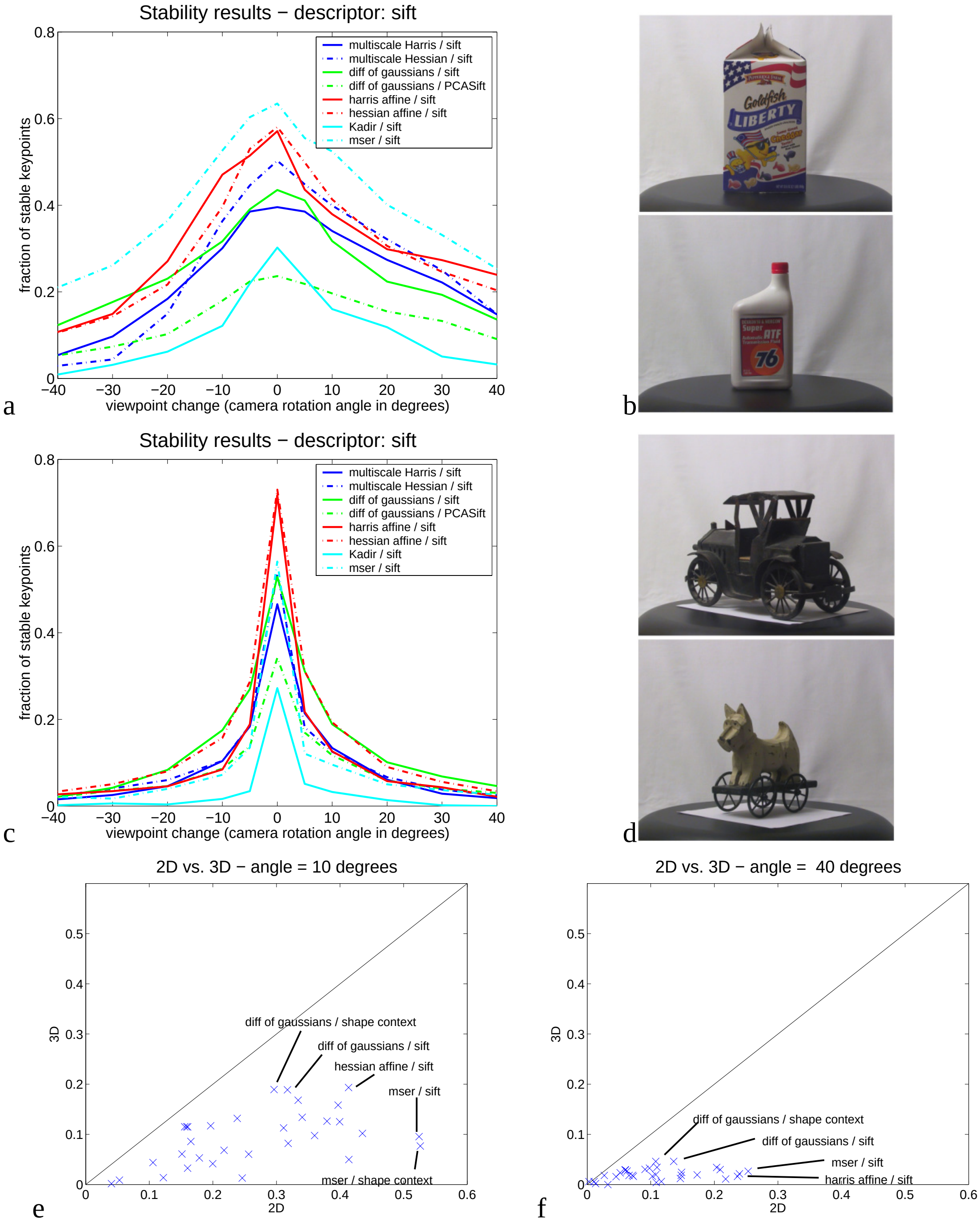


Figure 16: Flat vs. 3D objects - Panel a. shows the stability curves obtained for SIFT for the two piecewise flat objects in panel b. Similarly, panel c. shows the SIFT stability curves for the two 3D objects in panel d. 'Flat' features are significantly more robust to viewpoint change. Panels e-f show the fractions of stable features for the same piecewise 2D objects versus the same 3D objects, for all combinations of detectors / descriptors in this study. Scatter plots are displayed for rotations of 10° and 40°. A few combinations whose relative performance changes significantly are highlighted.

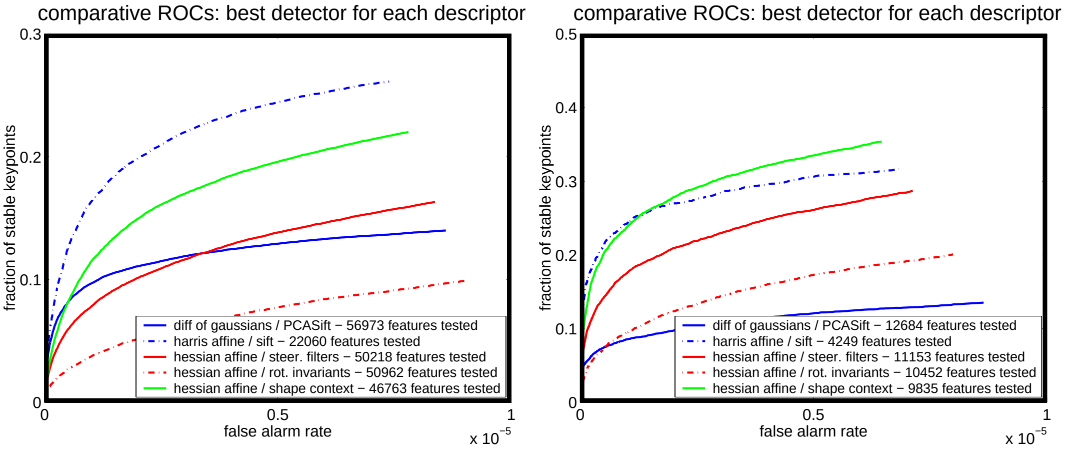


Figure 17: (Left panel) ROCs for variations in lighting conditions. Results are averaged over 3 lighting conditions. (Right panel) ROCs for variations in scale.

versely, difference-of-Gaussians/SIFT, difference-of-Gaussians/shape context, and hessian-affine/shape context, which were the best combinations for 3D scenes, were outperformed on 2D objects.

6.4 Lighting and scale change

Fig.17(left) shows the results obtained when changing lighting conditions and keeping the viewpoint unchanged. This task is easier: since the position of the features shouldn't change, we don't need to introduce the auxiliary image B. As a result, the detection rates reported in the ROC curves are significantly higher than in the study of viewpoint changes. Only the 'summary' panels with the best detector for each descriptor are displayed. This time, the combination which achieved best performance was Harris-affine combined with SIFT.

Fig.17(right) displays the results for a change of scale. The scale change was performed by switching the camera's focal length from 14.6mm to 7.0. Again, the figure displays only the 'summary' panel. Hessian-affine combined with shape context and Harris-affine combined with SIFT obtained the best results.

7 Discussion and Conclusions

We compared the most popular feature detectors and descriptors on a benchmark designed to assess their performance in recognition of 3D objects. In a nutshell: we find that the best overall choice is using an affine-rectified detector [20] followed by a SIFT [18] or shape-context descriptor [2]. These detectors and descriptor were the best when tested for robustness to change in viewpoint, change in lighting and change in scale. Amongst detectors, runner-ups are the Hessian-affine detector [20], which performed well for viewpoint change and scale change, and the Harris-affine detector [20], which performed well for lighting change and scale change.

Our benchmark differs from previous work from Mikolajczyk & Schmid in that we use a large and heterogeneous collection of 100 3D objects, rather than a handful of flat scenes. We also use Lowe's ratio criterion, rather than absolute distance, in order to establish correspondence in appearance space. This is a more realistic approximation of object recognition. A major difference with their findings is a significantly lower stability of 3D features. Only a small fraction of all features (less than 3%) can be matched for viewpoint changes beyond 30°. The situation is a bit better when the goal is stereo-vision or mosaicking (Fig.14-c), where features are matched across a small number of images. Our results on descriptors favor SIFT and shape context descriptors, and are in agreement with [23]. However, regarding detectors, not all affine-invariant methods are equivalent as suggested in [21], e.g. MSER performs poorly on 3D objects while it is very stable on flat surfaces.

We find significant differences in performance with respect to a previous study on 3D scenes [8]. One possible reason for these differences is the particular statistics of their scenes, which appear to contain a high proportion of highly textured quasi-flat surfaces (boxes, desktops, building facades, see Fig.6 in [8]). This hypothesis is supported by the fact that our measurements on piecewise flat objects (Fig.16) are more consistent with their findings. Another difference with their study is that we establish ground truth correspondence purely geometrically, while they use appearance matching as well, which may bias the evaluation.

An additional contribution of this paper is a new method for establishing geometrical features matches in different views of 3D objects. Using epipolar constraints, we are able to extract with high reliability (2% wrong matches) ground truth matches from 3D images. This allowed us to step up detector-descriptor evaluations from 2D scenes to 3D objects. Comparing to other 3D benchmarks, the ability to rely on an automatic method, rather than painfully acquired manual ground truth, allowed us to work with a large number of heterogeneous 3D objects. Our setup is inexpensive and easy to reproduce for collecting statistics on correct matches between 3D images. In particular, those statistics will be helpful for tuning recognition algorithms such as [18, 5, 25, 26]. Our database of 100 objects viewed from 72 positions with three lighting conditions will be available on our web site.

8 Acknowledgments

The authors are grateful to Timor Kadir, Yan Ke, David Lowe, Jiri Matas and Krystian Mikolajczyk for providing part or all of their detectors and descriptors code. They are very grateful to David Lowe and Andrew Zisserman for comments on this manuscript.

References

[1] P.R. Beaudet, “Rotationally invariant image operators”, in *International Joint Conference on Pattern Recognition,* Kyoto, Japan, 1978, pp.579-583

[2] S. Belongie, J. Malik, J. Puzicha, “Shape matching and object recognition using shape contexts”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[3] J.Y. Bouguet, “Visual methods for three-dimensional modeling”, *PhD thesis, Caltech*, 1999.

[4] M. Brown and D.G. Lowe, “Recognising panoramas”, in *International Conference on Computer Vision*, 2003.

[5] G. Carneiro, A.D. Jepson, “Flexible spatial models for grouping local image features”, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004

[6] J.L. Crowley and A.C. Parker, “A representation for shape based on peaks and ridges in the difference of low-pass transform”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 156-168, 1984.

[7] W. Forstner, “A feature based correspondence algorithms for image matching”, *Intl. Arch. Photogrammetry and Remote Sensing*, vol. 24, pp 160-166, 1986.

[8] F. Fraundorfer and H. Bischof “Evaluation of local detectors on non-planar scenes”, *OAGM/AAPR workshop, Austrian Association for Pattern Recognition*, 2004.

[9] W. Freeman and E. Adelson, “The design and use of steerable filters”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891-906, 1991.

[10] C. Harris and M. Stephens, “A combined corner and edge detector”, in *Alvey Vision Conference*, 147-151,1988.

[11] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision”, *Cambridge editor*, 2000.

[12] T. Kadir, A. Zisserman and M. Brady “An affine invariant salient Region Detector”, in *European Conference on Computer Vision* 228-241, 2004.

[13] Y.Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors”, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[14] B. Leibe, E. Seemann, and B. Schiele, “Pedestrian detection in crowded scenes”. *In IEEE Conference on Computer Vision and Pattern Recognition*, pp.878-885, 2005.

[15] T. Lindeberg, “Scale-space theory: a basic tool for analising structures at different scales”, *Journal of Applied Statistics*, 21(2), pp.225-270, 1994.

[16] B.D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision”, in *7th International Joint Conference on Artificial Intelligence*, pp.674679,1981.

[17] T.Lindeberg and J.Garding, “Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure”, *Image and Vision Computing*, 15(6):415-434, 1997.

[18] D.G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, 60(2):91-110, 2004.

[19] J. Matas, O. Chum, M. Urban, T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”, *British Machine Vision Conference*, 384-393, 2002.

[20] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detector”, *European Conference on Computer Vision*, 2002.

[21] K. Mikolajczyk et al., “A comparison of affine region detectors”, *submitted, International Journal of Computer Vision*, 2004.

[22] K. Mikolajczyk, B. Leibe, B. Schiele, “Local features for object class recognition”, in *International Conference on Computer Vision*, 2005.

[23] K. Mikolajczyk, C. Schmid. “A performance evaluation of local descriptors”, To appear, *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[24] P. Moreels and P. Perona, “Evaluation of features detectors and descriptors based on 3D objects”, in *International Conference on Computer Vision*, 2005

[25] P. Moreels and P. Perona, “Common-frame model for object recognition”, in *Neural Information Processing Systems*, 2004

[26] P. Moreels and P. Perona, “Probabilistic coarse-to-fine object recognition”, Technical report, 2005.

[27] F. Schaffalitzky and A. Zisserman, “Viewpoint invariant texture matching and wide baseline stereo”, *in International Conference on Computer Vision*, pp.636-643, 2001.

[28] C. Schmid and R. Mohr, “Local greyvalue invariants for image retrieval”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530-535, 1997.

[29] C. Schmid, R. Mohr and C. Bauckhage, “Evaluation of interest point detectors”, in *International Journal of Computer Vision*, 37(2):151-172,2000.

[30] S. Se, D.G. Lowe and J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks”, in *International Journal of Robotics Research*, 21(8)735-738,2002.

[31] J.G.Semple and G.T.Kneebone, “Algebraic projective geometry”,Oxford Science Publication, 1952.

[32] A. Shashua and M. Werman, “On the trilinear tensor of three perspective views and its underlying geomtry”, *International Conference on Computer Vision*, 1995.

[33] J.Shi and C.Tomasi “Good features to track”, *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.

[34] T. Tuytelaars and L. Van Gool, “Wide baseline stereo matching based on local affinely invariant regions”, in *British Machine Vision Conference*, 2000.

[35] T. Tuytelaars and L. Van Gool, “Matching widely separated views based on affine invariant regions”, *In International Journal on Computer Vision*, vol. 59(1), pp.61-85, 2004.

[36] L.J. van Vliet, I.T. Young and P.W.Verbeek, “Recursive gaussian derivative filters”, in *International Conference on Pattern Recognition*, pp.509-514, 1998.

[37] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.