

Image/Video Classification

- D.G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision, 2004
- K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, *Evaluating Color Descriptors for Object and Scene Recognition*, IEEE transactions on Pattern Analysis and Machine Intelligence, 2009.

Distinctive Image Features from Scale-Invariant Keypoints

David G. Lowe

Computer Science Department
University of British Columbia
Vancouver, B.C., Canada
lowe@cs.ubc.ca

January 5, 2004

Abstract

This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

1 Introduction

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. This paper describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition.

The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. Following are the major stages of computation used to generate the set of image features:

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.
2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.
3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

This approach has been named the Scale Invariant Feature Transform (SIFT), as it transforms image data into scale-invariant coordinates relative to local features.

An important aspect of this approach is that it generates large numbers of features that densely cover the image over the full range of scales and locations. A typical image of size 500x500 pixels will give rise to about 2000 stable features (although this number depends on both image content and choices for various parameters). The quantity of features is particularly important for object recognition, where the ability to detect small objects in cluttered backgrounds requires that at least 3 features be correctly matched from each object for reliable identification.

For image matching and recognition, SIFT features are first extracted from a set of reference images and stored in a database. A new image is matched by individually comparing each feature from the new image to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors. This paper will discuss fast nearest-neighbor algorithms that can perform this computation rapidly against large databases.

The keypoint descriptors are highly distinctive, which allows a single feature to find its correct match with good probability in a large database of features. However, in a cluttered

image, many features from the background will not have any correct match in the database, giving rise to many false matches in addition to the correct ones. The correct matches can be filtered from the full set of matches by identifying subsets of keypoints that agree on the object and its location, scale, and orientation in the new image. The probability that several features will agree on these parameters by chance is much lower than the probability that any individual feature match will be in error. The determination of these consistent clusters can be performed rapidly by using an efficient hash table implementation of the generalized Hough transform.

Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed verification. First, a least-squared estimate is made for an affine approximation to the object pose. Any other image features consistent with this pose are identified, and outliers are discarded. Finally, a detailed computation is made of the probability that a particular set of features indicates the presence of an object, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

2 Related research

The development of image matching by using a set of local interest points can be traced back to the work of Moravec (1981) on stereo matching using a corner detector. The Moravec detector was improved by Harris and Stephens (1988) to make it more repeatable under small image variations and near edges. Harris also showed its value for efficient motion tracking and 3D structure from motion recovery (Harris, 1992), and the Harris corner detector has since been widely used for many other image matching tasks. While these feature detectors are usually called corner detectors, they are not selecting just corners, but rather any image location that has large gradients in all directions at a predetermined scale.

The initial applications were to stereo and short-range motion tracking, but the approach was later extended to more difficult problems. Zhang *et al.* (1995) showed that it was possible to match Harris corners over a large image range by using a correlation window around each corner to select likely matches. Outliers were then removed by solving for a fundamental matrix describing the geometric constraints between the two views of rigid scene and removing matches that did not agree with the majority solution. At the same time, a similar approach was developed by Torr (1995) for long-range motion matching, in which geometric constraints were used to remove outliers for rigid objects moving within an image.

The ground-breaking work of Schmid and Mohr (1997) showed that invariant local feature matching could be extended to general image recognition problems in which a feature was matched against a large database of images. They also used Harris corners to select interest points, but rather than matching with a correlation window, they used a rotationally invariant descriptor of the local image region. This allowed features to be matched under arbitrary orientation change between the two images. Furthermore, they demonstrated that multiple feature matches could accomplish general recognition under occlusion and clutter by identifying consistent clusters of matched features.

The Harris corner detector is very sensitive to changes in image scale, so it does not provide a good basis for matching images of different sizes. Earlier work by the author (Lowe, 1999) extended the local feature approach to achieve scale invariance. This work also described a new local descriptor that provided more distinctive features while being less

sensitive to local image distortions such as 3D viewpoint change. This current paper provides a more in-depth development and analysis of this earlier work, while also presenting a number of improvements in stability and feature invariance.

There is a considerable body of previous research on identifying representations that are stable under scale change. Some of the first work in this area was by Crowley and Parker (1984), who developed a representation that identified peaks and ridges in scale space and linked these into a tree structure. The tree structure could then be matched between images with arbitrary scale change. More recent work on graph-based matching by Shokoufandeh, Marsic and Dickinson (1999) provides more distinctive feature descriptors using wavelet coefficients. The problem of identifying an appropriate and consistent scale for feature detection has been studied in depth by Lindeberg (1993, 1994). He describes this as a problem of scale selection, and we make use of his results below.

Recently, there has been an impressive body of work on extending local features to be invariant to full affine transformations (Baumberg, 2000; Tuytelaars and Van Gool, 2000; Mikolajczyk and Schmid, 2002; Schaffalitzky and Zisserman, 2002; Brown and Lowe, 2002). This allows for invariant matching to features on a planar surface under changes in orthographic 3D projection, in most cases by resampling the image in a local affine frame. However, none of these approaches are yet fully affine invariant, as they start with initial feature scales and locations selected in a non-affine-invariant manner due to the prohibitive cost of exploring the full affine space. The affine frames are also more sensitive to noise than those of the scale-invariant features, so in practice the affine features have lower repeatability than the scale-invariant features unless the affine distortion is greater than about a 40 degree tilt of a planar surface (Mikolajczyk, 2002). Wider affine invariance may not be important for many applications, as training views are best taken at least every 30 degrees rotation in viewpoint (meaning that recognition is within 15 degrees of the closest training view) in order to capture non-planar changes and occlusion effects for 3D objects.

While the method to be presented in this paper is not fully affine invariant, a different approach is used in which the local descriptor allows relative feature positions to shift significantly with only small changes in the descriptor. This approach not only allows the descriptors to be reliably matched across a considerable range of affine distortion, but it also makes the features more robust against changes in 3D viewpoint for non-planar surfaces. Other advantages include much more efficient feature extraction and the ability to identify larger numbers of features. On the other hand, affine invariance is a valuable property for matching planar surfaces under very large view changes, and further research should be performed on the best ways to combine this with non-planar 3D viewpoint invariance in an efficient and stable manner.

Many other feature types have been proposed for use in recognition, some of which could be used in addition to the features described in this paper to provide further matches under differing circumstances. One class of features are those that make use of image contours or region boundaries, which should make them less likely to be disrupted by cluttered backgrounds near object boundaries. Matas *et al.*, (2002) have shown that their maximally-stable extremal regions can produce large numbers of matching features with good stability. Mikolajczyk *et al.*, (2003) have developed a new descriptor that uses local edges while ignoring unrelated nearby edges, providing the ability to find stable features even near the boundaries of narrow shapes superimposed on background clutter. Nelson and Selinger (1998) have shown good results with local features based on groupings of image contours. Similarly,

Pope and Lowe (2000) used features based on the hierarchical grouping of image contours, which are particularly useful for objects lacking detailed texture.

The history of research on visual recognition contains work on a diverse set of other image properties that can be used as feature measurements. Carneiro and Jepson (2002) describe phase-based local features that represent the phase rather than the magnitude of local spatial frequencies, which is likely to provide improved invariance to illumination. Schiele and Crowley (2000) have proposed the use of multidimensional histograms summarizing the distribution of measurements within image regions. This type of feature may be particularly useful for recognition of textured objects with deformable shapes. Basri and Jacobs (1997) have demonstrated the value of extracting local region boundaries for recognition. Other useful properties to incorporate include color, motion, figure-ground discrimination, region shape descriptors, and stereo depth cues. The local feature approach can easily incorporate novel feature types because extra features contribute to robustness when they provide correct matches, but otherwise do little harm other than their cost of computation. Therefore, future systems are likely to combine many feature types.

3 Detection of scale-space extrema

As described in the introduction, we will detect keypoints using a cascade filtering approach that uses efficient algorithms to identify candidate locations that are then examined in further detail. The first stage of keypoint detection is to identify locations and scales that can be repeatably assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space (Witkin, 1983).

It has been shown by Koenderink (1984) and Lindeberg (1994) that under a variety of reasonable assumptions the only possible scale-space kernel is the Gaussian function. Therefore, the scale space of an image is defined as a function, $L(x, y, \sigma)$, that is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image, $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where $*$ is the convolution operation in x and y , and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

To efficiently detect stable keypoint locations in scale space, we have proposed (Lowe, 1999) using scale-space extrema in the difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k :

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \tag{1}$$

There are a number of reasons for choosing this function. First, it is a particularly efficient function to compute, as the smoothed images, L , need to be computed in any case for scale space feature description, and D can therefore be computed by simple image subtraction.

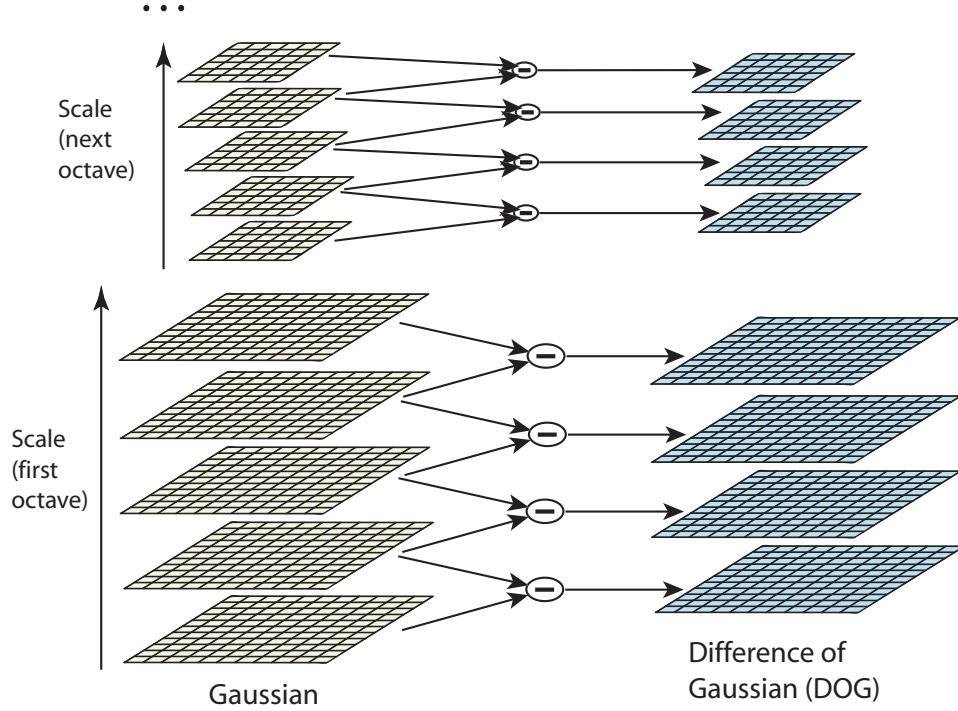


Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

In addition, the difference-of-Gaussian function provides a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2 \nabla^2 G$, as studied by Lindeberg (1994). Lindeberg showed that the normalization of the Laplacian with the factor σ^2 is required for true scale invariance. In detailed experimental comparisons, Mikołajczyk (2002) found that the maxima and minima of $\sigma^2 \nabla^2 G$ produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function.

The relationship between D and $\sigma^2 \nabla^2 G$ can be understood from the heat diffusion equation (parameterized in terms of σ rather than the more usual $t = \sigma^2$):

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

From this, we see that $\nabla^2 G$ can be computed from the finite difference approximation to $\partial G / \partial \sigma$, using the difference of nearby scales at $k\sigma$ and σ :

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

and therefore,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$$

This shows that when the difference-of-Gaussian function has scales differing by a constant factor it already incorporates the σ^2 scale normalization required for the scale-invariant

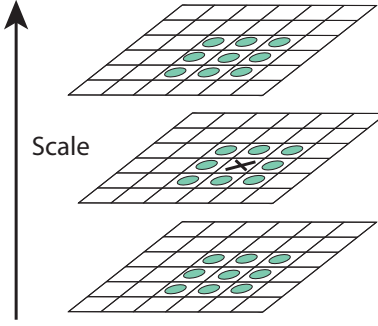


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Laplacian. The factor $(k - 1)$ in the equation is a constant over all scales and therefore does not influence extrema location. The approximation error will go to zero as k goes to 1, but in practice we have found that the approximation has almost no impact on the stability of extrema detection or localization for even significant differences in scale, such as $k = \sqrt{2}$.

An efficient approach to construction of $D(x, y, \sigma)$ is shown in Figure 1. The initial image is incrementally convolved with Gaussians to produce images separated by a constant factor k in scale space, shown stacked in the left column. We choose to divide each octave of scale space (i.e., doubling of σ) into an integer number, s , of intervals, so $k = 2^{1/s}$. We must produce $s + 3$ images in the stack of blurred images for each octave, so that final extrema detection covers a complete octave. Adjacent image scales are subtracted to produce the difference-of-Gaussian images shown on the right. Once a complete octave has been processed, we resample the Gaussian image that has twice the initial value of σ (it will be 2 images from the top of the stack) by taking every second pixel in each row and column. The accuracy of sampling relative to σ is no different than for the start of the previous octave, while computation is greatly reduced.

3.1 Local extrema detection

In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see Figure 2). It is selected only if it is larger than all of these neighbors or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks.

An important issue is to determine the frequency of sampling in the image and scale domains that is needed to reliably detect the extrema. Unfortunately, it turns out that there is no minimum spacing of samples that will detect all extrema, as the extrema can be arbitrarily close together. This can be seen by considering a white circle on a black background, which will have a single scale space maximum where the circular positive central region of the difference-of-Gaussian function matches the size and location of the circle. For a very elongated ellipse, there will be two maxima near each end of the ellipse. As the locations of maxima are a continuous function of the image, for some ellipse with intermediate elongation there will be a transition from a single maximum to two, with the maxima arbitrarily close to

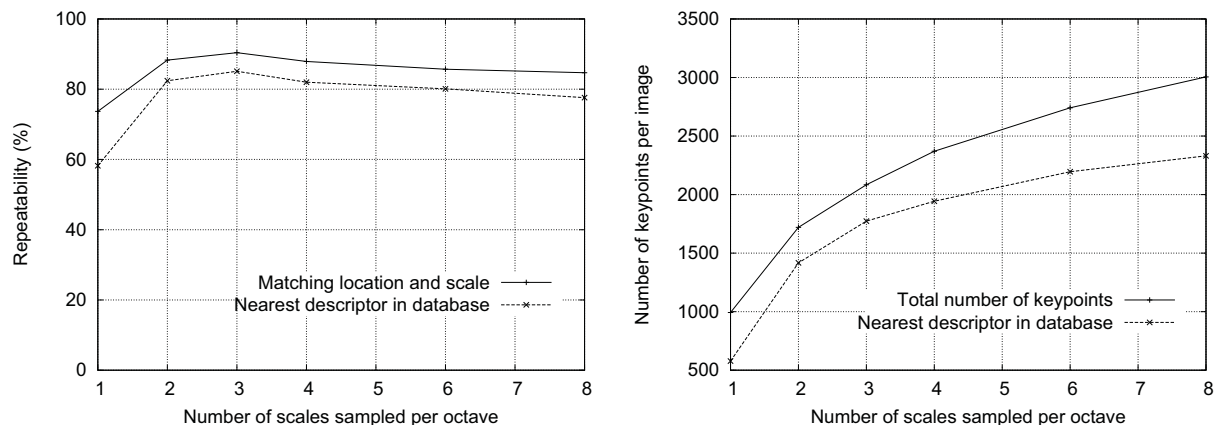


Figure 3: The top line of the first graph shows the percent of keypoints that are repeatably detected at the same location and scale in a transformed image as a function of the number of scales sampled per octave. The lower line shows the percent of keypoints that have their descriptors correctly matched to a large database. The second graph shows the total number of keypoints detected in a typical image as a function of the number of scale samples.

each other near the transition.

Therefore, we must settle for a solution that trades off efficiency with completeness. In fact, as might be expected and is confirmed by our experiments, extrema that are close together are quite unstable to small perturbations of the image. We can determine the best choices experimentally by studying a range of sampling frequencies and using those that provide the most reliable results under a realistic simulation of the matching task.

3.2 Frequency of sampling in scale

The experimental determination of sampling frequency that maximizes extrema stability is shown in Figures 3 and 4. These figures (and most other simulations in this paper) are based on a matching task using a collection of 32 real images drawn from a diverse range, including outdoor scenes, human faces, aerial photographs, and industrial images (the image domain was found to have almost no influence on any of the results). Each image was then subject to a range of transformations, including rotation, scaling, affine stretch, change in brightness and contrast, and addition of image noise. Because the changes were synthetic, it was possible to precisely predict where each feature in an original image should appear in the transformed image, allowing for measurement of correct repeatability and positional accuracy for each feature.

Figure 3 shows these simulation results used to examine the effect of varying the number of scales per octave at which the image function is sampled prior to extrema detection. In this case, each image was resampled following rotation by a random angle and scaling by a random amount between 0.2 of 0.9 times the original size. Keypoints from the reduced resolution image were matched against those from the original image so that the scales for all keypoints would be present in the matched image. In addition, 1% image noise was added, meaning that each pixel had a random number added from the uniform interval $[-0.01, 0.01]$ where pixel values are in the range $[0, 1]$ (equivalent to providing slightly less than 6 bits of accuracy for image pixels).

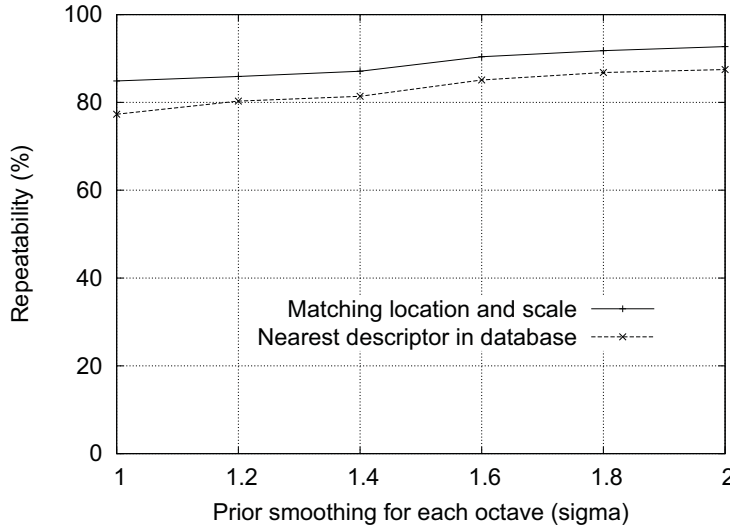


Figure 4: The top line in the graph shows the percent of keypoint locations that are repeatably detected in a transformed image as a function of the prior image smoothing for the first level of each octave. The lower line shows the percent of descriptors correctly matched against a large database.

The top line in the first graph of Figure 3 shows the percent of keypoints that are detected at a matching location and scale in the transformed image. For all examples in this paper, we define a matching scale as being within a factor of $\sqrt{2}$ of the correct scale, and a matching location as being within σ pixels, where σ is the scale of the keypoint (defined from equation (1) as the standard deviation of the smallest Gaussian used in the difference-of-Gaussian function). The lower line on this graph shows the number of keypoints that are correctly matched to a database of 40,000 keypoints using the nearest-neighbor matching procedure to be described in Section 6 (this shows that once the keypoint is repeatably located, it is likely to be useful for recognition and matching tasks). As this graph shows, the highest repeatability is obtained when sampling 3 scales per octave, and this is the number of scale samples used for all other experiments throughout this paper.

It might seem surprising that the repeatability does not continue to improve as more scales are sampled. The reason is that this results in many more local extrema being detected, but these extrema are on average less stable and therefore are less likely to be detected in the transformed image. This is shown by the second graph in Figure 3, which shows the average number of keypoints detected and correctly matched in each image. The number of keypoints rises with increased sampling of scales and the total number of correct matches also rises. Since the success of object recognition often depends more on the quantity of correctly matched keypoints, as opposed to their percentage correct matching, for many applications it will be optimal to use a larger number of scale samples. However, the cost of computation also rises with this number, so for the experiments in this paper we have chosen to use just 3 scale samples per octave.

To summarize, these experiments show that the scale-space difference-of-Gaussian function has a large number of extrema and that it would be very expensive to detect them all. Fortunately, we can detect the most stable and useful subset even with a coarse sampling of scales.

3.3 Frequency of sampling in the spatial domain

Just as we determined the frequency of sampling per octave of scale space, so we must determine the frequency of sampling in the image domain relative to the scale of smoothing. Given that extrema can be arbitrarily close together, there will be a similar trade-off between sampling frequency and rate of detection. Figure 4 shows an experimental determination of the amount of prior smoothing, σ , that is applied to each image level before building the scale space representation for an octave. Again, the top line is the repeatability of keypoint detection, and the results show that the repeatability continues to increase with σ . However, there is a cost to using a large σ in terms of efficiency, so we have chosen to use $\sigma = 1.6$, which provides close to optimal repeatability. This value is used throughout this paper and was used for the results in Figure 3.

Of course, if we pre-smooth the image before extrema detection, we are effectively discarding the highest spatial frequencies. Therefore, to make full use of the input, the image can be expanded to create more sample points than were present in the original. We double the size of the input image using linear interpolation prior to building the first level of the pyramid. While the equivalent operation could effectively have been performed by using sets of subpixel-offset filters on the original image, the image doubling leads to a more efficient implementation. We assume that the original image has a blur of at least $\sigma = 0.5$ (the minimum needed to prevent significant aliasing), and that therefore the doubled image has $\sigma = 1.0$ relative to its new pixel spacing. This means that little additional smoothing is needed prior to creation of the first octave of scale space. The image doubling increases the number of stable keypoints by almost a factor of 4, but no significant further improvements were found with a larger expansion factor.

4 Accurate keypoint localization

Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

The initial implementation of this approach (Lowe, 1999) simply located keypoints at the location and scale of the central sample point. However, recently Brown has developed a method (Brown and Lowe, 2002) for fitting a 3D quadratic function to the local sample points to determine the interpolated location of the maximum, and his experiments showed that this provides a substantial improvement to matching and stability. His approach uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (2)$$

where D and its derivatives are evaluated at the sample point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to \mathbf{x} and setting it to zero, giving

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}. \quad (3)$$

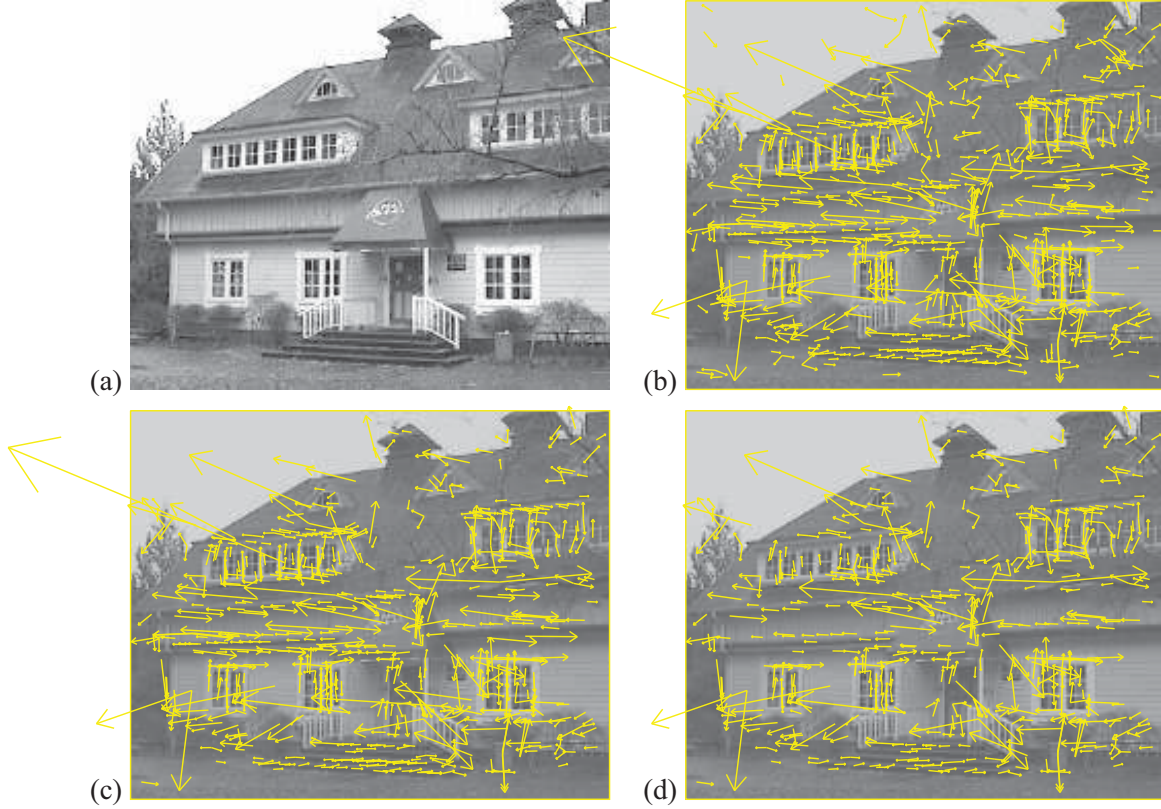


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

As suggested by Brown, the Hessian and derivative of D are approximated by using differences of neighboring sample points. The resulting 3x3 linear system can be solved with minimal cost. If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, then it means that the extremum lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed instead about that point. The final offset $\hat{\mathbf{x}}$ is added to the location of its sample point to get the interpolated estimate for the location of the extremum.

The function value at the extremum, $D(\hat{\mathbf{x}})$, is useful for rejecting unstable extrema with low contrast. This can be obtained by substituting equation (3) into (2), giving

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}.$$

For the experiments in this paper, all extrema with a value of $|D(\hat{\mathbf{x}})|$ less than 0.03 were discarded (as before, we assume image pixel values in the range $[0,1]$).

Figure 5 shows the effects of keypoint selection on a natural image. In order to avoid too much clutter, a low-resolution 233 by 189 pixel image is used and keypoints are shown as vectors giving the location, scale, and orientation of each keypoint (orientation assignment is described below). Figure 5 (a) shows the original image, which is shown at reduced contrast behind the subsequent figures. Figure 5 (b) shows the 832 keypoints at all detected maxima

and minima of the difference-of-Gaussian function, while (c) shows the 729 keypoints that remain following removal of those with a value of $|D(\hat{\mathbf{x}})|$ less than 0.03. Part (d) will be explained in the following section.

4.1 Eliminating edge responses

For stability, it is not sufficient to reject keypoints with low contrast. The difference-of-Gaussian function will have a strong response along edges, even if the location along the edge is poorly determined and therefore unstable to small amounts of noise.

A poorly defined peak in the difference-of-Gaussian function will have a large principal curvature across the edge but a small one in the perpendicular direction. The principal curvatures can be computed from a 2x2 Hessian matrix, \mathbf{H} , computed at the location and scale of the keypoint:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (4)$$

The derivatives are estimated by taking differences of neighboring sample points.

The eigenvalues of \mathbf{H} are proportional to the principal curvatures of D . Borrowing from the approach used by Harris and Stephens (1988), we can avoid explicitly computing the eigenvalues, as we are only concerned with their ratio. Let α be the eigenvalue with the largest magnitude and β be the smaller one. Then, we can compute the sum of the eigenvalues from the trace of \mathbf{H} and their product from the determinant:

$$\begin{aligned} \text{Tr}(\mathbf{H}) &= D_{xx} + D_{yy} = \alpha + \beta, \\ \text{Det}(\mathbf{H}) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \end{aligned}$$

In the unlikely event that the determinant is negative, the curvatures have different signs so the point is discarded as not being an extremum. Let r be the ratio between the largest magnitude eigenvalue and the smaller one, so that $\alpha = r\beta$. Then,

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

which depends only on the ratio of the eigenvalues rather than their individual values. The quantity $(r + 1)^2/r$ is at a minimum when the two eigenvalues are equal and it increases with r . Therefore, to check that the ratio of principal curvatures is below some threshold, r , we only need to check

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r + 1)^2}{r}.$$

This is very efficient to compute, with less than 20 floating point operations required to test each keypoint. The experiments in this paper use a value of $r = 10$, which eliminates keypoints that have a ratio between the principal curvatures greater than 10. The transition from Figure 5 (c) to (d) shows the effects of this operation.

5 Orientation assignment

By assigning a consistent orientation to each keypoint based on local image properties, the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation. This approach contrasts with the orientation invariant descriptors of Schmid and Mohr (1997), in which each image property is based on a rotationally invariant measure. The disadvantage of that approach is that it limits the descriptors that can be used and discards image information by not requiring all measures to be based on a consistent rotation.

Following experimentation with a number of approaches to assigning a local orientation, the following approach was found to give the most stable results. The scale of the keypoint is used to select the Gaussian smoothed image, L , with the closest scale, so that all computations are performed in a scale-invariant manner. For each image sample, $L(x, y)$, at this scale, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, is precomputed using pixel differences:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint. The orientation histogram has 36 bins covering the 360 degree range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint.

Peaks in the orientation histogram correspond to dominant directions of local gradients. The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation. Therefore, for locations with multiple peaks of similar magnitude, there will be multiple keypoints created at the same location and scale but different orientations. Only about 15% of points are assigned multiple orientations, but these contribute significantly to the stability of matching. Finally, a parabola is fit to the 3 histogram values closest to each peak to interpolate the peak position for better accuracy.

Figure 6 shows the experimental stability of location, scale, and orientation assignment under differing amounts of image noise. As before the images are rotated and scaled by random amounts. The top line shows the stability of keypoint location and scale assignment. The second line shows the stability of matching when the orientation assignment is also required to be within 15 degrees. As shown by the gap between the top two lines, the orientation assignment remains accurate 95% of the time even after addition of $\pm 10\%$ pixel noise (equivalent to a camera providing less than 3 bits of precision). The measured variance of orientation for the correct matches is about 2.5 degrees, rising to 3.9 degrees for 10% noise. The bottom line in Figure 6 shows the final accuracy of correctly matching a keypoint descriptor to a database of 40,000 keypoints (to be discussed below). As this graph shows, the SIFT features are resistant to even large amounts of pixel noise, and the major cause of error is the initial location and scale detection.

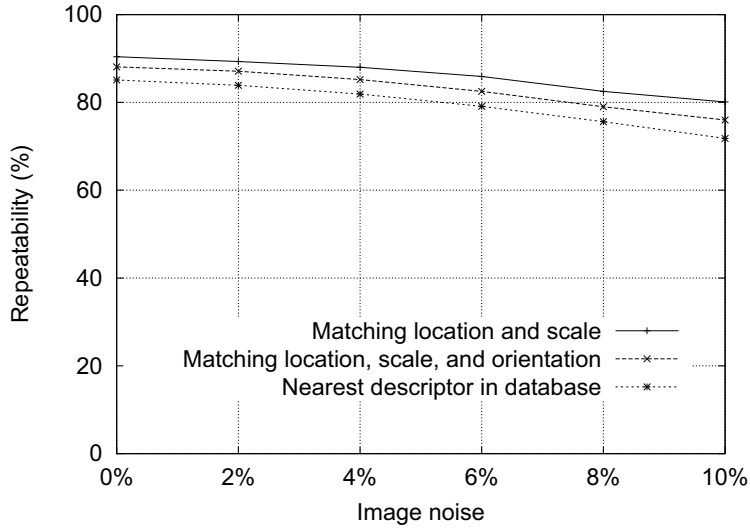


Figure 6: The top line in the graph shows the percent of keypoint locations and scales that are repeatably detected as a function of pixel noise. The second line shows the repeatability after also requiring agreement in orientation. The bottom line shows the final percent of descriptors correctly matched to a large database.

6 The local image descriptor

The previous operations have assigned an image location, scale, and orientation to each keypoint. These parameters impose a repeatable local 2D coordinate system in which to describe the local image region, and therefore provide invariance to these parameters. The next step is to compute a descriptor for the local image region that is highly distinctive yet is as invariant as possible to remaining variations, such as change in illumination or 3D viewpoint.

One obvious approach would be to sample the local image intensities around the keypoint at the appropriate scale, and to match these using a normalized correlation measure. However, simple correlation of image patches is highly sensitive to changes that cause misregistration of samples, such as affine or 3D viewpoint change or non-rigid deformations. A better approach has been demonstrated by Edelman, Intrator, and Poggio (1997). Their proposed representation was based upon a model of biological vision, in particular of complex neurons in primary visual cortex. These complex neurons respond to a gradient at a particular orientation and spatial frequency, but the location of the gradient on the retina is allowed to shift over a small receptive field rather than being precisely localized. Edelman *et al.* hypothesized that the function of these complex neurons was to allow for matching and recognition of 3D objects from a range of viewpoints. They have performed detailed experiments using 3D computer models of object and animal shapes which show that matching gradients while allowing for shifts in their position results in much better classification under 3D rotation. For example, recognition accuracy for 3D objects rotated in depth by 20 degrees increased from 35% for correlation of gradients to 94% using the complex cell model. Our implementation described below was inspired by this idea, but allows for positional shift using a different computational mechanism.

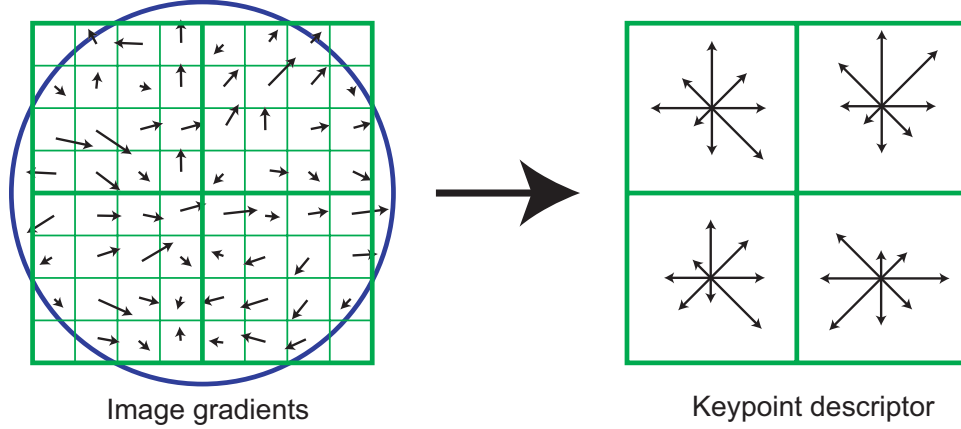


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples, whereas the experiments in this paper use 4×4 descriptors computed from a 16×16 sample array.

6.1 Descriptor representation

Figure 7 illustrates the computation of the keypoint descriptor. First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. For efficiency, the gradients are precomputed for all levels of the pyramid as described in Section 5. These are illustrated with small arrows at each sample location on the left side of Figure 7.

A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. This is illustrated with a circular window on the left side of Figure 7, although, of course, the weight falls off smoothly. The purpose of this Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far from the center of the descriptor, as these are most affected by misregistration errors.

The keypoint descriptor is shown on the right side of Figure 7. It allows for significant shift in gradient positions by creating orientation histograms over 4×4 sample regions. The figure shows eight directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts.

It is important to avoid all boundary affects in which the descriptor abruptly changes as a sample shifts smoothly from being within one histogram to another or from one orientation to another. Therefore, trilinear interpolation is used to distribute the value of each gradient sample into adjacent histogram bins. In other words, each entry into a bin is multiplied by a weight of $1 - d$ for each dimension, where d is the distance of the sample from the central value of the bin as measured in units of the histogram bin spacing.

The descriptor is formed from a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the arrows on the right side of Figure 7. The figure shows a 2×2 array of orientation histograms, whereas our experiments below show that the best results are achieved with a 4×4 array of histograms with 8 orientation bins in each. Therefore, the experiments in this paper use a $4 \times 4 \times 8 = 128$ element feature vector for each keypoint.

Finally, the feature vector is modified to reduce the effects of illumination change. First, the vector is normalized to unit length. A change in image contrast in which each pixel value is multiplied by a constant will multiply gradients by the same constant, so this contrast change will be canceled by vector normalization. A brightness change in which a constant is added to each image pixel will not affect the gradient values, as they are computed from pixel differences. Therefore, the descriptor is invariant to affine changes in illumination. However, non-linear illumination changes can also occur due to camera saturation or due to illumination changes that affect 3D surfaces with differing orientations by different amounts. These effects can cause a large change in relative magnitudes for some gradients, but are less likely to affect the gradient orientations. Therefore, we reduce the influence of large gradient magnitudes by thresholding the values in the unit feature vector to each be no larger than 0.2, and then renormalizing to unit length. This means that matching the magnitudes for large gradients is no longer as important, and that the distribution of orientations has greater emphasis. The value of 0.2 was determined experimentally using images containing differing illuminations for the same 3D objects.

6.2 Descriptor testing

There are two parameters that can be used to vary the complexity of the descriptor: the number of orientations, r , in the histograms, and the width, n , of the $n \times n$ array of orientation histograms. The size of the resulting descriptor vector is rn^2 . As the complexity of the descriptor grows, it will be able to discriminate better in a large database, but it will also be more sensitive to shape distortions and occlusion.

Figure 8 shows experimental results in which the number of orientations and size of the descriptor were varied. The graph was generated for a viewpoint transformation in which a planar surface is tilted by 50 degrees away from the viewer and 4% image noise is added. This is near the limits of reliable matching, as it is in these more difficult cases that descriptor performance is most important. The results show the percent of keypoints that find a correct match to the single closest neighbor among a database of 40,000 keypoints. The graph shows that a single orientation histogram ($n = 1$) is very poor at discriminating, but the results continue to improve up to a 4×4 array of histograms with 8 orientations. After that, adding more orientations or a larger descriptor can actually hurt matching by making the descriptor more sensitive to distortion. These results were broadly similar for other degrees of viewpoint change and noise, although in some simpler cases discrimination continued to improve (from already high levels) with 5×5 and higher descriptor sizes. Throughout this paper we use a 4×4 descriptor with 8 orientations, resulting in feature vectors with 128 dimensions. While the dimensionality of the descriptor may seem high, we have found that it consistently performs better than lower-dimensional descriptors on a range of matching tasks and that the computational cost of matching remains low when using the approximate nearest-neighbor methods described below.

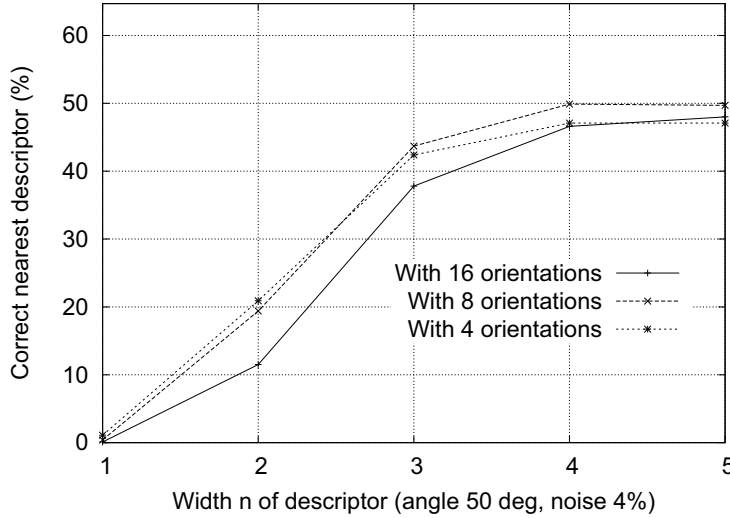


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and addition of 4% noise.

6.3 Sensitivity to affine change

The sensitivity of the descriptor to affine change is examined in Figure 9. The graph shows the reliability of keypoint location and scale selection, orientation assignment, and nearest-neighbor matching to a database as a function of rotation in depth of a plane away from a viewer. It can be seen that each stage of computation has reduced repeatability with increasing affine distortion, but that the final matching accuracy remains above 50% out to a 50 degree change in viewpoint.

To achieve reliable matching over a wider viewpoint angle, one of the affine-invariant detectors could be used to select and resample image regions, as discussed in Section 2. As mentioned there, none of these approaches is truly affine-invariant, as they all start from initial feature locations determined in a non-affine-invariant manner. In what appears to be the most affine-invariant method, Mikolajczyk (2002) has proposed and run detailed experiments with the Harris-affine detector. He found that its keypoint repeatability is below that given here out to about a 50 degree viewpoint angle, but that it then retains close to 40% repeatability out to an angle of 70 degrees, which provides better performance for extreme affine changes. The disadvantages are a much higher computational cost, a reduction in the number of keypoints, and poorer stability for small affine changes due to errors in assigning a consistent affine frame under noise. In practice, the allowable range of rotation for 3D objects is considerably less than for planar surfaces, so affine invariance is usually not the limiting factor in the ability to match across viewpoint change. If a wide range of affine invariance is desired, such as for a surface that is known to be planar, then a simple solution is to adopt the approach of Pritchard and Heidrich (2003) in which additional SIFT features are generated from 4 affine-transformed versions of the training image corresponding to 60 degree viewpoint changes. This allows for the use of standard SIFT features with no additional cost when processing the image to be recognized, but results in an increase in the size of the feature database by a factor of 3.

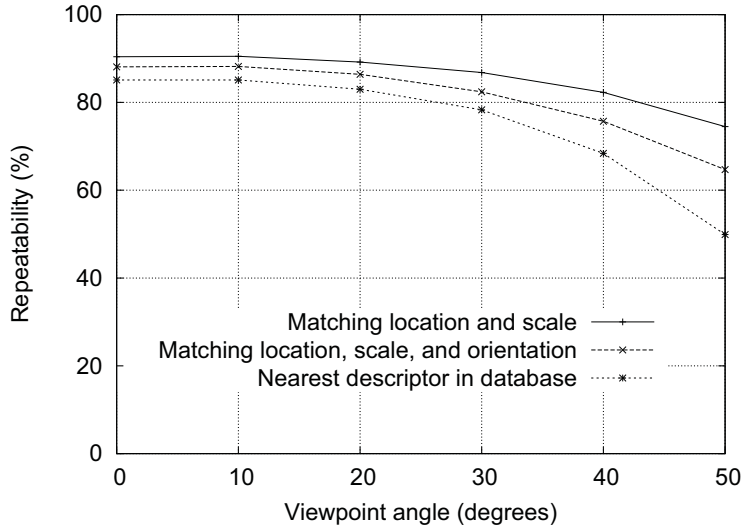


Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

6.4 Matching to large databases

An important remaining issue for measuring the distinctiveness of features is how the reliability of matching varies as a function of the number of features in the database being matched. Most of the examples in this paper are generated using a database of 32 images with about 40,000 keypoints. Figure 10 shows how the matching reliability varies as a function of database size. This figure was generated using a larger database of 112 images, with a viewpoint depth rotation of 30 degrees and 2% image noise in addition to the usual random image rotation and scale change.

The dashed line shows the portion of image features for which the nearest neighbor in the database was the correct match, as a function of database size shown on a logarithmic scale. The leftmost point is matching against features from only a single image while the rightmost point is selecting matches from a database of all features from the 112 images. It can be seen that matching reliability does decrease as a function of the number of distractors, yet all indications are that many correct matches will continue to be found out to very large database sizes.

The solid line is the percentage of keypoints that were identified at the correct matching location and orientation in the transformed image, so it is only these points that have any chance of having matching descriptors in the database. The reason this line is flat is that the test was run over the full database for each value, while only varying the portion of the database used for distractors. It is of interest that the gap between the two lines is small, indicating that matching failures are due more to issues with initial feature localization and orientation assignment than to problems with feature distinctiveness, even out to large database sizes.

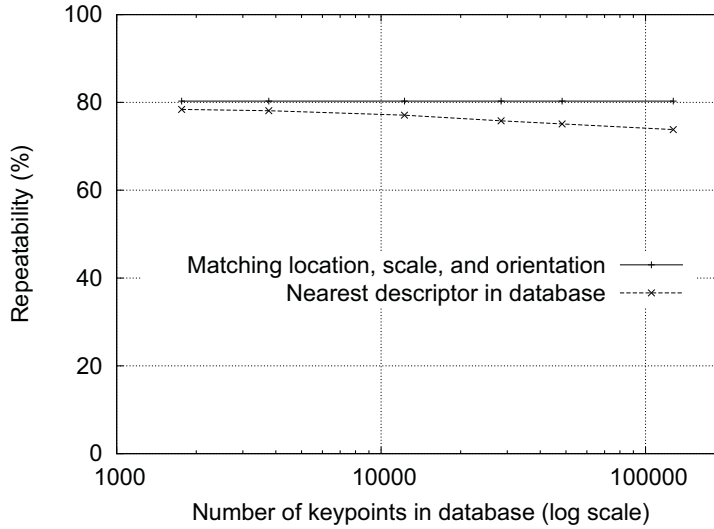


Figure 10: The dashed line shows the percent of keypoints correctly matched to a database as a function of database size (using a logarithmic scale). The solid line shows the percent of keypoints assigned the correct location, scale, and orientation. Images had random scale and rotation changes, an affine transform of 30 degrees, and image noise of 2% added prior to matching.

7 Application to object recognition

The major topic of this paper is the derivation of distinctive invariant keypoints, as described above. To demonstrate their application, we will now give a brief description of their use for object recognition in the presence of clutter and occlusion. More details on applications of these features to recognition are available in other papers (Lowe, 1999; Lowe, 2001; Se, Lowe and Little, 2002).

Object recognition is performed by first matching each keypoint independently to the database of keypoints extracted from training images. Many of these initial matches will be incorrect due to ambiguous features or features that arise from background clutter. Therefore, clusters of at least 3 features are first identified that agree on an object and its pose, as these clusters have a much higher probability of being correct than individual feature matches. Then, each cluster is checked by performing a detailed geometric fit to the model, and the result is used to accept or reject the interpretation.

7.1 Keypoint matching

The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector as was described in Section 6.

However, many features from an image will not have any correct match in the training database because they arise from background clutter or were not detected in the training images. Therefore, it would be useful to have a way to discard features that do not have any good match to the database. A global threshold on distance to the closest feature does not perform well, as some descriptors are much more discriminative than others. A more effective measure is obtained by comparing the distance of the closest neighbor to that of the

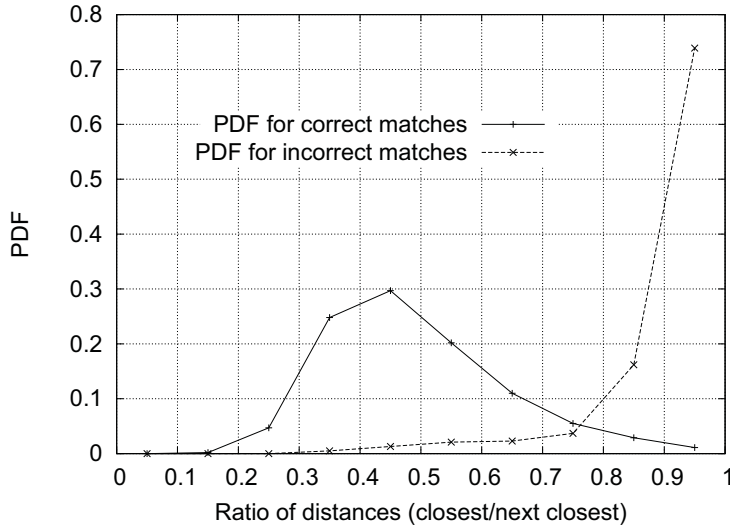


Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect.

second-closest neighbor. If there are multiple training images of the same object, then we define the second-closest neighbor as being the closest neighbor that is known to come from a different object than the first, such as by only using images known to contain different objects. This measure performs well because correct matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances due to the high dimensionality of the feature space. We can think of the second-closest match as providing an estimate of the density of false matches within this portion of the feature space and at the same time identifying specific instances of feature ambiguity.

Figure 11 shows the value of this measure for real image data. The probability density functions for correct and incorrect matches are shown in terms of the ratio of closest to second-closest neighbors of each keypoint. Matches for which the nearest neighbor was a correct match have a PDF that is centered at a much lower ratio than that for incorrect matches. For our object recognition implementation, we reject all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches. This figure was generated by matching images following random scale and orientation change, a depth rotation of 30 degrees, and addition of 2% image noise, against a database of 40,000 keypoints.

7.2 Efficient nearest neighbor indexing

No algorithms are known that can identify the exact nearest neighbors of points in high dimensional spaces that are any more efficient than exhaustive search. Our keypoint descriptor has a 128-dimensional feature vector, and the best algorithms, such as the k-d tree (Friedman *et al.*, 1977) provide no speedup over exhaustive search for more than about 10 dimensional spaces. Therefore, we have used an approximate algorithm, called the Best-Bin-First (BBF) algorithm (Beis and Lowe, 1997). This is approximate in the sense that it returns the closest

neighbor with high probability.

The BBF algorithm uses a modified search ordering for the k-d tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This priority search order was first examined by Arya and Mount (1993), and they provide further study of its computational properties in (Arya *et al.*, 1998). This search order requires the use of a heap-based priority queue for efficient determination of the search order. An approximate answer can be returned with low cost by cutting off further search after a specific number of the nearest bins have been explored. In our implementation, we cut off search after checking the first 200 nearest-neighbor candidates. For a database of 100,000 keypoints, this provides a speedup over exact nearest neighbor search by about 2 orders of magnitude yet results in less than a 5% loss in the number of correct matches. One reason the BBF algorithm works particularly well for this problem is that we only consider matches in which the nearest neighbor is less than 0.8 times the distance to the second-nearest neighbor (as described in the previous section), and therefore there is no need to exactly solve the most difficult cases in which many neighbors are at very similar distances.

7.3 Clustering with the Hough transform

To maximize the performance of object recognition for small or highly occluded objects, we wish to identify objects with the fewest possible number of feature matches. We have found that reliable recognition is possible with as few as 3 features. A typical image contains 2,000 or more features which may come from many different objects as well as background clutter. While the distance ratio test described in Section 7.1 will allow us to discard many of the false matches arising from background clutter, this does not remove matches from other valid objects, and we often still need to identify correct subsets of matches containing less than 1% inliers among 99% outliers. Many well-known robust fitting methods, such as RANSAC or Least Median of Squares, perform poorly when the percent of inliers falls much below 50%. Fortunately, much better performance can be obtained by clustering features in pose space using the Hough transform (Hough, 1962; Ballard, 1981; Grimson 1990).

The Hough transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with the feature. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. Each of our keypoints specifies 4 parameters: 2D location, scale, and orientation, and each matched keypoint in the database has a record of the keypoint's parameters relative to the training image in which it was found. Therefore, we can create a Hough transform entry predicting the model location, orientation, and scale from the match hypothesis. This prediction has large error bounds, as the similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object and also does not account for any non-rigid deformations. Therefore, we use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times the maximum projected training image dimension (using the predicted scale) for location. To avoid the problem of boundary effects in bin assignment, each keypoint match votes for the 2 closest bins in each dimension, giving a total of 16 entries for each hypothesis and further broadening the pose range.

In most implementations of the Hough transform, a multi-dimensional array is used to represent the bins. However, many of the potential bins will remain empty, and it is difficult to compute the range of possible bin values due to their mutual dependence (for example,

the dependency of location discretization on the selected scale). These problems can be avoided by using a pseudo-random hash function of the bin values to insert votes into a one-dimensional hash table, in which collisions are easily detected.

7.4 Solution for affine parameters

The Hough transform is used to identify all clusters with at least 3 entries in a bin. Each such cluster is then subject to a geometric verification procedure in which a least-squares solution is performed for the best affine projection parameters relating the training image to the new image.

An affine transformation correctly accounts for 3D rotation of a planar surface under orthographic projection, but the approximation can be poor for 3D rotation of non-planar objects. A more general solution would be to solve for the fundamental matrix (Luong and Faugeras, 1996; Hartley and Zisserman, 2000). However, a fundamental matrix solution requires at least 7 point matches as compared to only 3 for the affine solution and in practice requires even more matches for good stability. We would like to perform recognition with as few as 3 feature matches, so the affine solution provides a better starting point and we can account for errors in the affine approximation by allowing for large residual errors. If we imagine placing a sphere around an object, then rotation of the sphere by 30 degrees will move no point within the sphere by more than 0.25 times the projected diameter of the sphere. For the examples of typical 3D objects used in this paper, an affine solution works well given that we allow residual errors up to 0.25 times the maximum projected dimension of the object. A more general approach is given in (Brown and Lowe, 2002), in which the initial solution is based on a similarity transform, which then progresses to solution for the fundamental matrix in those cases in which a sufficient number of matches are found.

The affine transformation of a model point $[x \ y]^T$ to an image point $[u \ v]^T$ can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

where the model translation is $[t_x \ t_y]^T$ and the affine rotation, scale, and stretch are represented by the m_i parameters.

We wish to solve for the transformation parameters, so the equation above can be rewritten to gather the unknowns into a column vector:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

This equation shows a single match, but any number of further matches can be added, with each match contributing two more rows to the first and last matrix. At least 3 matches are needed to provide a solution.

We can write this linear system as

$$\mathbf{Ax} = \mathbf{b}$$



Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

The least-squares solution for the parameters \mathbf{x} can be determined by solving the corresponding normal equations,

$$\mathbf{x} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b},$$

which minimizes the sum of the squares of the distances from the projected model locations to the corresponding image locations. This least-squares approach could readily be extended to solving for 3D pose and internal parameters of articulated and flexible objects (Lowe, 1991).

Outliers can now be removed by checking for agreement between each image feature and the model. Given the more accurate least-squares solution, we now require each match to agree within half the error range that was used for the parameters in the Hough transform bins. If fewer than 3 points remain after discarding outliers, then the match is rejected. As outliers are discarded, the least-squares solution is re-solved with the remaining points, and the process iterated. In addition, a top-down matching phase is used to add any further matches that agree with the projected model position. These may have been missed from the Hough transform bin due to the similarity transform approximation or other errors.

The final decision to accept or reject a model hypothesis is based on a detailed probabilistic model given in a previous paper (Lowe, 2001). This method first computes the expected number of false matches to the model pose, given the projected size of the model, the number of features within the region, and the accuracy of the fit. A Bayesian analysis then gives the probability that the object is present based on the actual number of matching features found. We accept a model if the final probability for a correct interpretation is greater than 0.98. For objects that project to small regions of an image, 3 features may be sufficient for reliable recognition. For large objects covering most of a heavily textured image, the expected number of false matches is higher, and as many as 10 feature matches may be necessary.

8 Recognition examples

Figure 12 shows an example of object recognition for a cluttered and occluded image containing 3D objects. The training images of a toy train and a frog are shown on the left.

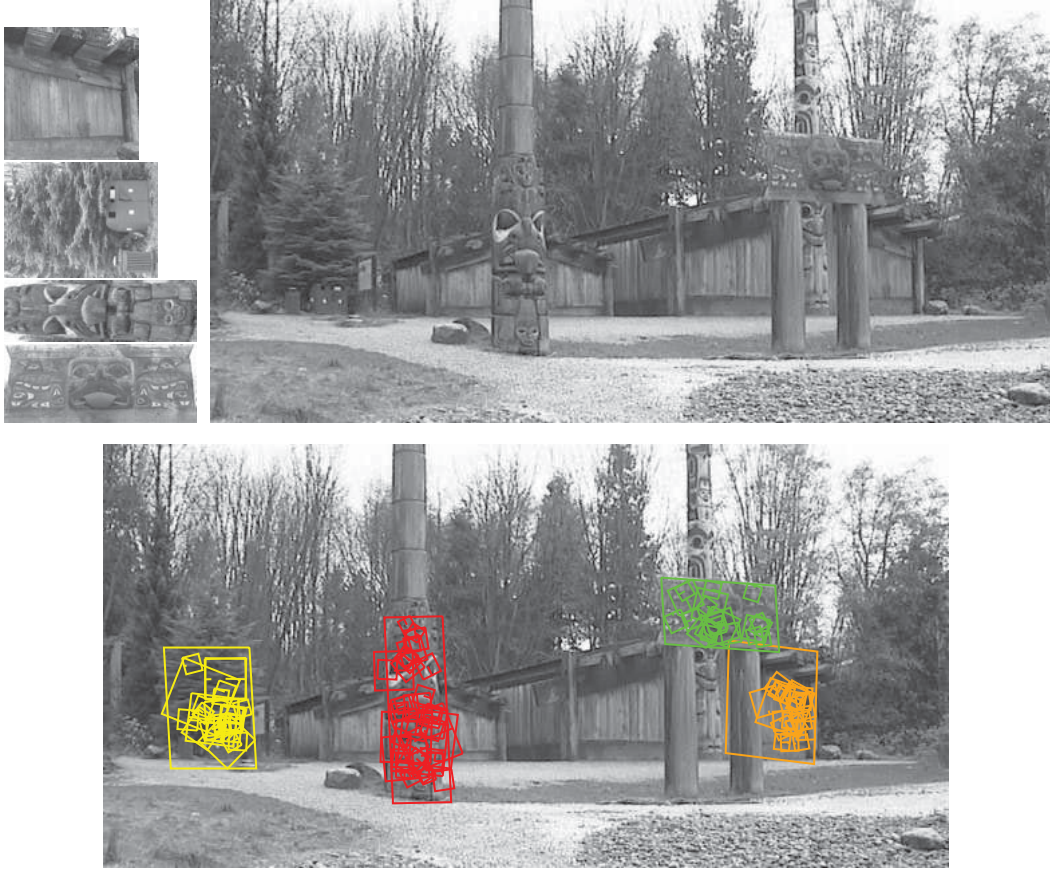


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.

The middle image (of size 600x480 pixels) contains instances of these objects hidden behind others and with extensive background clutter so that detection of the objects may not be immediate even for human vision. The image on the right shows the final correct identification superimposed on a reduced contrast version of the image. The keypoints that were used for recognition are shown as squares with an extra line to indicate orientation. The sizes of the squares correspond to the image regions used to construct the descriptor. An outer parallelogram is also drawn around each instance of recognition, with its sides corresponding to the boundaries of the training images projected under the final affine transformation determined during recognition.

Another potential application of the approach is to place recognition, in which a mobile device or vehicle could identify its location by recognizing familiar locations. Figure 13 gives an example of this application, in which training images are taken of a number of locations. As shown on the upper left, these can even be of such seemingly non-distinctive items as a wooden wall or a tree with trash bins. The test image (of size 640 by 315 pixels) on the upper right was taken from a viewpoint rotated about 30 degrees around the scene from the original positions, yet the training image locations are easily recognized.

All steps of the recognition process can be implemented efficiently, so the total time to recognize all objects in Figures 12 or 13 is less than 0.3 seconds on a 2GHz Pentium 4 processor. We have implemented these algorithms on a laptop computer with attached video camera, and have tested them extensively over a wide range of conditions. In general, textured planar surfaces can be identified reliably over a rotation in depth of up to 50 degrees in any direction and under almost any illumination conditions that provide sufficient light and do not produce excessive glare. For 3D objects, the range of rotation in depth for reliable recognition is only about 30 degrees in any direction and illumination change is more disruptive. For these reasons, 3D object recognition is best performed by integrating features from multiple views, such as with local feature view clustering (Lowe, 2001).

These keypoints have also been applied to the problem of robot localization and mapping, which has been presented in detail in other papers (Se, Lowe and Little, 2001). In this application, a trinocular stereo system is used to determine 3D estimates for keypoint locations. Keypoints are used only when they appear in all 3 images with consistent disparities, resulting in very few outliers. As the robot moves, it localizes itself using feature matches to the existing 3D map, and then incrementally adds features to the map while updating their 3D positions using a Kalman filter. This provides a robust and accurate solution to the problem of robot localization in unknown environments. This work has also addressed the problem of place recognition, in which a robot can be switched on and recognize its location anywhere within a large map (Se, Lowe and Little, 2002), which is equivalent to a 3D implementation of object recognition.

9 Conclusions

The SIFT keypoints described in this paper are particularly useful due to their distinctiveness, which enables the correct match for a keypoint to be selected from a large database of other keypoints. This distinctiveness is achieved by assembling a high-dimensional vector representing the image gradients within a local region of the image. The keypoints have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. Large numbers of keypoints can be extracted from typical images, which leads to robustness in extracting small objects among clutter. The fact that keypoints are detected over a complete range of scales means that small local features are available for matching small and highly occluded objects, while large keypoints perform well for images subject to noise and blur. Their computation is efficient, so that several thousand keypoints can be extracted from a typical image with near real-time performance on standard PC hardware.

This paper has also presented methods for using the keypoints for object recognition. The approach we have described uses approximate nearest-neighbor lookup, a Hough transform for identifying clusters that agree on object pose, least-squares pose determination, and final verification. Other potential applications include view matching for 3D reconstruction, motion tracking and segmentation, robot localization, image panorama assembly, epipolar calibration, and any others that require identification of matching locations between images.

There are many directions for further research in deriving invariant and distinctive image features. Systematic testing is needed on data sets with full 3D viewpoint and illumination changes. The features described in this paper use only a monochrome intensity image, so further distinctiveness could be derived from including illumination-invariant color descriptors

(Funt and Finlayson, 1995; Brown and Lowe, 2002). Similarly, local texture measures appear to play an important role in human vision and could be incorporated into feature descriptors in a more general form than the single spatial frequency used by the current descriptors. An attractive aspect of the invariant local feature approach to matching is that there is no need to select just one feature type, and the best results are likely to be obtained by using many different features, all of which can contribute useful matches and improve overall robustness.

Another direction for future research will be to individually learn features that are suited to recognizing particular objects categories. This will be particularly important for generic object classes that must cover a broad range of possible appearances. The research of Weber, Welling, and Perona (2000) and Fergus, Perona, and Zisserman (2003) has shown the potential of this approach by learning small sets of local features that are suited to recognizing generic classes of objects. In the long term, feature sets are likely to contain both prior and learned features that will be used according to the amount of training data that has been available for various object classes.

Acknowledgments

I would particularly like to thank Matthew Brown, who has suggested numerous improvements to both the content and presentation of this paper and whose own work on feature localization and invariance has contributed to this approach. In addition, I would like to thank many others for their valuable suggestions, including Stephen Se, Jim Little, Krystian Mikolajczyk, Cordelia Schmid, Tony Lindeberg, and Andrew Zisserman. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and through the Institute for Robotics and Intelligent Systems (IRIS) Network of Centres of Excellence.

References

- Arya, S., and Mount, D.M. 1993. Approximate nearest neighbor queries in fixed dimensions. In *Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pp. 271-280.
- Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., and Wu, A.Y. 1998. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891-923.
- Ballard, D.H. 1981. Generalizing the Hough transform to detect arbitrary patterns. *Pattern Recognition*, 13(2):111-122.
- Basri, R., and Jacobs, D.W. 1997. Recognition using region correspondences. *International Journal of Computer Vision*, 25(2):145-166.
- Baumberg, A. 2000. Reliable feature matching across widely separated views. In *Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, pp. 774-781.
- Beis, J. and Lowe, D.G. 1997. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Conference on Computer Vision and Pattern Recognition*, Puerto Rico, pp. 1000-1006.
- Brown, M. and Lowe, D.G. 2002. Invariant features from interest point groups. In *British Machine Vision Conference*, Cardiff, Wales, pp. 656-665.
- Carneiro, G., and Jepson, A.D. 2002. Phase-based local features. In *European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, pp. 282-296.
- Crowley, J. L. and Parker, A.C. 1984. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(2):156-170.

- Edelman, S., Intrator, N. and Poggio, T. 1997. Complex cells and object recognition. Unpublished manuscript: <http://kybele.psych.cornell.edu/~edelman/archive.html>
- Fergus, R., Perona, P., and Zisserman, A. 2003. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, pp. 264-271.
- Friedman, J.H., Bentley, J.L. and Finkel, R.A. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209-226.
- Funt, B.V. and Finlayson, G.D. 1995. Color constant color indexing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(5):522-529.
- Grimson, E. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*, The MIT Press: Cambridge, MA.
- Harris, C. 1992. Geometry from visual motion. In *Active Vision*, A. Blake and A. Yuille (Eds.), MIT Press, pp. 263-284.
- Harris, C. and Stephens, M. 1988. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, Manchester, UK, pp. 147-151.
- Hartley, R. and Zisserman, A. 2000. *Multiple view geometry in computer vision*, Cambridge University Press: Cambridge, UK.
- Hough, P.V.C. 1962. *Method and means for recognizing complex patterns*. U.S. Patent 3069654.
- Koenderink, J.J. 1984. The structure of images. *Biological Cybernetics*, 50:363-396.
- Lindeberg, T. 1993. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283-318.
- Lindeberg, T. 1994. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224-270.
- Lowe, D.G. 1991. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441-450.
- Lowe, D.G. 1999. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, Corfu, Greece, pp. 1150-1157.
- Lowe, D.G. 2001. Local feature view clustering for 3D object recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, pp. 682-688.
- Luong, Q.T., and Faugeras, O.D. 1996. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43-76.
- Matas, J., Chum, O., Urban, M., and Pajdla, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, Cardiff, Wales, pp. 384-393.
- Mikolajczyk, K. 2002. *Detection of local features invariant to affine transformations*, Ph.D. thesis, Institut National Polytechnique de Grenoble, France.
- Mikolajczyk, K., and Schmid, C. 2002. An affine invariant interest point detector. In *European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, pp. 128-142.
- Mikolajczyk, K., Zisserman, A., and Schmid, C. 2003. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, Norwich, U.K.
- Moravec, H. 1981. Rover visual obstacle avoidance. In *International Joint Conference on Artificial Intelligence*, Vancouver, Canada, pp. 785-790.
- Nelson, R.C., and Selinger, A. 1998. Large-scale tests of a keyed, appearance-based 3-D object recognition system. *Vision Research*, 38(15):2469-88.
- Pope, A.R., and Lowe, D.G. 2000. Probabilistic models of appearance for 3-D object recognition. *International Journal of Computer Vision*, 40(2):149-167.

- Pritchard, D., and Heidrich, W. 2003. Cloth motion capture. *Computer Graphics Forum (Eurographics 2003)*, 22(3):263-271.
- Schaffalitzky, F., and Zisserman, A. 2002. Multi-view matching for unordered image sets, or ‘How do I organize my holiday snaps?’” In *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 414-431.
- Schiele, B., and Crowley, J.L. 2000. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31-50.
- Schmid, C., and Mohr, R. 1997. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530-534.
- Se, S., Lowe, D.G., and Little, J. 2001. Vision-based mobile robot localization and mapping using scale-invariant features. In *International Conference on Robotics and Automation*, Seoul, Korea, pp. 2051-58.
- Se, S., Lowe, D.G., and Little, J. 2002. Global localization using distinctive visual features. In *International Conference on Intelligent Robots and Systems, IROS 2002*, Lausanne, Switzerland, pp. 226-231.
- Shokoufandeh, A., Marsic, I., and Dickinson, S.J. 1999. View-based object recognition using saliency maps. *Image and Vision Computing*, 17:445-460.
- Torr, P. 1995. *Motion Segmentation and Outlier Detection*, Ph.D. Thesis, Dept. of Engineering Science, University of Oxford, UK.
- Tuytelaars, T., and Van Gool, L. 2000. Wide baseline stereo based on local, affinely invariant regions. In *British Machine Vision Conference*, Bristol, UK, pp. 412-422.
- Weber, M., Welling, M. and Perona, P. 2000. Unsupervised learning of models for recognition. In *European Conference on Computer Vision*, Dublin, Ireland, pp. 18-32.
- Witkin, A.P. 1983. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp. 1019-1022.
- Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.T. 1995. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87-119.

Evaluating Color Descriptors for Object and Scene Recognition

Koen E. A. van de Sande, *Student Member, IEEE*, Theo Gevers, *Member, IEEE*,
and Cees G. M. Snoek, *Member, IEEE*

Abstract—Image category recognition is important to access visual information on the level of objects and scene types. So far, intensity-based descriptors have been widely used for feature extraction at salient points. To increase illumination invariance and discriminative power, color descriptors have been proposed. Because many different descriptors exist, a structured overview is required of color invariant descriptors in the context of image category recognition.

Therefore, this paper studies the invariance properties and the distinctiveness of color descriptors¹ in a structured way. The analytical invariance properties of color descriptors are explored, using a taxonomy based on invariance properties with respect to photometric transformations, and tested experimentally using a dataset with known illumination conditions. In addition, the distinctiveness of color descriptors is assessed experimentally using two benchmarks, one from the image domain and one from the video domain.

From the theoretical and experimental results, it can be derived that invariance to light intensity changes and light color changes affects category recognition. The results reveal further that, for light intensity shifts, the usefulness of invariance is category-specific. Overall, when choosing a single descriptor and no prior knowledge about the dataset and object and scene categories is available, the OpponentSIFT is recommended. Furthermore, a combined set of color descriptors outperforms intensity-based SIFT and improves category recognition by 8% on the PASCAL VOC 2007 and by 7% on the Mediamill Challenge.

Index Terms—Image/video retrieval, evaluation/methodology, color, invariants, pattern recognition.

I. INTRODUCTION

Image category recognition is important to access visual information on the level of objects (buildings, cars, *etc.*) and scene types (outdoor, vegetation, *etc.*). In general, systems for category recognition on images [1], [2], [3], [4], [5] and video [6], [7], [8] use machine learning based on image descriptions to distinguish object and scene categories. However, there can be large variations in viewing and lighting conditions for real-world scenes, complicating the description of images and consequently the image category recognition task. This is illustrated in figure 1. A change in *viewpoint* will yield shape variations such as the orientation and scale of the object. Salient point detection methods and corresponding region descriptors can robustly detect regions which are translation-, rotation- and scale-invariant, addressing these viewpoint changes [9], [10], [11]. In addition, changes in the *illumination*

of a scene can greatly affect the performance of object and scene type recognition if the descriptors used are not robust to these changes. To increase photometric invariance and discriminative power, color descriptors have been proposed which are robust against certain photometric changes [12], [13], [14], [15], [16]. As there are many different methods to obtain color descriptors, however, it is unclear what similarities these methods have and how they are different. To arrange color invariant descriptors in the context of image category recognition, a taxonomy is required based on principles of photometric changes.

Therefore, this paper studies the *invariance* properties and the *distinctiveness* of color descriptors in a structured way. First, a taxonomy of invariant properties is presented. The taxonomy is derived by considering the diagonal model of illumination change [17], [18], [19]. Using this model, a systematic approach is adopted to provide a set of invariance properties which achieve different amounts of *invariance*, such as invariance to light intensity changes, light intensity shifts, light color changes and light color changes and shifts. Color descriptors are tested experimentally with respect to this set of invariance properties through an object recognition dataset with known illumination changes [20]. Then, the *distinctiveness* of color descriptors is analyzed experimentally using two benchmarks from the image domain [21] and the video domain [22]. The benchmarks are very different in nature: the image benchmark consists of photographs and the video benchmark consists of keyframes from broadcast news videos. However, they share a common characteristic: both contain the illumination conditions as encountered in the real world. Based on extensive experiments on this large set of real-world image data, the usefulness of the different invariant properties is derived. As a result, new color descriptors can be designed according to the obtained invariance criteria. Finally, recommendations are given on which color descriptors to use under which circumstances and datasets.

This paper is organized as follows. In section II, the reflectance model is presented. Further, its relation to the diagonal model of illumination change is discussed. In section III, a taxonomy of color descriptors and their invariance properties is given. The experimental setup is presented in section IV. In section V, a discussion of the results is given. Finally, in section VI, conclusions are drawn.

Manuscript received August 28, 2008; revised March 8, 2009; revised June 11, 2009; accepted July 11, 2009.

¹Software to compute the color descriptors from this paper is available from <http://www.colordescriptors.com>



Fig. 1. Illustration of variations in viewing and illumination conditions for real-world scenes containing potted plants. The potted plants vary in imaging scale and are imaged under outdoor lighting, indoor lighting and a combination of the two, respectively. Images are from an image benchmark [21].

II. REFLECTANCE MODEL

An image \mathbf{f} can be modelled under the assumption of Lambertian reflectance as follows:

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} e(\lambda) \rho_k(\lambda) s(\mathbf{x}, \lambda) d\lambda, \quad (1)$$

where $e(\lambda)$ is the color of the light source, $s(\mathbf{x}, \lambda)$ is the surface reflectance and $\rho_k(\lambda)$ is the camera sensitivity function ($k \in \{R, G, B\}$). Further, ω and \mathbf{x} are the visible spectrum and the spatial coordinates respectively.

Shafer [23] proposes to add a diffuse term to the model of eq. (1). In fact, the term includes a wider range of possible causes than only diffuse light, such as interreflections, infrared sensitivity of the camera sensor, scattering in the medium or lens. The diffuse light is considered to have a lower intensity and to originate from all directions in equal amounts:

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} e(\lambda) \rho_k(\lambda) s(\mathbf{x}, \lambda) d\lambda + \int_{\omega} A(\lambda) \rho_k(\lambda) d\lambda, \quad (2)$$

where $A(\lambda)$ is the term that models the diffuse light.

By computing the derivative of image \mathbf{f} , it can be easily derived that the effect of $a(\lambda)$ is cancelled out, since it is independent of the surface reflectance term. Then, the reflection model of the spatial derivative of \mathbf{f} at location \mathbf{x} on scale σ is given by:

$$\mathbf{f}_{\mathbf{x},\sigma}(\mathbf{x}) = \int_{\omega} e(\lambda) \rho_k(\lambda) s_{\mathbf{x},\sigma}(\mathbf{x}, \lambda) d\lambda. \quad (3)$$

Hence, derivatives will yield invariance to diffuse light. The reflection model of eq. (1) corresponds to the diagonal model of illumination change under the assumption of narrow band filters. This is detailed in the next section.

A. Diagonal Model

Changes in the illumination can be modeled by a diagonal mapping or *von Kries Model* [18] as follows:

$$\mathbf{f}^c = \mathcal{D}^{u,c} \mathbf{f}^u, \quad (4)$$

where \mathbf{f}^u is the image taken under an unknown light source, \mathbf{f}^c is the same image transformed, so it appears as if it was taken under the reference light (called canonical illuminant), and $\mathcal{D}^{u,c}$ is a diagonal matrix which maps colors that are taken under an unknown light source u to their corresponding colors under the canonical illuminant c :

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix}. \quad (5)$$

To include the ‘diffuse’ light term, Finlayson *et al.* [24] extended the diagonal model with an offset $(o_1, o_2, o_3)^T$, resulting in the diagonal-offset model:

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix} + \begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix}. \quad (6)$$

The diagonal model with offset term corresponds to eq. (2) assuming narrow-band filters measured at wavelengths λ_R , λ_G and λ_B at position \mathbf{x} with surface reflectance $s(\mathbf{x}, \lambda_C)$ as follows:

$$\begin{pmatrix} e^c(\lambda_R) \\ e^c(\lambda_G) \\ e^c(\lambda_B) \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} e^u(\lambda_R) \\ e^u(\lambda_G) \\ e^u(\lambda_B) \end{pmatrix} + \begin{pmatrix} A(\lambda_R) \\ A(\lambda_G) \\ A(\lambda_B) \end{pmatrix}. \quad (7)$$

As the surface reflectance $s(\mathbf{x}, \lambda_C)$ is equal for both the canonical and the unknown illuminant, equation (7) is a simplification of $e^c(\lambda_R)s(\mathbf{x}, \lambda_R) = ae^u(\lambda_R)s(\mathbf{x}, \lambda_R) + A(\lambda_R)$, $e^c(\lambda_G)s(\mathbf{x}, \lambda_G) = be^u(\lambda_G)s(\mathbf{x}, \lambda_G) + A(\lambda_G)$ and $e^c(\lambda_B)s(\mathbf{x}, \lambda_B) = ce^u(\lambda_B)s(\mathbf{x}, \lambda_B) + A(\lambda_B)$.

For broad-band cameras, spectral sharpening can be applied to obtain narrow-band filters [17]. Note that similar to eq. (3), when image derivatives are taken (first or higher order image statistics), the offset in the diagonal-offset model will cancel out.

B. Photometric Analysis

Based on the diagonal model and the diagonal-offset model, five types of common changes in the image values $\mathbf{f}(\mathbf{x})$ are categorized in this section.

Firstly, for eq. (5), when the image values change by a constant factor in all channels (*i.e.* $a = b = c$), this is equal to a *light intensity change*:

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix}. \quad (8)$$

In addition to differences in the intensity of the light source, light intensity changes also include (no-colored) shadows and shading. Hence, when a descriptor is invariant to light intensity changes, it is *scale-invariant* with respect to (light) intensity.

Secondly, an equal shift in image intensity values in all channels, *i.e.* *light intensity shift*, where $(o_1 = o_2 = o_3)$ and $(a = b = c = 1)$ will yield:

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}. \quad (9)$$

Light intensity shifts are due to diffuse lighting including scattering of a white light source, object highlights (specular component of the surface) under a white light source, inter-reflections and infrared sensitivity of the camera sensor. When a descriptor is invariant to a light intensity shift, it is *shift-invariant* with respect to light intensity.

Thirdly, the above classes of changes can be combined to model both intensity changes and shifts:

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}; \quad (10)$$

i.e. an image descriptor robust to these changes is scale-invariant and shift-invariant with respect to light intensity.

Fourthly, in the full diagonal model (*i.e.* allowing $a \neq b \neq c$), the image channels scale independently (eq. (5)). This allows for *light color changes* in the image. Hence, this class of changes can model a change in the illuminant color and light scattering, amongst others.

Finally, the full diagonal-offset model (eq. (6)) models arbitrary offsets ($o_1 \neq o_2 \neq o_3$), besides the light color changes ($a \neq b \neq c$) offered by the full diagonal model. This type of change is called *light color change and shift*.

In conclusion, five types of common changes have been identified based on the diagonal-offset model of illumination change, *i.e.* variations to light intensity changes, light intensity shifts, light intensity changes and shifts, light color changes and light color changes and shifts.

III. COLOR DESCRIPTORS AND INVARIANT PROPERTIES

In this section, color descriptors are presented and their invariance properties are summarized. First, color descriptors based on histograms are discussed. Then, color moments and color moment invariants are presented. Finally, color descriptors based on SIFT are discussed. These three types of descriptors were chosen due to their distinct nature and wide-spread use. Color histograms do not contain local spatial information and are inherently pixel-based. Color moments do contain local photometrical and spatial information derived from pixel values. SIFT descriptors contain local spatial information and are derivative-based.

See table I for an overview of the descriptors and their invariance properties. We define *invariance* of a descriptor to condition A as follows: under a condition A, the descriptor is independent of changes in condition A. The independence is derived analytically under the assumption that no color clipping occurs. Color clipping occurs when the color of a pixel falls outside the valid range and is subsequently clipped to the minimum or maximum of the range. For example, for a very large scaling of the intensity in eq. (8), color clipping occurs if the scaled values exceed 255, the maximum value typically used for image storage.

A. Histograms

RGB histogram The *RGB* histogram is a combination of three 1-D histograms based on the *R*, *G* and *B* channels of

the *RGB* color space. This histogram possesses no invariance properties.

Opponent histogram The opponent histogram is a combination of three 1-D histograms based on the channels of the opponent color space:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}. \quad (11)$$

The intensity information is represented by channel O_3 and the color information by O_1 and O_2 . Due to the subtraction in O_1 and O_2 , the offsets will cancel out if they are equal for all channels (e.g. a white light source). This is verified by substituting the unknown illuminant from eq. (9) with offset o_1 :

$$\begin{aligned} \begin{pmatrix} O_1 \\ O_2 \end{pmatrix} &= \begin{pmatrix} \frac{R^c - G^c}{\sqrt{2}} \\ \frac{R^c + G^c - 2B^c}{\sqrt{6}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{(R^u + o_1) - (G^u + o_1)}{\sqrt{2}} \\ \frac{(R^u + o_1) + (G^u + o_1) - 2(B^u + o_1)}{\sqrt{6}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{R^u - G^u}{\sqrt{2}} \\ \frac{R^u + G^u - 2B^u}{\sqrt{6}} \end{pmatrix}. \end{aligned} \quad (12)$$

Therefore, these O_1 and O_2 are shift-invariant with respect to light intensity. The intensity channel O_3 has no invariance properties.

Hue histogram In the *HSV* color space, it is known that the hue becomes unstable near the grey axis. To this end, Van de Weijer *et al.* [14] apply an error propagation analysis to the hue transformation. The analysis shows that the certainty of the hue is inversely proportional to the saturation. Therefore, the hue histogram is made more robust by weighing each sample of the hue by its saturation. The *H* color model is scale-invariant and shift-invariant with respect to light intensity [14].

rg-histogram In the normalized *RGB* color model, the chromaticity components *r* and *g* describe the color information in the image (*b* is redundant as $r + g + b = 1$):

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}. \quad (13)$$

Because of the normalization, *r* and *g* are scale-invariant and thereby invariant to light intensity changes, shadows and shading [25] from eq. (8):

$$\begin{aligned} \begin{pmatrix} r \\ g \end{pmatrix} &= \begin{pmatrix} \frac{R^c}{R^c + G^c + B^c} \\ \frac{G^c}{R^c + G^c + B^c} \end{pmatrix} = \begin{pmatrix} \frac{aR^u}{aR^u + aG^u + aB^u} \\ \frac{aG^u}{aR^u + aG^u + aB^u} \end{pmatrix} \\ &= \begin{pmatrix} \frac{aR^u}{a(R^u + G^u + B^u)} \\ \frac{aG^u}{a(R^u + G^u + B^u)} \end{pmatrix} = \begin{pmatrix} \frac{R^u}{R^u + G^u + B^u} \\ \frac{G^u}{R^u + G^u + B^u} \end{pmatrix}. \end{aligned} \quad (14)$$

Transformed color distribution An *RGB* histogram is not invariant to changes in lighting conditions. However, by normalizing the pixel value distributions, scale-invariance and shift-invariance is achieved with respect to light intensity. Because each channel is normalized independently, the descriptor

TABLE I

INVARIANCE OF DESCRIPTORS (SECTION III) AGAINST TYPES OF CHANGES IN THE DIAGONAL-OFFSET MODEL AND ITS SPECIALIZATIONS (SECTION II-B). INVARIANCE IS INDICATED WITH '+', LACK OF INVARIANCE IS INDICATED WITH '-'. THE INVARIANCE OF A DESCRIPTOR TO CONDITION A IS DEFINED AS FOLLOWS: UNDER A CONDITION A, THE DESCRIPTOR IS INDEPENDENT OF CHANGES IN CONDITION A. THE INDEPENDENCE IS DERIVED ANALYTICALLY UNDER THE ASSUMPTION THAT NO COLOR CLIPPING OCCURS.

	Light intensity change $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$	Light intensity shift $\begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$	Light intensity change and shift $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$	Light color change $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$	Light color change and shift $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix}$
RGB Histogram	-	-	-	-	-
O_1, O_2	-	+	-	-	-
O_3 , Intensity	-	-	-	-	-
Hue	+	+	+	-	-
Saturation	-	-	-	-	-
r, g	+	-	-	-	-
Transformed color	+	+	+	+	+
Color moments	-	+	-	-	-
Moment invariants	+	+	+	+	+
SIFT (∇I)	+	+	+	-	-
HSV-SIFT	-	-	-	-	-
HueSIFT	+	+	+	-	-
OpponentSIFT	+	+	+	-	-
C-SIFT	+	-	-	-	-
rg SIFT	+	-	-	-	-
Transf. color SIFT	+	+	+	+	+
RGB-SIFT	+	+	+	+	+

is also normalized against changes in light color and arbitrary offsets:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R - \mu_R}{\sigma_R} \\ \frac{G - \mu_G}{\sigma_G} \\ \frac{B - \mu_B}{\sigma_B} \end{pmatrix}, \quad (15)$$

with μ_C the mean and σ_C the standard deviation of the distribution in channel C computed over the area under consideration (e.g. a patch or image). This yields for every channel a distribution where $\mu = 0$ and $\sigma = 1$.

B. Color Moments and Moment Invariants

A color image corresponds to a function I defining RGB triplets for image positions (x, y) : $I : (x, y) \mapsto (R(x, y), G(x, y), B(x, y))$. By regarding RGB triplets as data points coming from a distribution, it is possible to define moments. Mindru *et al.* [26] have defined *generalized color moments* M_{pq}^{abc} :

$$M_{pq}^{abc} = \int \int x^p y^q [I_R(x, y)]^a [I_G(x, y)]^b [I_B(x, y)]^c dx dy. \quad (16)$$

M_{pq}^{abc} is referred to as a generalized color moment of *order* $p + q$ and *degree* $a + b + c$. Note that moments of order 0 do not contain any spatial information, while moments of degree 0 do not contain any photometric information. Thus, moment descriptions of order 0 are rotationally invariant, while higher orders are not. A large number of moments can be created with small values for the order and degree. However, for larger values the moments are less stable. Typically, generalized color moments up to the first order and the second degree are used.

By using the proper combination of moments, it is possible to normalize against photometric changes. These combinations are called *color moment invariants*. Invariants involving only a single color channel (e.g. out of a, b and c two are 0) are called 1-band invariants. Similarly there are 2-band invariants involving only two out of three color bands. 3-band invariants

involve all color channels, but these can always be created by using 2-band invariants for different combinations of channels.

Color moments The color moment descriptor uses all generalized color moments up to the second degree and the first order. This lead to nine possible combinations for the degree: $M_{pq}^{000}, M_{pq}^{100}, M_{pq}^{010}, M_{pq}^{001}, M_{pq}^{200}, M_{pq}^{110}, M_{pq}^{020}, M_{pq}^{011}, M_{pq}^{002}$ and M_{pq}^{101} . Combined with three possible combinations for the order: $M_{00}^{abc}, M_{10}^{abc}$ and M_{01}^{abc} , the color moment descriptor has 27 dimensions. These color moments only have shift-invariance. This is achieved by subtracting the average in all input channels before computing the moments.

Color moment invariants Color moment invariants can be constructed from generalized color moments. All 3-band invariants are computed from Mindru *et al.* [26]. To be comparable, the \tilde{C}_{02} invariants are considered. This gives a total of 24 color moment invariants, which are invariant to all the properties listed in table I.

C. Color SIFT Descriptors

SIFT The SIFT descriptor proposed by Lowe [9] describes the local shape of a region using edge orientation histograms. The gradient of an image is shift-invariant: taking the derivative cancels out offsets (section II-B). Under light intensity changes, *i.e.* a scaling of the intensity channel, the gradient direction and the relative gradient magnitude remain the same. Because the SIFT descriptor is normalized, the gradient magnitude changes have no effect on the final descriptor. The SIFT descriptor is not invariant to light color changes, because the intensity channel is a combination of the R, G and B channels. To compute SIFT descriptors, the version described by Lowe [9] is used.

HSV-SIFT Bosch *et al.* [16] compute SIFT descriptors over all three channels of the HSV color model. This gives 3x128 dimensions per descriptor, 128 per channel. As stated earlier,

[†]Because it is constant, the moment M_{pq}^{000} is excluded.

the H color model is scale-invariant and shift-invariant with respect to light intensity. However, due to the combination of the HSV channels, the complete descriptor has no invariance properties. Further, the instability of the hue for low saturation is not addressed here.

HueSIFT Van de Weijer *et al.* [14] introduce a concatenation of the hue histogram (see section III-A) with the SIFT descriptor. When compared to HSV-SIFT, the usage of the weighed hue histogram addresses the instability of the hue near the grey axis. Because the bins of the hue histogram are independent, the periodicity of the hue channel for HueSIFT is addressed. Similar to the hue histogram, the HueSIFT descriptor is scale-invariant and shift-invariant.

OpponentSIFT OpponentSIFT describes all the channels in the opponent color space (eq. (11)) using SIFT descriptors. The information in the O_3 channel is equal to the intensity information, while the other channels describe the color information in the image. These other channels do contain some intensity information, but due to the normalization of the SIFT descriptor they are invariant to changes in light intensity.

C-SIFT In the opponent color space, the O_1 and O_2 channels still contain some intensity information. To add invariance to intensity changes, [13] proposes the C-invariant which eliminates the remaining intensity information from these channels. The use of color invariants as input for SIFT was first suggested by Abdel-Hakim and Farag [12]. The C-SIFT descriptor [15] uses the C invariant, which can be intuitively seen as the normalized opponent color space $\frac{O_1}{O_3}$ and $\frac{O_2}{O_3}$. Because of the division by intensity, the scaling in the diagonal model will cancel out, making C-SIFT scale-invariant with respect to light intensity. Due to the definition of the color space, the offset does not cancel out when taking the derivative: it is not shift-invariant.

rgSIFT For the *rgSIFT* descriptor, descriptors are added for the r and g chromaticity components of the normalized RGB color model from eq. (13), which is already scale-invariant.

Transformed color SIFT For the transformed color SIFT, the same normalization is applied to the RGB channels as for the transformed color histogram (eq. (15)). For every normalized channel, the SIFT descriptor is computed. The descriptor is scale-invariant, shift-invariant and invariant to light color changes and shift.

RGB-SIFT For the RGB-SIFT descriptor, SIFT descriptors are computed for every RGB channel independently. An interesting property of this descriptor, is that its descriptor values are equal to the transformed color SIFT descriptor. This is explained by looking at the transformed color space (eq. (15)): this transformation is already implicitly performed when SIFT is applied to each RGB channel independently. Because the SIFT descriptor operates on derivatives only, the subtraction of the means in the transformed color model is redundant, as this offset is already cancelled out by taking derivatives. Similarly, the division by the standard deviation is already implicitly performed by the normalization of the vector length of SIFT descriptors. Therefore, as the RGB-SIFT and transformed color SIFT descriptors are equal, we will use the RGB-SIFT name throughout this paper.

D. Conclusion

In this section, three different groups of color descriptors were discussed: histograms in different color spaces, color moments and moment invariants and color extensions of SIFT. For each color descriptor, the invariance with respect to illumination changes in the diagonal-offset model were analyzed. The results are summarized in table I.

IV. EXPERIMENTAL SETUP

In this section, the experimental setup to evaluate the different color descriptors is outlined. The *invariance* properties of the color descriptors, which were derived analytically in the previous section, are verified experimentally as well using a dataset with known illumination conditions. The *distinctiveness* of the color descriptors is assessed experimentally through their discriminative power on the dataset with known imaging conditions, an image benchmark and a video benchmark.

First, implementation details of the descriptors in an object and scene recognition setting are discussed. Then, the datasets used for evaluation are described. After discussing these benchmarks and their datasets, evaluation criteria are given.

A. Feature Extraction Pipelines

To empirically test the different color descriptors, the descriptors are computed at scale-invariant points [5], [9]. See figure 2 for an overview of the processing pipeline. In the pipeline shown, scale-invariant points are obtained with the Harris-Laplace point detector on the intensity channel. Other region detectors [10], such as the dense sampling detector, Maximally Stable Extremal Regions [27] and Maximally Stable Color Regions [28], can be plugged in. For the experiments, the Harris-Laplace point detector is used because it has shown good performance for category recognition [5]. This detector uses the Harris corner detector to find potential scale-invariant points. It then selects a subset of these points for which the Laplacian-of-Gaussians reaches a maximum over scale. The color descriptors from section III are computed over the area around the points. The size of this area depends on the maximum scale of the Laplacian-of-Gaussians [10].

To obtain fixed-length feature vectors per image, the bag-of-words model is used [29]. The bag-of-words model is also known as ‘textons’ [30], ‘object parts’ [31] and ‘codebooks’ [32], [33]. The bag-of-words model performs vector quantization of the color descriptors in an image against a visual codebook. A descriptor is assigned to the codebook element which is closest in Euclidian space. To be independent of the total number of descriptors in an image, the feature vector is normalized to sum to 1.

The visual codebook is constructed by applying k -means clustering to 200,000 randomly sampled descriptors from the set of images available for training. In this paper, visual codebooks with 4,000 elements are used.

Color descriptor software implementing this processing pipeline is available from our website². It performs point sampling, color descriptor computation and vector quantization.

²<http://www.colordescriptors.com>

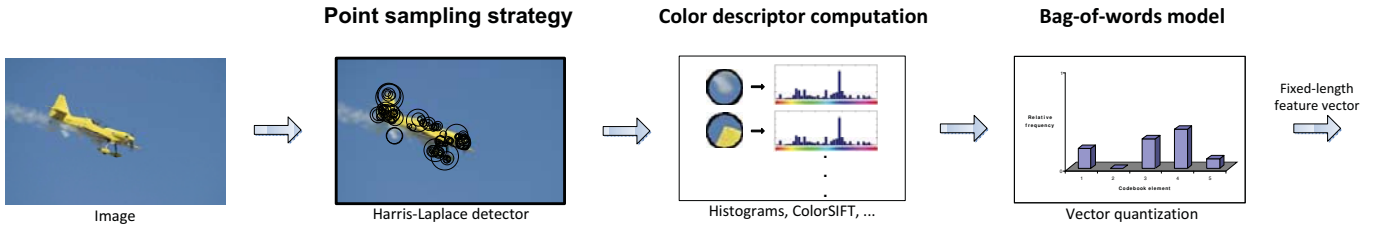


Fig. 2. The stages of the primary feature extraction pipeline used in this paper. First, the Harris-Laplace salient point detector is applied to the image. Then, for every point a color descriptor is computed over the area around the point. All the color descriptors of an image are subsequently vector quantized against a codebook of prototypical color descriptors. This results in a fixed-length feature vector representing the image.

After these steps, an image is represented by a fixed-length feature vector.

B. Classification

For datasets where only a single training example is available per object or scene category, a nearest neighbor classifier is used with χ^2 distances between feature vectors F and F' :

$$\text{dist}_{\chi^2}(\vec{F}, \vec{F}') = \frac{1}{2} \sum_{i=1}^n \frac{(\vec{F}_i - \vec{F}'_i)^2}{\vec{F}_i + \vec{F}'_i}, \quad (17)$$

with n the size of the feature vectors. For notational convenience, $\frac{0}{0}$ is assumed to be equal to 0 iff $\vec{F}_i = \vec{F}'_i = 0$.

For datasets with multiple training examples, the support vector machines classifier is used. The decision function of a support vector machines classifier for a test sample with feature vector \vec{F}' has the form:

$$g(\vec{F}') = \sum_{\vec{F} \in \text{trainset}} \alpha_{\vec{F}} y_{\vec{F}} k(\vec{F}, \vec{F}') - \beta, \quad (18)$$

where $y_{\vec{F}}$ is the class label of \vec{F} (-1 or $+1$), $\alpha_{\vec{F}}$ is the learned weight of train sample \vec{F} , β is a learned threshold and $k(\vec{F}, \vec{F}')$ is the value of a kernel function based on the χ^2 distance, which has shown good results in object recognition [5]:

$$k(\vec{F}, \vec{F}') = e^{-\frac{1}{D} \text{dist}_{\chi^2}(\vec{F}, \vec{F}')}, \quad (19)$$

where D is a scalar which normalizes the distances. We set D to the average χ^2 distance between all elements of the train set.

The LibSVM implementation [34] is used to train the classifier. As parameters for the training phase, the weight of the positive class is set to $\frac{\#pos + \#neg}{\#pos}$ and the weight of the negative class is set to $\frac{\#pos + \#neg}{\#neg}$, with $\#pos$ the number of positive instances in the train set and $\#neg$ the number of negative instances. The cost parameter is optimized using 3-fold cross-validation with a parameter range of 2^{-4} through 2^4 .

To use multiple features, instead of relying on a single feature, the kernel function is extended in a weighted fashion for m features:

$$k(\{\vec{F}_{(1)}, \dots, \vec{F}_{(m)}\}, \{\vec{F}'_{(1)}, \dots, \vec{F}'_{(m)}\}) = e^{-\frac{1}{\sum_{j=1}^m w_j} \left(\sum_{j=1}^m \frac{w_j}{D_j} \text{dist}(\vec{F}_{(j)}, \vec{F}'_{(j)}) \right)}, \quad (20)$$

with w_j the weight of the j^{th} feature, D_j the normalization factor for the j^{th} feature and $\vec{F}_{(j)}$ the j^{th} feature vector.

An example of the use of multiple features is the spatial pyramid [3]; it is illustrated in figure 3. When using the spatial pyramid, additional features are extracted for specific parts of the image. For example, in a 2×2 subdivision of the image, feature vectors are extracted for each image quarter with a weight of $\frac{1}{4}$ for each quarter. Similarly, a 1×3 subdivision consisting of three horizontal bars, which introduces three new features (each with a weight of $\frac{1}{3}$). In this setting, the feature vector for the entire image has a weight of 1.

C. Experiment 1: Illumination Changes

The Amsterdam Library of Object Images (ALOI) dataset [20] contains more than 48,000 images of 1,000 objects, under various illumination conditions. Light intensity scaling (eq. (8)) and light intensity shifts (eq. (9)) are not present in the dataset, therefore we have artificially added these two condition changes to the dataset. The effect of simultaneous light intensity changes and shifts (eq. (10)) is a combination of the previous two properties. Since these two properties are already evaluated individually, we refrain from evaluating this combined property. The light color change images from ALOI directly correspond to our light color changes (eq. (5)). The light color is varied by changing the illumination color temperature, resulting in objects illuminated under a reddish to white light. For completeness, the other conditions present in the ALOI dataset are also included: objects lighted by a different number of white lights at increasingly oblique angles (between one and three white lights around the object, introducing selfshadowing for up to half of the object), object rotation images and images with different levels of JPEG compression.

Because only a single training example is available per object category, the nearest neighbour classifier is used for the ALOI dataset.

D. Experiment 2: Image Benchmark

The PASCAL Visual Object Classes Challenge [21] provides a yearly benchmark for comparison of object classification systems. The PASCAL VOC Challenge 2007 dataset contains nearly 10,000 images of 20 different object categories, e.g. bird, bottle, car, dining table, motorbike and people. The dataset is divided into a predefined train set (5011 images) and test set (4952 images).

Experiment 1: Illumination Changes

Eq.

Color Descriptors

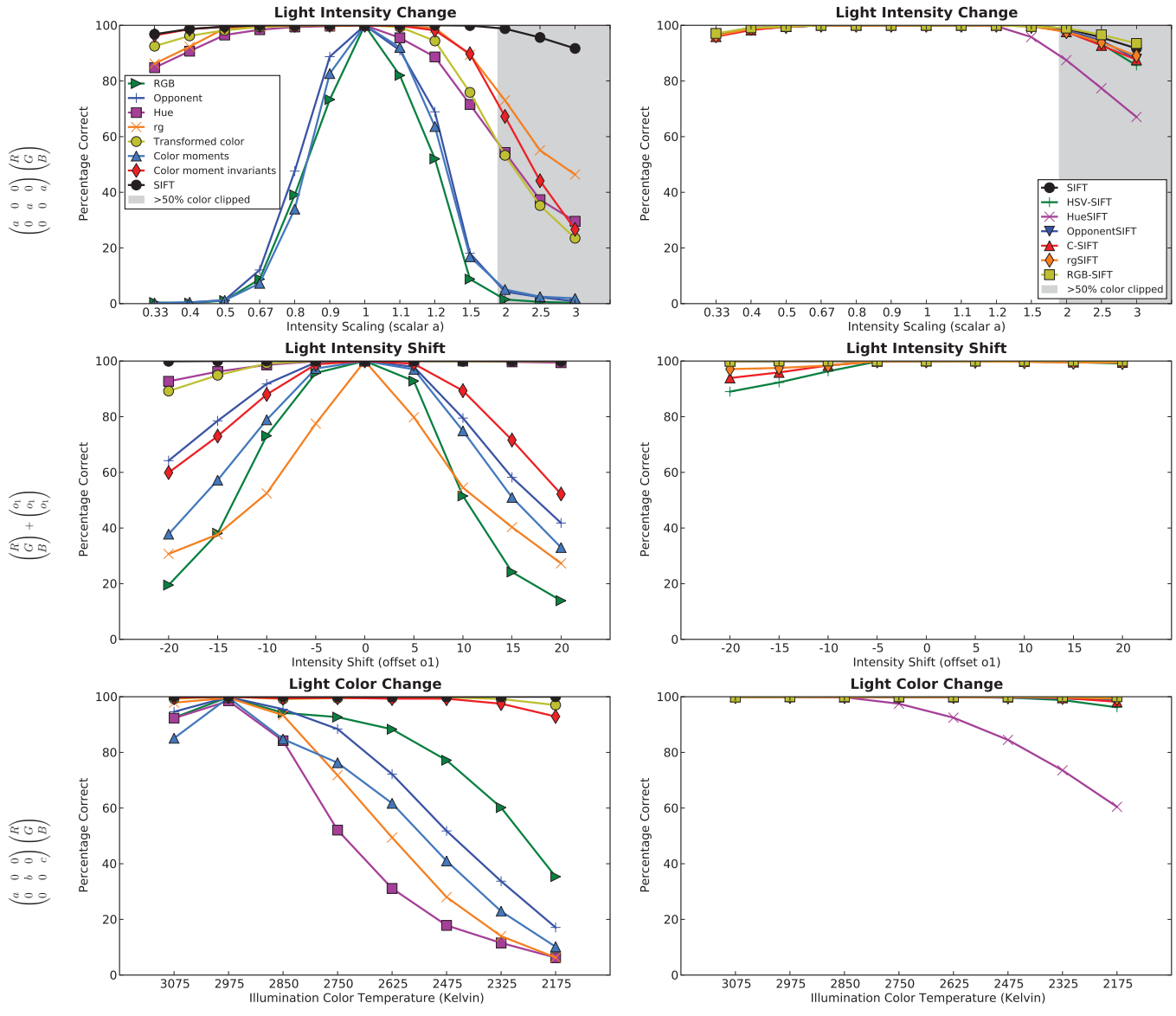


Fig. 4. Evaluation of the invariance properties of color descriptors under different illumination conditions, averaged over 1000 objects from the ALOI dataset [20]. Performance is measured using the percentage of correctly identified objects. For clarity of presentation, the results have been split into two parts. To allow for easier comparison, SIFT is shown in both the graphs on the left and the graphs on the right. The rows correspond to the invariant properties from section II, as listed in the graph titles and the equations shown. For light intensity shifts, the axis unit corresponds to image values in the range [0, 255]. For the light color changes, the light color is varied by changing the illumination color temperature, resulting in objects illuminated under a white to reddish light. Conditions where, on average, more than 50% of the object area is affected by color clipping (due to image values falling outside the range [0, 255]) are marked with a grey background.

V. RESULTS

A. Experiment 1: Illumination Changes

From the results in figure 4, the theoretical invariance properties of color descriptors are validated. By observing the results with respect to light intensity changes, the color descriptors without invariance to this property, such as the RGB histogram, the opponent color histogram and color moments, do not perform well. There is a clear distinction in performance between these descriptors and the invariant descriptors, such as the hue histogram, color moment invariants and SIFT. Overall, within this group of invariant descriptors,

the SIFT and color SIFT descriptors perform much better than histogram-based descriptors; they have higher discriminative power. HueSIFT, which is a combination of the hue histogram and the SIFT descriptor, falls between these descriptor classes in terms of performance. The HSV-SIFT descriptor, which is not invariant to light intensity changes, is the lowest-scoring SIFT descriptor after HueSIFT. For very large scaling factors, the performance of all descriptors drops. This is due to color clipping: scaled image values outside the range [0, 255] are clipped to 255. In figure 4, a grey background indicates under which conditions, on average, more than half of all object

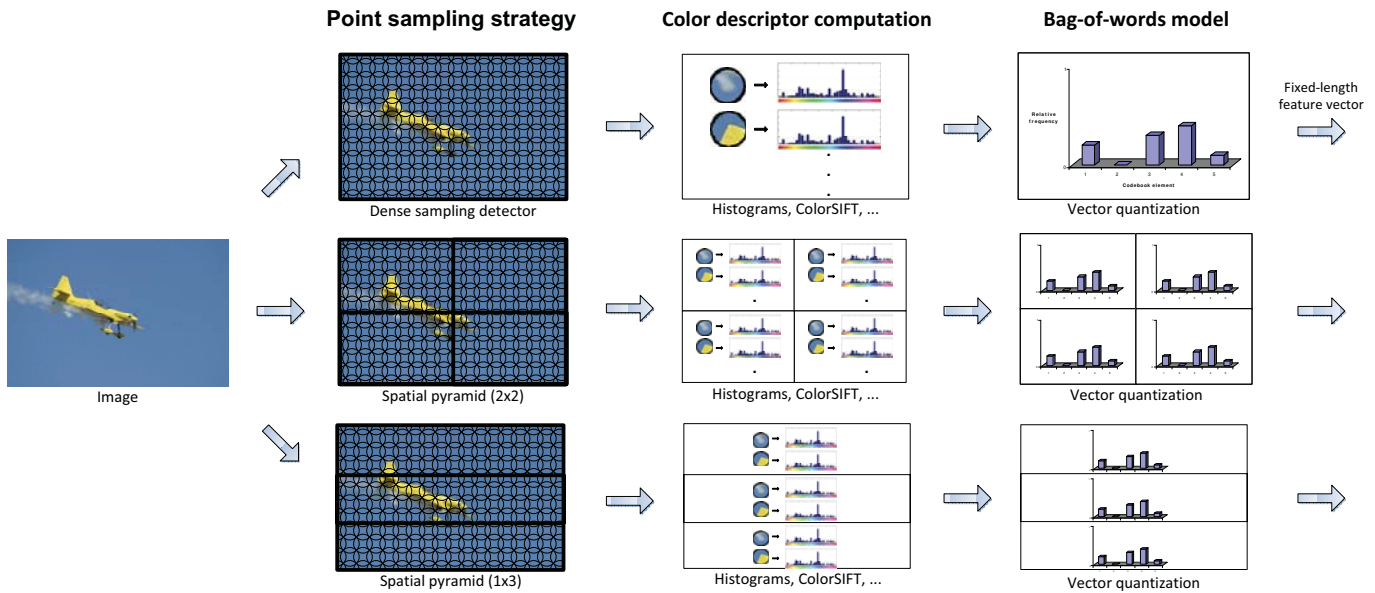


Fig. 3. Examples of additional feature extraction pipelines used in this paper, besides the primary pipeline shown in figure 2. The pipelines shown are examples of using a different point sampling strategy or a spatial pyramid [3]. The spatial pyramid constructs feature vectors for specific parts of the image. For every pipeline, first, a point sampling method is applied to the image. Then, for every point a color descriptor is computed over the area around the point. All the color descriptors of an image are subsequently vector quantized against a codebook of prototypical color descriptors. This results in a fixed-length feature vector representing the image.

E. Experiment 3: Video Benchmark

The Mediamill Challenge by Snoek *et al.* [22] provides an annotated video dataset, based on the training set of the NIST TRECVID 2005 benchmark [7]. Over this dataset, repeatable experiments have been defined. The experiments decompose automatic category recognition into a number of components, for which they provide a standard implementation. This provides an environment to analyze which components affect the performance most.

The dataset of 86 hours is divided into a Challenge training set (70% of the data or 30,993 shots) and a Challenge test set (30% of the data or 12,914 shots). For every shot, the Challenge provides a single representative keyframe image. So, the complete dataset consists of 43,907 images, one for every video shot. The dataset consists of television news from November 2004 broadcasted on six different TV channels in three different languages: English, Chinese and Arabic. On this dataset, the 39 LSCOM-Lite categories [35] are employed. These include object categories like aircraft, animal, car and faces, and scene categories such as desert, mountain, sky, urban and vegetation.

F. Evaluation Criteria

Experiments on the ALOI dataset perform object recognition using one example: given a query image of an object under unknown illumination conditions, the top-ranked result should be equal to the original image of the object for successful recognition. The percentage of objects where the top-ranked result is indeed the correct object is used as the performance on the ALOI dataset.

For our benchmark results, the average precision is taken as the performance metric for determining the accuracy of

ranked category recognition results. The average precision is a single-valued measure that is proportional to the area under a precision-recall curve. This value is the average of the precision over all images/keyframes judged to be relevant. Hence, it combines both precision and recall into a single performance value. For the PASCAL VOC Challenge 2007, the official standard is the 11-point interpolated average precision, and for TRECVID, the official standard is the non-interpolated average precision. The interpolated average precision is an approximation of the non-interpolated average precision. As the difference between the two is generally very small, we will follow the official standard for each dataset and refer to them as average precision scores. When performing experiments over multiple object and scene categories, the average precisions of the individual categories are aggregated. This aggregation, mean average precision, is calculated by taking the mean of the average precisions. As average precision depends on the number of correct object and scene categories present in the test set, the mean average precision depends on the dataset used.

To obtain an indication of significance, the bootstrap method [36], [37] is used to estimate confidence intervals for mean average precision. In bootstrap, multiple test sets T_B are created by selecting images at random from the original test set T , with replacement, until $|T| = |T_B|$. This has the effect that some images are replicated in T_B , whereas other images may be absent. This process is repeated 1000 times to generate 1000 test sets, each obtained by sampling from the original test set T . The statistical accuracy of the mean average precision score can then be evaluated by looking at the standard deviation of the mean average precision scores over the different bootstrap test sets.

Lighting Arrangement Changes, Viewpoint Changes and JPEG compression

Color Descriptors

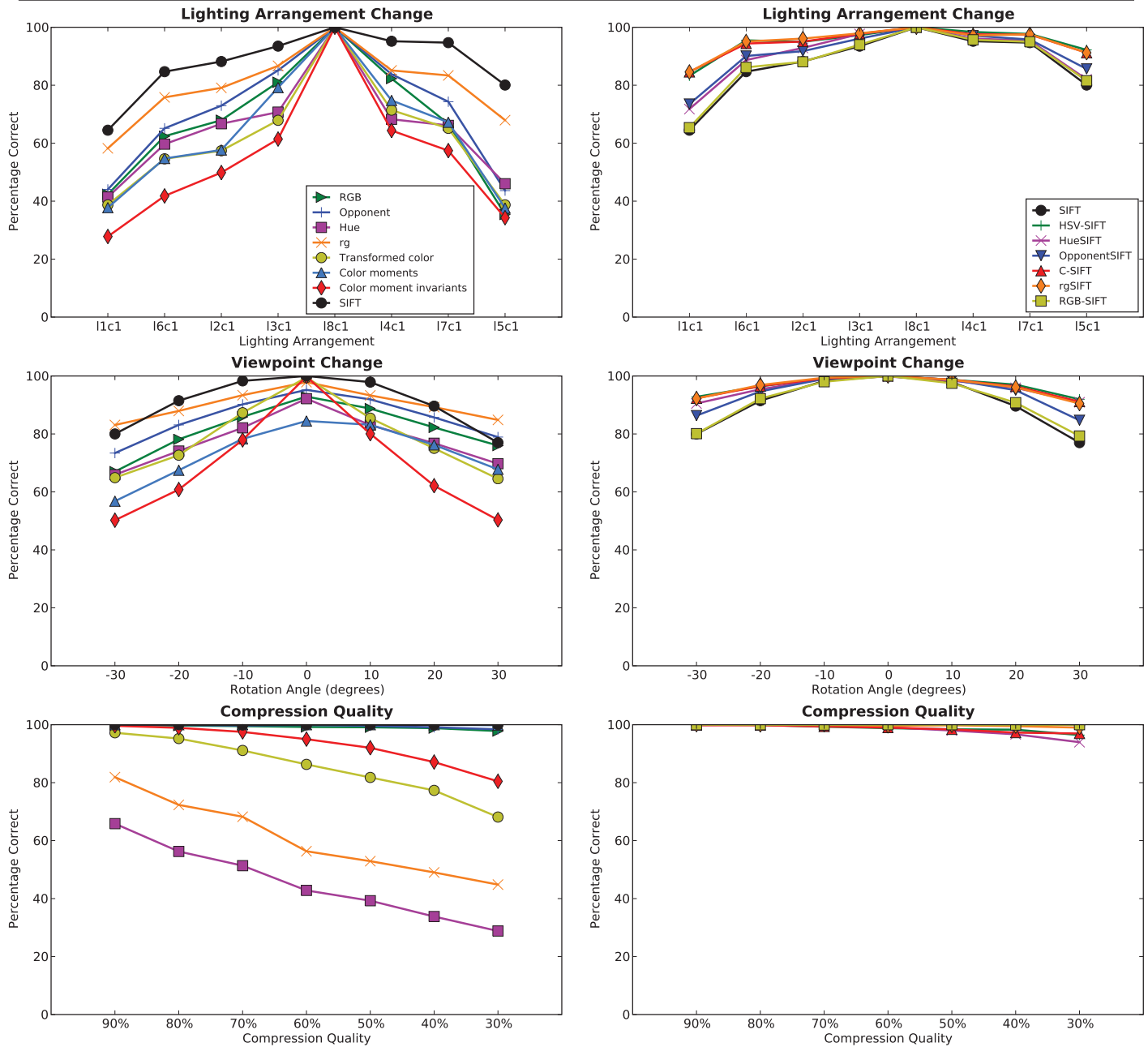


Fig. 5. For completeness, this figure contains the results for color descriptors under different lighting arrangements at increasingly oblique angles (between one and three of the lights around the object are on, introducing selfshadowing for up to half of the object), different viewpoint angles and different degrees of JPEG compression, averaged over 1000 objects from the ALOI dataset [20]. Performance is measured using the percentage of correctly identified objects. For clarity of presentation, the results have been split into two parts. To allow for easier comparison, SIFT is shown in both the graphs on the left and the graphs on the right.

pixels have been clipped.

For light intensity shifts, it is shown that the color descriptors which lack invariance, the *RGB* histogram, the opponent color histogram and the *rg* histogram, indeed have reduced performance. Additionally, color moments and color moment invariants are affected when the shift amount increases, these descriptors can only handle small light intensity shifts. The three color SIFT descriptors which lack shift-invariance, HSV-SIFT, C-SIFT and *rg*SIFT, show reduced performance for large shifts when compared to other SIFT variants, confirming their lack of invariance.

For light color changes, it is observed that histograms do not perform well. This is consistent with their lack of invariance. The exceptions are the transformed color histogram and the color moment invariants, which do possess invariance to light color changes and indeed perform much better. For the SIFT-based descriptors, only HSV-SIFT and HueSIFT degrade in performance as the light color changes. This is due to their lack of invariance. Of interest is that some of the descriptors which are not invariant to light color changes, *e.g.* OpponentSIFT, C-SIFT and *rg*SIFT, are (in practice) largely robust to the light color changes present in the ALOI dataset.

Besides the evaluation of the invariant properties, there are also different conditions which can be evaluated using ALOI. For the lighting arrangement changes, shown in figure 5, between one and three white lights around the object are turned on. This leads to shadows, shading and white highlights, *e.g.* to both light intensity scaling and shifts (eq. (10)), but also to partial visibility due to lack of light on certain parts of the object. In this setting, both the invariant properties and the discriminative power of color descriptors play an important role. The intensity scale-invariant C-SIFT and *rg*SIFT perform well, ahead of the OpponentSIFT descriptor, which is also shift-invariant. For the RGB-SIFT descriptor, which is invariant to light color changes in addition to begin scale-invariant and shift-invariant, the increased invariance comes at the price of reduced discriminative power: it is behind C-SIFT, *rg*SIFT and OpponentSIFT under this condition. For this condition, light intensity shifts and light color changes do not occur and therefore OpponentSIFT and RGB-SIFT are too invariant. A similar pattern is observed from the results in figure 5 for viewpoint changes due to object rotation. The scale-invariant C-SIFT and *rg*SIFT perform best, and the light intensity shift invariance offered by OpponentSIFT and RGB-SIFT is not needed, nor is the light color invariance of RGB-SIFT.

From the results shown in figure 5 for JPEG compression quality, it can be seen that the hue histogram, the *rg* histogram, the transformed color histogram and the color moment invariants are not robust to even moderate amounts of compression: compression artifacts cause large deviations in these descriptors.

In conclusion, changes in lighting conditions affect color descriptors. However, for object recognition, not just the invariance of a color descriptor to lighting conditions is important, but also the distinctiveness of the descriptor. An invariant descriptor is only useful for visual categorization when it has sufficient discriminative power as well. Finally, certain color descriptors are sensitive to compression artifacts,

Experiment 2: Descriptor performance on image benchmark

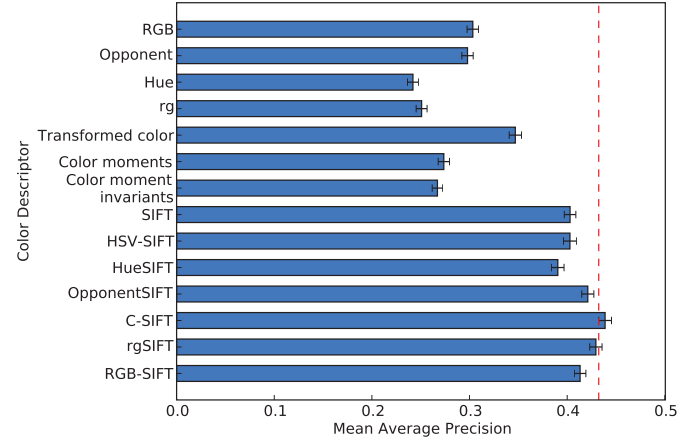


Fig. 6. Evaluation of color descriptors on an image benchmark, the PASCAL VOC Challenge 2007 [21], averaged over the 20 object categories. Error bars indicate the standard deviation in mean average precision, obtained using bootstrap. The dashed lines indicate the lower bound of the C-SIFT confidence interval.

Experiment 2: Descriptor performance split out per category

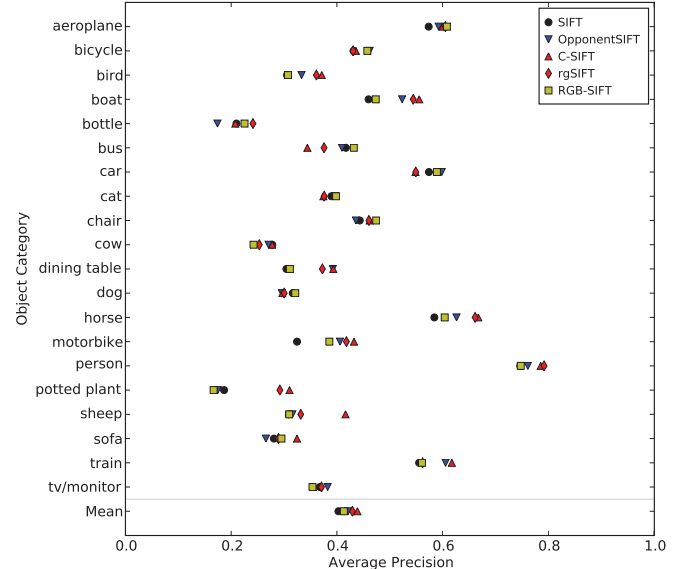


Fig. 7. Evaluation of color descriptors on an image benchmark, the PASCAL VOC Challenge 2007, split out per object category. SIFT and the best four color SIFT variants from figure 6 are shown.

reducing their usefulness. Although the best choice of color descriptor depends on the condition, the descriptors with the best overall performance are C-SIFT, *rg*SIFT, OpponentSIFT and RGB-SIFT.

B. Experiment 2: Image Benchmark

From the results shown in figure 6, it is observed that for object category recognition the SIFT variants perform significantly better than color moments, moment invariants and color histograms. The moments and histograms are not very distinctive when compared to SIFT-based descriptors: they contain too little relevant information to be competitive with SIFT.

For SIFT and the four best color SIFT descriptors from

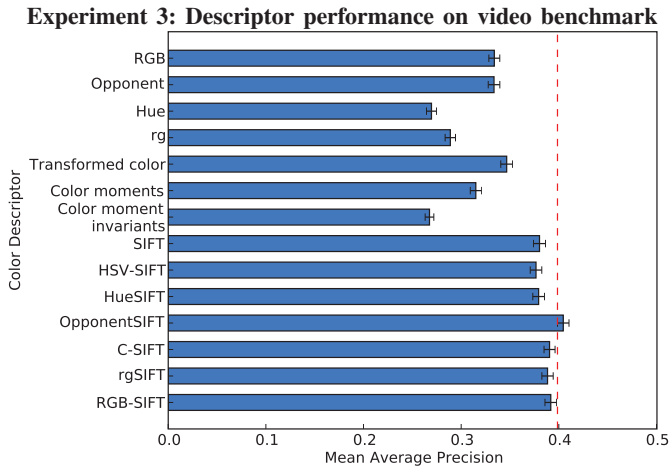


Fig. 8. Evaluation of color descriptors on a video benchmark, the Mediamill Challenge [22], averaged over 39 object and scene categories. Error bars indicate the standard deviation in mean average precision, obtained using bootstrap. The dashed line indicates the lower bound of the OpponentSIFT confidence interval.

figure 6 (OpponentSIFT, C-SIFT, rg SIFT and RGB-SIFT), the results per object category are shown in figure 7. For bird, boat, horse, motorbike, person, potted plant and sheep, it can be observed that the descriptors which perform best have scale-invariance for light intensity (C-SIFT and rg SIFT). Of these two scale-invariant descriptors, C-SIFT has the highest overall performance. The performance of the OpponentSIFT descriptor, which is also shift-invariant compared to C-SIFT, indicates that only scale-invariance, i.e. invariance to light intensity changes, is important for these object categories. RGB-SIFT includes additional invariance against light intensity shifts and light color changes and shifts when compared to C-SIFT. However, this additional invariance makes the descriptor less discriminative for these object categories, because a reduction in performance is observed. This is illustrated by the examples shown in figure 1 for potted plant, which are ranked significantly higher for C-SIFT and rg SIFT compared to OpponentSIFT and RGB-SIFT.

In conclusion, C-SIFT is significantly better than all other descriptors except rg SIFT (see figure 6) on the image benchmark. The corresponding invariant property of both of these descriptors is given by eq. (8). However, the difference between the rg SIFT descriptor and OpponentSIFT, which corresponds to eq. (10), is not significant. Therefore, the best choice for this dataset is C-SIFT.

C. Experiment 3: Video Benchmark

From the visual categorization results shown in figure 8, the same overall pattern as for the image benchmark is observed: SIFT and color SIFT variants perform significantly better than the other descriptors. The shift-invariant OpponentSIFT has left C-SIFT behind and is now the only descriptor which is significantly better than all other descriptors. An analysis on the individual object and scene categories shows that the OpponentSIFT descriptor performs best for building, meeting, mountain, office, outdoor, sky, studio, walking/running and weather news. All these concepts occur under a wide range

of light intensities and different amounts of diffuse lighting. Therefore, its invariance to light intensity changes and shifts makes OpponentSIFT a good feature for these categories, and explains why it is better than C-SIFT and rg SIFT for the video benchmark. RGB-SIFT, with additional invariance to light color changes and shifts, does not differ significantly from C-SIFT and rg SIFT. For some categories, there is a small performance gain, for others there is a small loss. This contrasts with the results on the image benchmark, where a performance reduction was observed.

In conclusion, OpponentSIFT is significantly better than all other descriptors on the video benchmark (see figure 8). The corresponding invariant property is given by eq. (10).

D. Comparison with state-of-the-art

So far, the performance of single descriptors has been analyzed. It is worthwhile to investigate combinations of several descriptors, since they are not completely redundant. State-of-the-art results on the PASCAL VOC Challenge 2007 also employ combinations of several methods. Table II gives an overview of combinations on this dataset. For example, the best entry in the PASCAL VOC Challenge 2007, by Marszałek *et al.* [38], has achieved a mean average precision of 0.594 using SIFT and HueSIFT descriptors, the spatial pyramid [3], additional point sampling strategies besides Harris-Laplace such as Laplacian point sampling and dense sampling, and a feature selection scheme. When the feature selection scheme is excluded and simple flat fusion is used, Marszałek reports a mean average precision of 0.575.

To illustrate the potential of the color descriptors from table I, a simple fusion experiment has been performed with SIFT and the best four color SIFT variants (section IV-B details how the combination is constructed). To be comparable, a setting similar to Marszałek is used: both Harris-Laplace point sampling and dense sampling are employed and the same spatial pyramid is used (see figure 2 for an overview of the feature extraction pipelines used). In this setting, the best single color descriptor achieve a mean average precision 0.566. The combination gives a mean average precision of 0.605. This convincing gain of 7% suggests that the color descriptors are not entirely redundant. Compared to the intensity-based SIFT descriptor, the gain is 8%. Further gains should be possible, if the descriptors with the right amount of invariance are fused, preferably using an automatic selection strategy.

As shown in table III, similar gains are observed on the Mediamill Challenge: mean average precision increases by 7% when combinations of color descriptors are used, instead of intensity-based SIFT only. Relative to the best single color descriptor, an increase of 3% is observed. Furthermore, when the descriptors of this paper are compared to the baseline provided by the Mediamill Challenge, there is a relative improvement of 104%.

For reference, combinations of color descriptors from this paper were submitted to the PASCAL VOC 2008 benchmark [40] and the TRECVID 2008 evaluation campaign [7]. In both cases, top performance was achieved. The color descriptors as presented in this paper were the foundation of

TABLE II

IN THIS TABLE, COMBINATIONS OF DESCRIPTORS ON THE IMAGE BENCHMARK ARE COMPARED TO MARSZALEK *et al.* [38], WHO OBTAINS STATE-OF-THE-ART RESULTS ON THIS DATASET. ADDING COLOR DESCRIPTORS IMPROVES OVER INTENSITY-BASED SIFT ALONE BY 8%.

Combinations on image benchmark				
Author	Point sampling	Descriptor	Spatial pyramid	Mean average precision
<i>This paper</i>	Harris-Laplace, dense sampling	SIFT	1x1+2x2+1x3	0.558
<i>This paper</i>	Harris-Laplace, dense sampling	C-SIFT	1x1+2x2+1x3	0.566
Marszałek <i>et al.</i> [38]	Harris-Laplace, dense sampling, Laplacian	SIFT, HueSIFT, other	1x1+2x2+1x3	0.575
Marszałek <i>et al.</i> [38]	Harris-Laplace, dense sampling, Laplacian	SIFT, HueSIFT, other; with feature selection	1x1+2x2+1x3	0.594
<i>This paper</i>	Harris-Laplace, dense sampling	SIFT, OpponentSIFT, <i>rg</i> SIFT, C-SIFT, RGB-SIFT	1x1+2x2+1x3	0.605

TABLE III

IN THIS TABLE, COMBINATIONS OF DESCRIPTORS ON THE VIDEO BENCHMARK ARE COMPARED TO THE BASELINE SET BY THE MEDIAMILL CHALLENGE [22] FOR THE 39 LSCOM-LITE CATEGORIES [35]. ADDING COLOR DESCRIPTORS IMPROVES OVER INTENSITY-BASED SIFT ALONE BY 7%.

Combinations on video benchmark				
Author	Point sampling	Descriptor	Spatial pyramid	Mean average precision
Snoek <i>et al.</i> [22]	Grid	Weibull [39]	1x1	0.250
<i>This paper</i>	Harris-Laplace, dense sampling	SIFT	1x1+2x2+1x3	0.476
<i>This paper</i>	Harris-Laplace, dense sampling	OpponentSIFT	1x1+2x2+1x3	0.494
<i>This paper</i>	Harris-Laplace, dense sampling	SIFT, OpponentSIFT, <i>rg</i> SIFT, C-SIFT, RGB-SIFT	1x1+2x2+1x3	0.510

TABLE IV

IN THIS TABLE, RESULTS OF DESCRIPTOR COMBINATIONS FROM THIS PAPER AS SUBMITTED TO THE CLASSIFICATION TASK OF THE PASCAL VOC CHALLENGE 2008 [40] ARE SHOWN.

PASCAL VOC 2008 evaluation: best overall performance				
Author	Point sampling	Descriptor	Spatial pyramid	Mean average precision
<i>This paper</i> and Tahir <i>et al.</i> [41]	Harris-Laplace, dense sampling	SIFT, OpponentSIFT, <i>rg</i> SIFT, C-SIFT, RGB-SIFT	1x1+2x2+1x3	0.549

TABLE V

IN THIS TABLE, RESULTS OF DESCRIPTOR COMBINATIONS FROM THIS PAPER AS SUBMITTED TO THE NIST TRECVID 2008 VIDEO BENCHMARK [7] ARE SHOWN.

NIST TRECVID 2008 evaluation: best overall performance				
Author	Point sampling	Descriptor	Spatial pyramid	Inferred mean average precision
<i>This paper</i> and Snoek <i>et al.</i> [43]	Harris-Laplace, dense sampling	SIFT, OpponentSIFT, <i>rg</i> SIFT, C-SIFT, RGB-SIFT	1x1+2x2+1x3	0.194

these submissions. For additional details, see table IV [41], [42] and table V [43].

E. Discussion

Using the ALOI dataset, the theoretical invariance properties of color descriptors were verified experimentally. However, possessing invariance properties alone is not sufficient to address category recognition: the descriptor should also be distinctive and robust to compression artifacts. Several histogram-based descriptors and color moment invariants were found to be sensitive to even moderate amounts of compression, thereby reducing their usefulness. On the other hand, the results show that the SIFT descriptor and most color extensions of the SIFT descriptor are robust to compression artifacts. Also, these SIFT-based descriptors outperform histogram-based and moment-based descriptors on both image and video category

recognition. Therefore, the rest of this discussion will focus on the properties of these descriptors in particular.

The results on two category recognition benchmarks show that SIFT-based descriptors which perform well are all invariant to light intensity changes. For light intensity shifts, the usefulness of invariance depends on the object or scene category. For those categories in real-world datasets where large variations in lighting conditions occur frequently, invariance to light intensity shifts is useful. Examples for the image benchmark are shown in figure 9: normally, sofas are found indoor. However, the dataset contains samples where the sofa is photographed outside on the street. As the ranking positions show, the OpponentSIFT descriptor, which is invariant to both light intensity changes and shifts, places these samples higher in the ranking. However, the converse also occurs, as the example of the potted plants shows. The descriptors

Positions in Rankings for Image Benchmark

				
Color Descriptor				
OpponentSIFT	769	1053	21	190
C-SIFT	1782	2813	103	591
<i>rg</i> SIFT	3075	1445	161	486
RGB-SIFT	1917	3522	6	11

			
Color Descriptor			
OpponentSIFT	194	709	1583
C-SIFT	8	19	43
<i>rg</i> SIFT	10	18	63
RGB-SIFT	264	2627	706

Fig. 9. From the PASCAL VOC Challenge 2007 [21], several positive examples for the object categories sofa, bus and potted plant are shown, together with their position in the ranked list of category recognition results for four different color descriptors. If, for one or more color descriptors, the ranked position is notably better than for the other color descriptors, it has been bold-faced. The ranking has 4952 elements.

Positions in Rankings for Video Benchmark





	Building	Building	Vegetation	Vegetation
				
Color Descriptor				
OpponentSIFT	26	34	1035	304
C-SIFT	677	2719	53	1
rgSIFT	1113	1512	111	46
RGB-SIFT	102	35	954	921

Fig. 10. From the Mediamill Challenge [22], several positive examples for the categories building and vegetation are shown, together with their position in the ranked list of category recognition results for four different color descriptors. If, for one or more color descriptors, the ranked position is notably better than for the other color descriptors, it has been bold-faced. The ranking has 12914 elements.

which are only scale-invariant place the samples higher in the ranking, and the shift-invariant OpponentSIFT and RGB-SIFT descriptors lag behind. For the video benchmark, figure 10 shows similar examples of both phenomena for buildings and vegetation.

From the results, it can be noticed that invariance to light color changes and shifts is domain-specific. For the image dataset, a significant reduction in performance was observed, whereas for the video dataset, there was no performance difference. However, there are specific samples where invariance to light color changes provides a benefit. An example is shown in figure 9 for busses: the bus illuminated by a setting sun benefits from light color invariance, as does the bus illuminated by red light tubes. Invariance to light intensity changes and shifts is not sufficient for the latter sample. However, the overall performance is not improved by light color invariance,

TABLE VI

THE RECOMMENDED CHOICE OF DESCRIPTORS FOR DIFFERENT DATASETS: THE PASCAL VOC 2007, MEDIAMILL CHALLENGE AND DATASETS WHERE NO PRIOR KNOWLEDGE ABOUT THE LIGHTING CONDITIONS OR THE OBJECT AND SCENE CATEGORIES IS AVAILABLE. WITHOUT SUCH PRIOR KNOWLEDGE, OPPONENTSIFT IS THE BEST CHOICE.

Recommended Color Descriptors Per Dataset

PASCAL VOC 2007	Mediamill Challenge	Unknown Data
1. C-SIFT	1. OpponentSIFT	1. OpponentSIFT
2. OpponentSIFT	2. RGB-SIFT	2. C-SIFT
3. RGB-SIFT	3. C-SIFT	3. RGB-SIFT
4. SIFT	4. SIFT	4. SIFT

presumably because light color changes are quite rare in both benchmarks due to the white balancing performed during data recording.

Overall, when choosing a single descriptor and no prior knowledge about the dataset and object and scene categories is available, the best choice is OpponentSIFT. The corresponding invariance property is scale- and shift-invariance, given by eq. (10). Second best is C-SIFT for which the corresponding invariance property is scale-invariance, given by eq. (8). Table VI summarizes the recommendations for the datasets from this paper and datasets where no prior knowledge is available.

To obtain state-of-the-art performance on real-world datasets with large variations in lighting conditions, multiple color descriptors should be chosen, each one with a different amount of invariance. As shown earlier, even a simple combination of color descriptors improves over the individual descriptors, suggesting that they are not completely redundant. This is illustrated by the keyframes shown in figure 10: depending on the visual category, the OpponentSIFT and C-SIFT descriptors both show their strong points. Results on the two categorization benchmarks have shown that the choice of a single descriptor for all categories is suboptimal (see figure 7). While the addition of color improves category recognition by 8–10% over intensity-based SIFT only, further gains should be possible if the descriptor with the appropriate amount of invariance is selected per category, using either a feature selection strategy or domain knowledge.

VI. CONCLUSION

In this paper, the invariance properties of color descriptors are studied using a taxonomy of invariance with respect to photometric transformations, see table I for an overview. These invariance properties were validated using a dataset with known photometric changes. In addition, the distinctiveness of color descriptors is assessed experimentally using two benchmarks from the image domain and the video domain. On these benchmarks, the addition of color descriptors over SIFT improves category recognition by 8% and 7%, respectively.

From the theoretical and experimental results, it can be derived that invariance to light intensity changes and light color changes affects object and scene category recognition. The results reveal further that, for light intensity shifts, the usefulness of invariance is category-specific. Therefore, a color descriptor with an appropriate level of invariance should be selected for automated recognition of individual object and scene categories. Overall, when choosing a single descriptor and no prior knowledge about the dataset and object and scene categories is available, the OpponentSIFT is recommended. Finally, a proper combination of color descriptors improves over the individual descriptors.

ACKNOWLEDGMENTS

This work was supported by the EC-FP6 VIDI-Video project.

REFERENCES

- [1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 1–60, 2008.
- [2] R. Fergus, F.-F. Li, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *IEEE International Conference on Computer Vision*, Beijing, China, 2005, pp. 1816–1823.
- [3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, New York, USA, 2006, pp. 2169–2178.
- [4] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content-based image retrieval," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 133–157, 2007.
- [5] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.
- [6] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo, "Large-Scale Multimodal Semantic Concept Detection for Consumer Video," in *ACM International Workshop on Multimedia Information Retrieval*, Augsburg, Germany, 2007, pp. 255–264.
- [7] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID," in *ACM International Workshop on Multimedia Information Retrieval*, Santa Barbara, USA, 2006, pp. 321–330.
- [8] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *ACM International Conference on Image and Video Retrieval*, Amsterdam, The Netherlands, 2007, pp. 494–501.
- [9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1–2, pp. 43–72, 2005.
- [11] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.
- [12] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: A SIFT descriptor with color invariant characteristics," in *IEEE Conference on Computer Vision and Pattern Recognition*, New York, USA, 2006, pp. 1978–1983.
- [13] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts, "Color invariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1338–1350, 2001.
- [14] J. van de Weijer, T. Gevers, and A. Bagdanov, "Boosting color saliency in image feature detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 150–156, 2006.
- [15] G. J. Burghouts and J. M. Geusebroek, "Performance evaluation of local color invariants," *Computer Vision and Image Understanding*, vol. 113, pp. 48–62, 2009.
- [16] A. Bosch, A. Zisserman, and X. Muoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 04, pp. 712–727, 2008.
- [17] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Spectral sharpening: sensor transformations for improved color constancy," *Journal of the Optical Society of America A*, vol. 11, no. 5, p. 1553, 1994.
- [18] J. von Kries, "Influence of adaptation on the effects produced by luminous stimuli," in *MacAdam, D.L. (Ed.), Sources of Color Vision*. MIT Press, Cambridge, MS., 1970.
- [19] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluation of color descriptors for object and scene recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA, June 2008.
- [20] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.
- [21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." [Online]. Available: <http://www.pascal-network.org/challenges/VOC/voc2007/>
- [22] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *ACM International Conference on Multimedia*, Santa Barbara, USA, 2006, pp. 421–430.
- [23] M. Shafer, "Using color to separate reflection components," *Color Research and Applications*, vol. 10, no. 4, pp. 210–218, 1985.
- [24] G. D. Finlayson, S. D. Hordley, and R. Xu, "Convex programming colour constancy with a diagonal-offset model," in *IEEE International Conference on Image Processing*, 2005, pp. 948–951.

- [25] T. Gevers, J. van de Weijer, and H. Stokman, *Color image processing: methods and applications: color feature detection: an overview*. CRC press, 2006, ch. 9, pp. 203–226.
- [26] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons, “Moment invariants for recognition under changing viewpoint and illumination,” *Computer Vision and Image Understanding*, vol. 94, no. 1-3, pp. 3–27, 2004.
- [27] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004.
- [28] P.-E. Forssén, “Maximally stable colour regions for recognition and matching,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007.
- [29] J. Sivic and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in *IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1470–1477.
- [30] T. K. Leung and J. Malik, “Representing and recognizing the visual appearance of materials using three-dimensional textons,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.
- [31] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 264–271.
- [32] F. Jurie and B. Triggs, “Creating efficient codebooks for visual recognition,” in *IEEE International Conference on Computer Vision*, Beijing, China, 2005, pp. 604–610.
- [33] B. Leibe and B. Schiele, “Interleaved object categorization and segmentation,” in *British Machine Vision Conference*, Norwich, UK, 2003, pp. 759–768.
- [34] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [35] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, “Large-scale concept ontology for multimedia,” *IEEE Multimedia*, vol. 13, no. 3, pp. 86–91, 2006.
- [36] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, August 2006.
- [37] B. Efron, “Bootstrap methods: Another look at the jackknife,” *Annals of Statistics*, vol. 7, pp. 1–26, 1979.
- [38] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer, “Learning object representations for visual object class recognition,” 2007, Visual Recognition Challenge workshop, in conjunction with IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil. [Online]. Available: <http://lear.inrialpes.fr/pubs/2007/MSHV07>
- [39] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders, “Robust scene categorization by learning image statistics in context,” in *CVPR Workshop on Semantic Learning Applications in Multimedia (SLAM)*, 2006.
- [40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results.” [Online]. Available: <http://www.pascal-network.org/challenges/VOC/voc2008/>
- [41] M. A. Tahir, K. E. A. van de Sande, J. R. R. Uijlings, and *et al.*, “University of Amsterdam and University of Surrey at PASCAL VOC 2008,” 2008, PASCAL Visual Object Classes Challenge Workshop, in conjunction with IEEE European Conference on Computer Vision, Marseille, France. [Online]. Available: <http://staff.science.uva.nl/~ksande/pub/vandesande-pascalvoc2008.pdf>
- [42] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, “Visual word ambiguity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
- [43] C. G. M. Snoek, K. E. A. van de Sande, O. de Rooij, B. Huurnink, J. C. van Gemert, J. R. R. Uijlings, and *et al.*, “The MediaMill TRECVID 2008 semantic video search engine,” in *Proceedings of the 6th TRECVID Workshop*, Gaithersburg, USA, November 2008.



Koen van de Sande Koen E.A. van de Sande received a BSc in Computer Science (2004), a BSc in Artificial Intelligence (2004) and a MSc in Computer Science (2007) from the University of Amsterdam, The Netherlands. Currently, he is pursuing the PhD degree at the University of Amsterdam. His research interests include computer vision, visual categorization, (color) image processing, statistical pattern recognition and large-scale benchmark evaluations. He is a co-organizer of the annual VideOlympics. He is a student member of the IEEE.



Theo Gevers Theo Gevers is an Associate Professor of Computer Science at the University of Amsterdam, The Netherlands. At the University of Amsterdam he is a teaching director of the MSc of Artificial Intelligence. He currently holds a VICI-award (for excellent researchers) from the Dutch Organisation for Scientific Research. His main research interests are in the fundamentals of content-based image retrieval, colour image processing and computer vision specifically in the theoretical foundation of geometric and photometric invariants. He is co-chair of the Internet Imaging Conference (SPIE 2005, 2006), co-organizer of the First International Workshop on Image Databases and Multi Media Search (1996), the International Conference on Visual Information Systems (1999, 2005), the Conference on Multimedia & Expo (ICME, 2005), and the European Conference on Colour in Graphics, Imaging, and Vision (CGIV, 2012). He is guest editor of the special issue on content-based image retrieval for the International Journal of Computer Vision (IJCV 2004) and the special issue on Colour for Image Indexing and Retrieval for the journal of Computer Vision and Image Understanding (CVIU 2004). He has published over 100 papers on colour image processing, image retrieval and computer vision. He is program committee member of a number of conferences, and an invited speaker at major conferences. He is a lecturer of post-doctoral courses given at various major conferences (CVPR, ICPR, SPIE, CGIV). He is member of the IEEE.



Cees Snoek Cees G.M. Snoek received the MSc degree in business information systems (2000) and the PhD degree in computer science (2005) both from the University of Amsterdam, where he is currently a senior researcher at the Intelligent Systems Lab. He was a visiting scientist at Carnegie Mellon University, in 2003. His research interests focus on visual categorization, statistical pattern recognition, social media retrieval, and large-scale benchmark evaluations, especially when applied in combination for video search. He has published over 70 refereed book chapters, journal and conference papers in these fields, and serves on the program committee of several conferences. Dr. Snoek is the lead researcher of the award-winning MediaMill Semantic Video Search Engine, which is a consistent top performer in the yearly NIST TRECVID evaluations. He is co-initiator and co-organizer of the annual VideOlympics and a lecturer of post-doctoral courses given at international conferences and summer schools. He is a member of the IEEE. Dr. Snoek received a young talent (VENI) grant from the Netherlands Organization for Scientific Research in 2008.