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 a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. 光照 *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.*

Accepted for publication in the *International Journal of Computer Vision*, 2004.

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 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

找出图片中对尺度变化"不变"的位置

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).

高斯核是唯一可能的尺度空间的核

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:

乘法的

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

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.

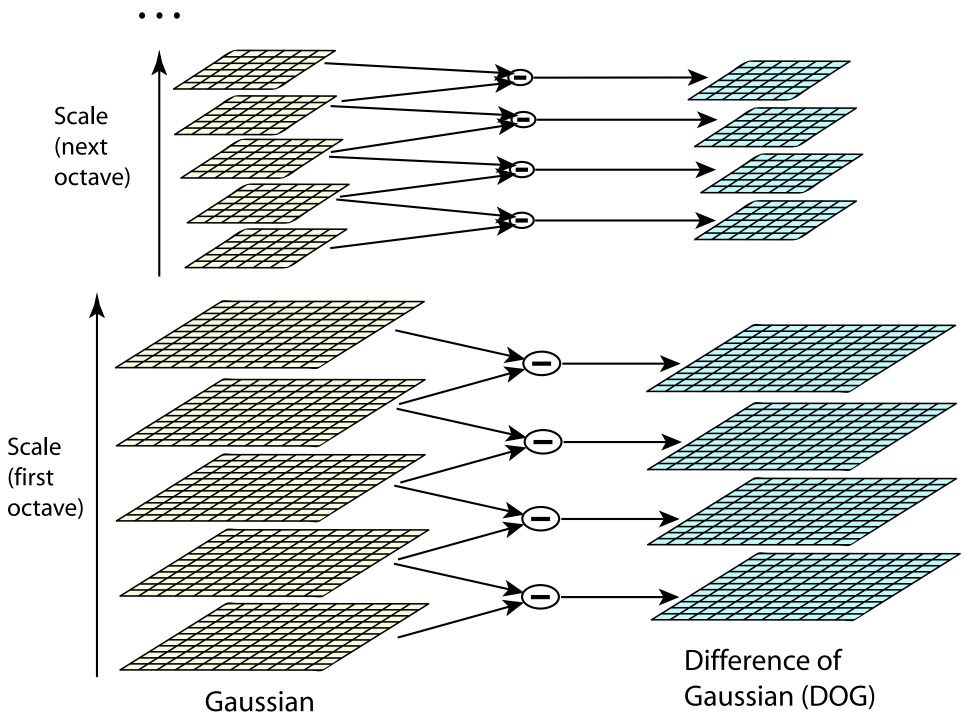

Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled 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, Mikolajczyk (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 $t = \sigma^2$):

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

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,

常数，不影响尺度空间极值点位置.
已满足σ^2尺度正则要求

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

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

DOG ≈ (k-1) LOG

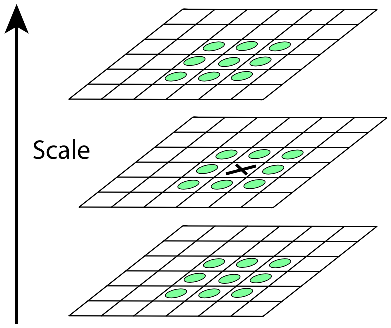

Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

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.

$\sigma(s) = \sigma_0 \cdot 2^{s/n}$

$= \sigma_0 \cdot 2^{n-1}(1, k^1, k^2, \cdots, k^{n-1})$

n为每个octave内interval层数.

| | | | : 4

| | | : 3 ↓ DOG数

| : 1

相邻3个DOG求极值点.

故由4→1，少了3个。于是生成s+3张图

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

没懂.

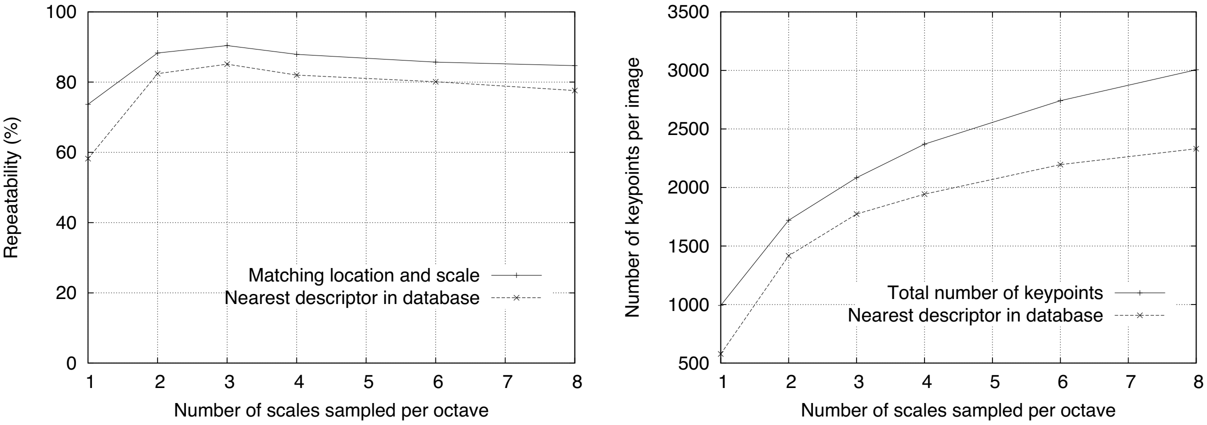

Figure 3: The top line of the first graph shows the percent of keypoints that are repeatably detected at the same location and scale in a transformed image as a function of the number of scales sampled per octave. The lower line shows the percent of keypoints that have their descriptors correctly matched 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 人造的, 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 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).

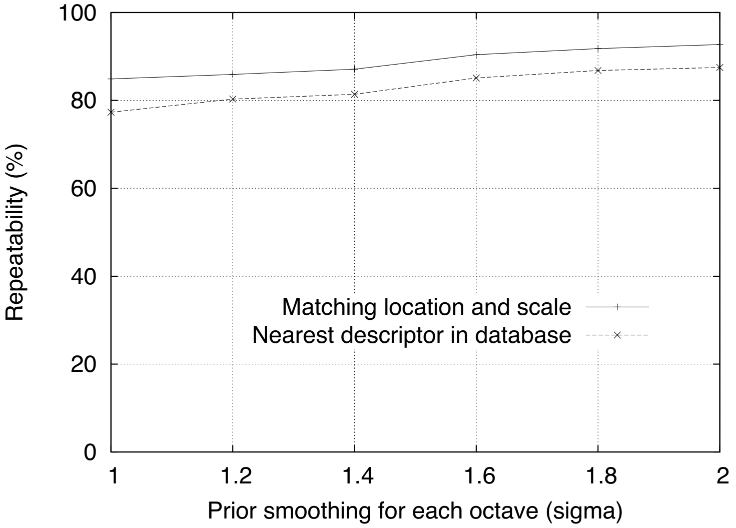

Figure 4: The top line in the graph shows the percent of keypoint locations that are repeatably detected in a transformed image as a function of the prior image smoothing for the first level of each octave. The lower line shows the percent of descriptors correctly matched against a large database.

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.

粗糙的

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.

预先高斯平滑.

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.

建立金字塔前先把第一层的图像通过线性插值扩大一倍.

离散空间的极值点并不是真正的极值点.
这一节利用离散空间点插值得到连续的空间极值点.

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.

剔除不稳定极值点(低对比度,因为对噪声敏感)

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:

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

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

$\hat{x}$: 插值中心的极值点.

$$\frac{\partial D}{\partial x} = 0 \Rightarrow \hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}. \qquad (3)$$

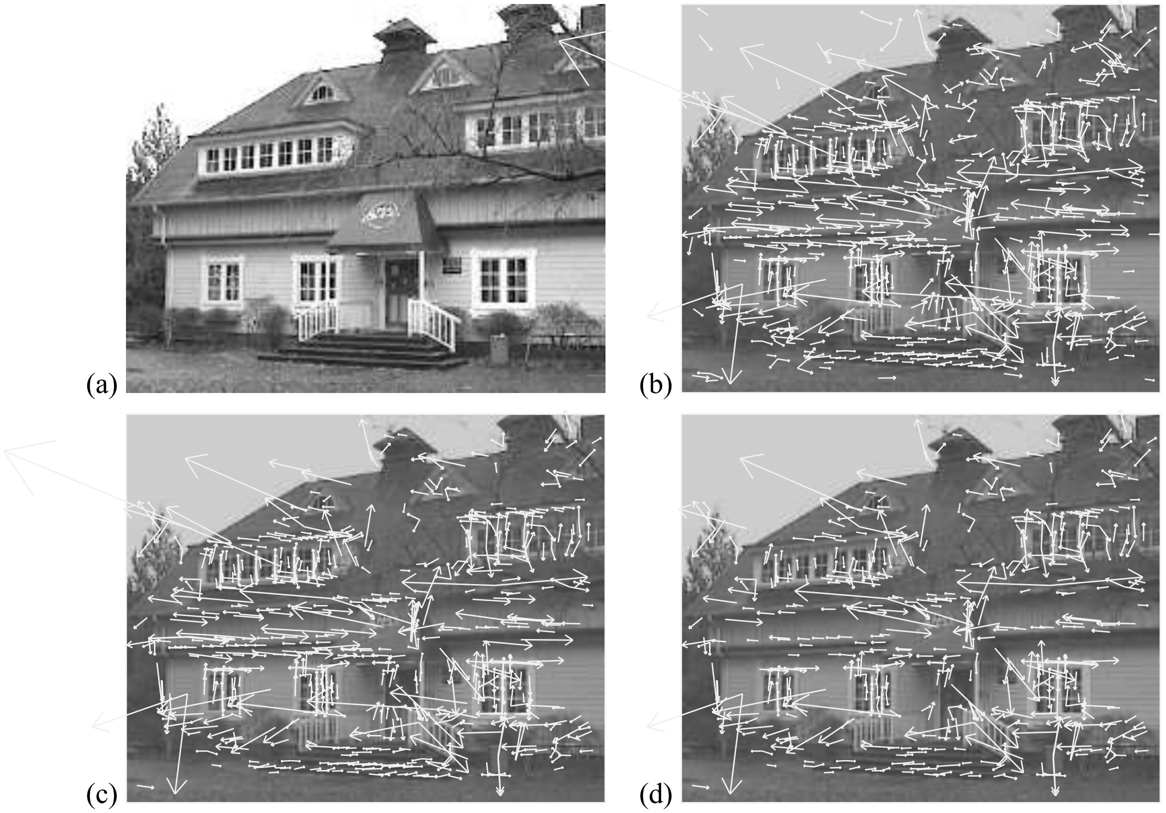

Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a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

$\hat{x}$>0.5时(x或y方向),必须改变关键点位置.

x或y方向

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

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

$|D(\hat{x})| < 0.03$

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

D0G具有很强的边缘响应，因此要去除.

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, **H**, computed at the location and scale of the keypoint:

主曲率

垂直的

不好的DoG极值点在横跨边缘处有很大的主曲率，在垂直方向有很小的主曲率.

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

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

The eigenvalues of **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 **H** and their product from the determinant:

H特征值与D主曲率成比例

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

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,

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

→ 关于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

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

$\frac{Tr^2}{Det} < (\frac{r+1}{r})^2$ 时保留

经验取值

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

实现对图像旋转的不变性.

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:

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

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

方向直方图. 36个柱, 数值为按m和高斯分布加权

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.

∴邻域窗口半径: $3 \times 1.5\sigma(s)$

高斯加权 圆窗: 3σ原则

$1.5\sigma(s)$

只保留峰值大于主方向峰值80%的方向

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.

抛物线

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.

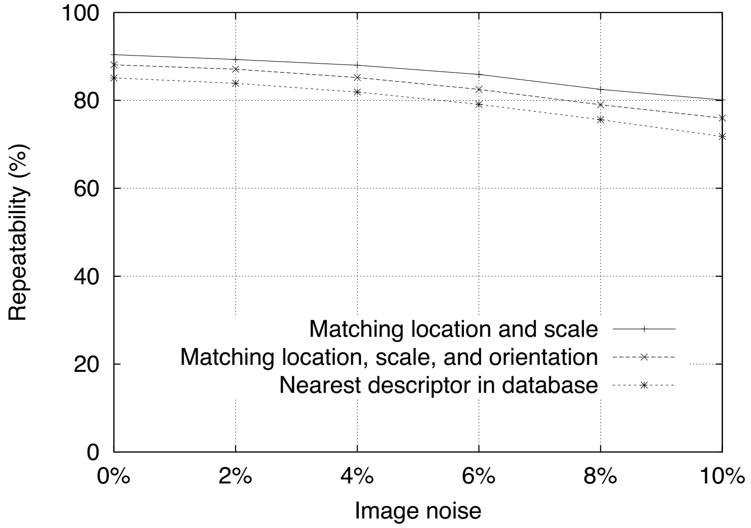

Figure 6: The top line in the graph shows the percent of keypoint locations and scales that are repeatably detected as a function of pixel noise. The second line shows the repeatability after also requiring agreement in orientation. The bottom line shows the final percent of descriptors correctly matched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

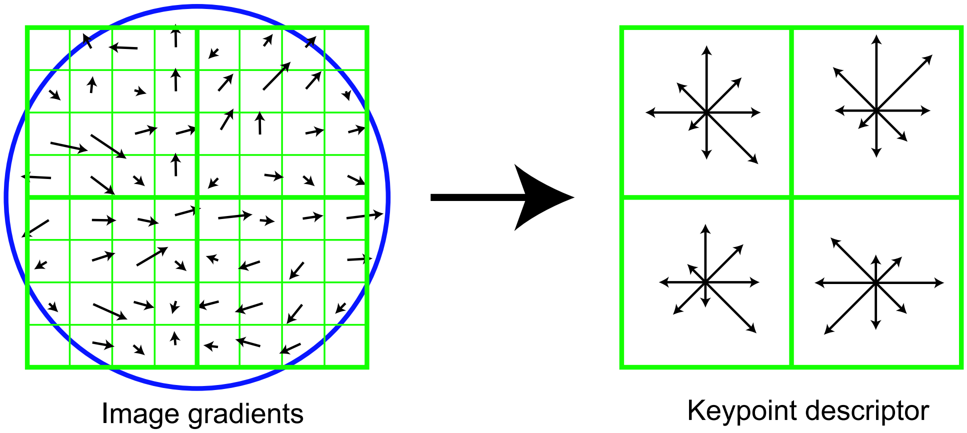

Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 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 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 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 4x4 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 2x2 array of orientation histograms, whereas our experiments below show that the best results are achieved with a 4x4 array of histograms with 8 orientation bins in each. Therefore, the experiments in this paper use a 4x4x8 = 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 4x4 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 5x5 and higher descriptor sizes. Throughout this paper we use a 4x4 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.

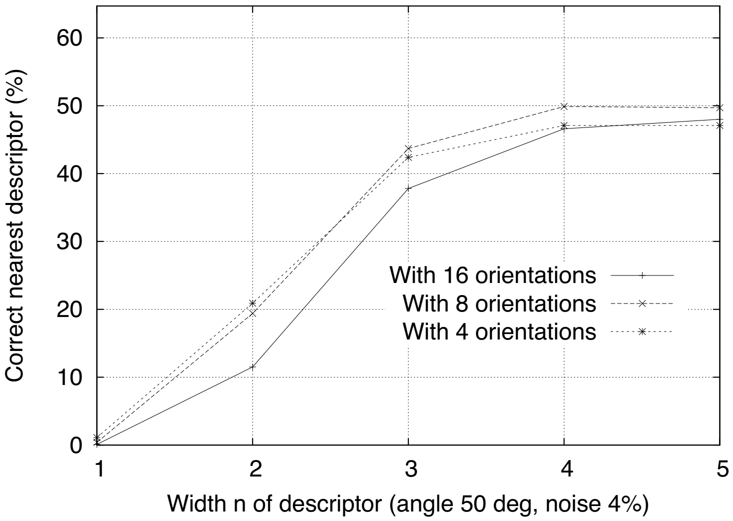

Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and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

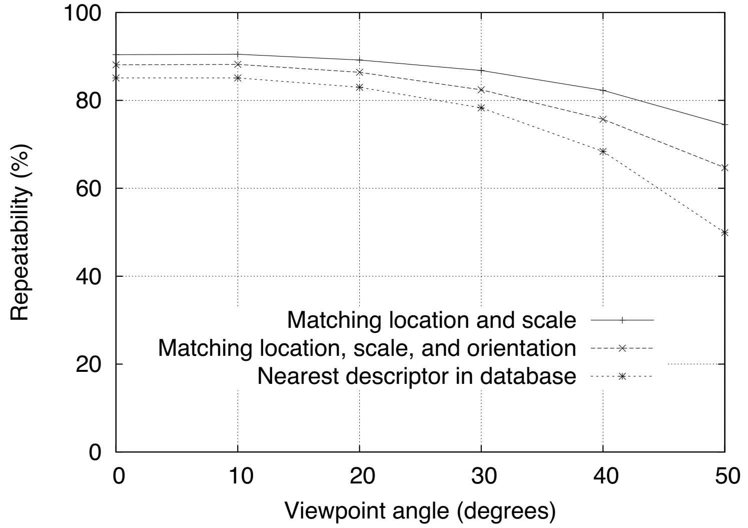

Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

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.

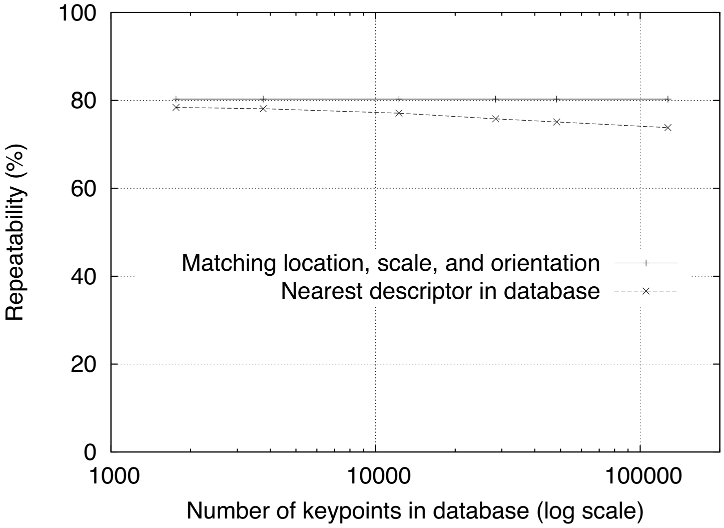

Figure 10: The dashed line shows the percent of keypoints correctly matched to a database as a function of database size (using a logarithmic scale). The solid line shows the percent of keypoints assigned the correct location, scale, and orientation. Images had random scale and rotation changes, an affine 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 have 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

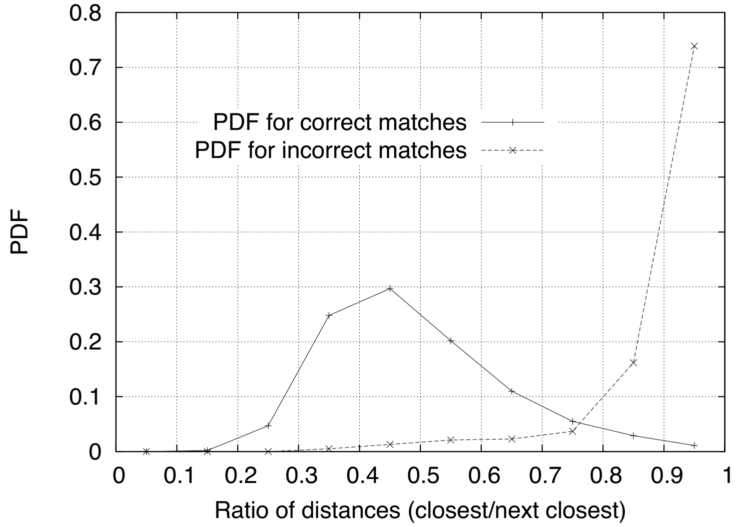

Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches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

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

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

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

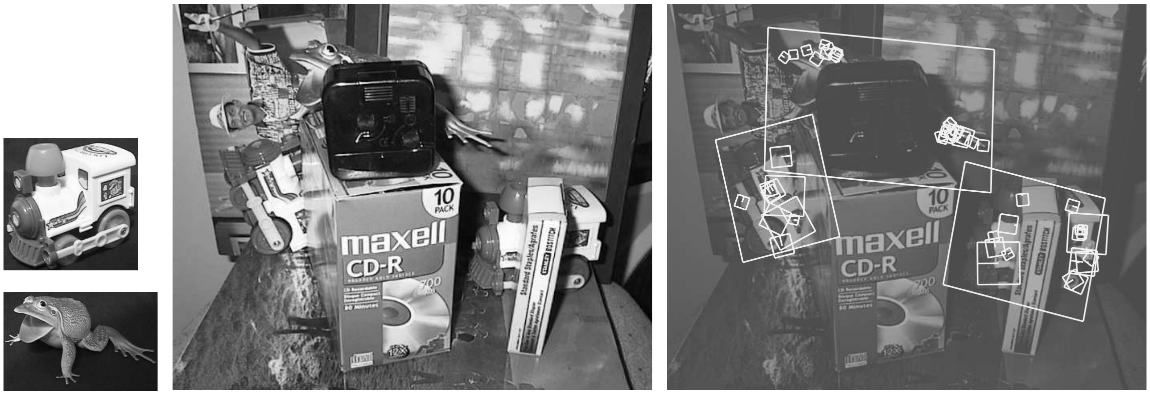

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original 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,

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

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.

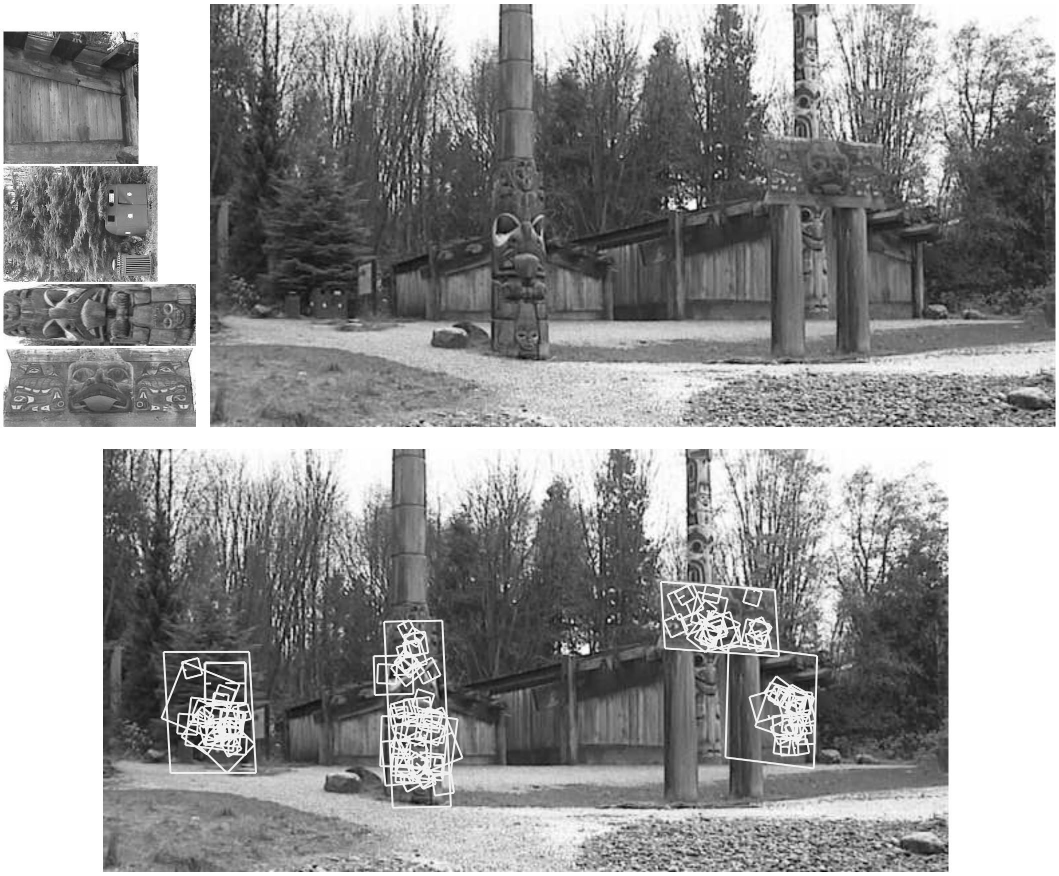

Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as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