

Review on Distinctive Image Features from Scale-Invariant Key-Points

Sonali S. Thangan, Ankit R. Mune

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-neighbour 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.

Keywords: Extracting Distinctive, Approach, Real-Time Performance, the Features Are Highly Distinctive,

I. 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[1]. 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:

Revised Version Manuscript Received on April 30, 2018.

Sonali S. Thangan, M.E. Student, Department of Computer Science & Engineering, Dr. Rajendra Gode Institute of Technology & Research, Amravati (Maharashtra)-444602, India.

Prof. Ankit R. Mune, Department of Computer Science & Engineering, Dr. Rajendra Gode Institute of Technology & Research, Amravati (Maharashtra)-444602, India.

A. 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.

B. Key Point Localization:

At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

C. 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.

D. 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-neighbour 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[4].

II. 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 repeatedly 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 [2]. 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 :

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

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[5].

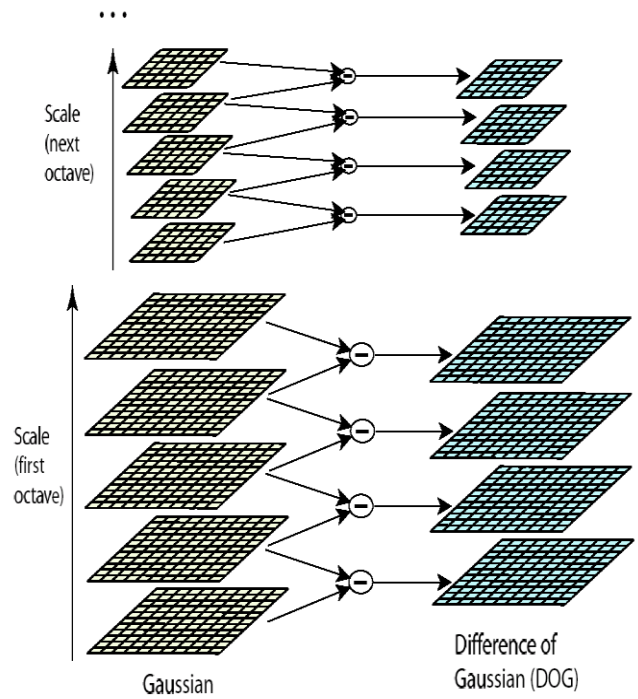


Figure 1: For Each Octave of Scale Space, the Initial Image is repeatedly Convolved with Gaussians to Produce the set of Scale Space Images Shown on the left. Adjacent Gaussian images are subtracted to produce the Difference-of-Gaussian Images on the Right. After each Octave, the Gaussian Image is Down-Sampled 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[8].

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

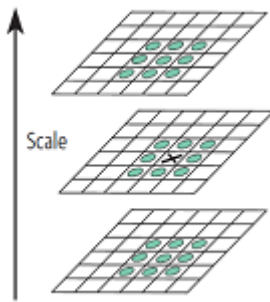


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

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 $k = 2^{1/s}$ [9]. 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[10].

III. ACCURATE KEYPOINT LOCALIZATION

Once a keypoint candidate has been found by comparing a pixel to its neighbours, 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[11]. 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:

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

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

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$

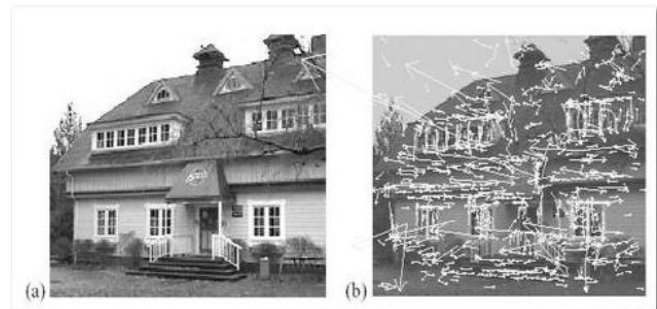


Figure 3: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location.

As suggested by Brown, the Hessian and derivative of D are approximated by using differences of neighbouring 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 3(a) shows the original image, which is shown at reduced contrast behind the subsequent figures. Figure 3 (b) shows the 832 keypoints at all detected maxima 11 and minima of the difference-of-Gaussian function[19].

IV. RECOGNITION EXAMPLES



Figure 4: 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.

Above figure 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. We have implemented these algorithms on a laptop computer with attached video camera, and have tested them extensively over a wide range of conditions[10]. 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[15].

V. 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, epi-polar 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[20]. 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[21].

REFERENCES

1. Arya, S., and Mount, D.M. 1993. Approximate nearest neighbour queries in fixed dimensions. In Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'93), pp. 271-280.
2. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., and Wu, A.Y. 1998. An optimal algorithm for approximate nearest neighbour searching. *Journal of the ACM*, 45:891-923.
3. Ballard, D.H. 1981. Generalizing the Hough transform to detect arbitrary patterns. *Pattern Recognition*, 13(2):111-122.
4. Basri, R., and Jacobs, D.W. 1997. Recognition using region correspondences. *International Journal of Computer Vision*, 25(2):145-166.
5. 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.
6. 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.
7. Brown, M. and Lowe, D.G. 2002. Invariant features from interest point groups. In British Machine Vision Conference, Cardiff, Wales, pp. 656-665.
8. Carneiro, G., and Jepson, A.D. 2002. Phase-based local features. In European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp. 282-296.
9. 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.
10. Edelman, S., Intrator, N. and Poggio, T. 1997. Complex cells and object recognition. Unpublished manuscript: <http://kybele.psych.cornell.edu/~edelman/archive.html>
11. Fergus, R., Perona, P., and Zisserman, A. 2003. Object class recognition by unsupervised scaleinvariant learning. In IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, pp. 264-271.
12. 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.
13. Funt, B.V. and Finlayson, G.D. 1995. Color constant color indexing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(5):522-529.
14. Grimson, E. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*, The MIT Press: Cambridge, MA.
15. Harris, C. 1992. Geometry from visual motion. In *Active Vision*, A. Blake and A. Yuille (Eds.), MIT Press, pp. 263-284.
16. Harris, C. and Stephens, M. 1988. A combined corner and edge detector. In Fourth Alvey Vision Conference, Manchester, UK, pp. 147-151.
17. Hartley, R. and Zisserman, A. 2000. *Multiple view geometry in computer vision*, Cambridge University Press: Cambridge, UK.
18. 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.
19. 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.
20. Lindeberg, T. 1994. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224-270.
21. Lowe, D.G. 1991. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441-450.