Performance Evaluation of Scale-Interpolated Hessian-Laplace and Haar Descriptors for Feature Matching

Akshay Bhatia, Robert Laganière
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
abhatia,laganier@site.uottawa.ca

Gerhard Roth
Institute for Information Technology
National Research Council of Canada
Ottawa, Ontario, Canada, K1A 0R6
Gerhard.Roth@nrc-cnrc.gc.ca

Abstract

This paper studies the performance of various scale-invariant detectors in the context of feature matching. In particular, we propose an implementation of the Hessian-Laplace operator that we called Scale-Interpolated Hessian-Laplace. This research also proposes to use Haar descriptors which are derived from the Haar wavelet transform. It offers the advantage of being computationally inexpensive and smaller in size when compared to other descriptors.

1. Introduction

A common step in most computer vision algorithms requires representing image content in terms of features. These features which represent specific visual patterns can be used to identify corresponding structures between images. In the last decade, a lot of research has been done to study the properties of invariant features. These are features detected and described in a way that is invariant to scale and affine changes in the image. The rapid development in the domain of invariant features has led to a significant improvement in the performance of recognition algorithms.

In this paper, we propose a comparative study of the performance of the Scale-Interpolated Hessian-Laplace to detect feature points. The detector uses the Hessian matrix to locate points in the image plane and a Laplacian function to compute scale for those points. A localization step ensures that the location and scale of the points detected is close to their true location. We also introduce Haar descriptors which are based on the Haar wavelet transform. Haar descriptors offer the advantage of being compact and easy to compute.

2 Scale Invariant Features

The concept of scale-space representation was introduced by Witkin[12] for representing one dimensional signals at multiple scales. The scale-space representation for an image is built by convolving the image with different size of kernels. The scale parameter associated with each image is directly related to the σ value of the kernel convolved with that image. Various studies in the literature have shown that Gaussian kernel is the most optimal kernel to build such a representation.

Using Gaussian kernels, the process of generating a scale-space representation can be mathematically expressed as

$$G_n(x, y) = g(x, y, \sigma_n) * I(x, y) \tag{1}$$

where $G_n(x, y)$ denotes the n^{th} level Gaussian image in the scale-space representation and $g(x, y, \sigma_n)$ is the two dimensional Gaussian kernel given and σ_n corresponds to the standard deviation of the kernel at the n^{th} scale, with:

$$\sigma_n = s^{n-1}\sigma_1 \tag{2}$$

where s denotes the scale ratio between adjacent images.

Scale-invariant features are extracted using multi-scale image representations where image points are associated with a scale parameter by searching for maxima of some function. The scale parameter thus obtained is used to assign a circular region to each feature point. Hence, unlike ordinary features, scale invariant features have associated regions. These regions are later used to generate descriptors for these feature points which are eventually used to match feature points between images.

Lindeberg[5] proposed a method for detecting blobs like features in a scale-space representation. In order to detect points and compute their scale, a search for 3D maxima of scale normalized Laplacian of Gaussian is performed. Lowe[6] proposed a scale invariant detector based on a multi-scale representation constructed using differences of

Gaussian images. Interest points which correspond to blobs are detected by looking for 3D extrema of the difference of Gaussian function. The Harris matrix has been frequently explored to detect points which are scale invariant. Mikolajczyk and Schmid[7] extended the detector by combining it with the Laplacian function to form the Harris-Laplace detector. For Harris points computed at different scales, a scale normalized Laplacian response is calculated over all the scales. The local extrema of this response is then used to select the scale for a feature.

3 Scale Interpolated Hessian-Laplace Detector

The Hessian matrix is composed of second order partial derivatives derived from Taylor series expansion. This matrix has been frequently used to analyze local image structures. The 2x2 Hessian matrix can be expressed as:

$$H = \begin{bmatrix} I_{xx}(x;\sigma_D) & I_{xy}(x;\sigma_D) \\ I_{yx}(x;\sigma_D) & I_{yy}(x;\sigma_D) \end{bmatrix} \quad (3)$$

where I_{xx}, I_{yy} and I_{xy} are the second order derivatives computed using Gaussian kernels of standard deviation σ_D.

The determinant of the Hessian matrix can be used to detect image structures which have strong signal variations in two directions. The scale-space representation is built by convolving the image with Gaussians of increasing size [8]. The scale of a scale-space image is equal to the standard deviation of Gaussian kernel used to generate that image.

3.1 Scale Selection

Once we have the spatial location of points detected on different levels of the scale-space representation, the next stage involves computing the proper scale for these points. The scales where the description of the image points convey the maximum information are termed as characteristic scales. A number of previous experiments have shown that the Laplacian function is the most suitable function for detecting the characteristic scale for an image structure. The scale normalized Laplacian function can be expressed as:

$$Laplacian(x;\sigma_D) = \sigma_D^2 |I_{xx}(x;\sigma_D) + I_{yy}(x;\sigma_D)| \quad (4)$$

where I_{xx} and I_{yy} are second order derivatives. One of the advantages of using the Hessian matrix is evident here as the Laplacian function corresponds to the trace of the Hessian matrix.

For a point detected in a scale-space image, its Laplacian is computed over all scales and the scale for which the Laplacian attains a local maximum is assigned as the characteristic scale. Local maximum here corresponds to response for a given scale being greater than its adjacent scales and above a given threshold. This is the strategy used in the regular Hessian-Laplace operator.

3.2 Keypoint Localization through scale interpolation

The keypoints detected using the previous approach will not be detected precisely at signal changes but in the neighborhood of those changes. This drift will cause the spatial location of a point to move away from its true location. To localize points both in scale and space, it is better if this procedure is carried out simultaneously.

Brown and Lowe[1] use a 3D quadratic function to estimate the new location and scale of a point using an iterative procedure. This procedure was used by Lowe[6] to localize points detected with difference of Gaussian detector.

An observation that can be made from a scale-space representation is that for large scale values the scale difference between successive images is greater. This causes the localization error to increase for larger scales. This error in the scale value also affects the computation of orientation and descriptor as the these operations require the selection of an image patch around the point that is proportional to its scale value. Hence, here we focus more on choosing the correct scale for a point than localizing it in the spatial domain. Given a point at a scale image s, we fit a parabola between the image s and its adjacent scale images. The scale for which the parabola attains a maximum is selected as the new scale for the point. We also compute the sub pixel location of the point using bilinear interpolation in a 3x3 neighborhood. This interpolation is performed on the scale-space image where the point was originally detected. Even though the Hessian-Laplace points obtained using this scale-interpolation method are not perfectly localized in space (2D location in the image plane), these points are well localized in scale which we will show being crucial for feature matching.

4 Repeatability Tests

Repeatability is one of the most important criteria used for evaluating the stability of feature detectors. It measures the ability of a detector to extract the same feature points across images irrespective of imaging conditions. For two images having n_1 and n_2 feature points, the repeatability rate is defined as:

$$Repeatability\,rate = \frac{n_3}{min(n_1, n_2)} \quad (5)$$

where n_3 is the number of repeatable feature points computed between the two images using n_1 and n_2.

In the context of scale invariant detectors, the scale parameter also has to be incorporated into the computation of

repeatability. The scale of a point is used to associate a region with a point proportional to its scale value. The overlap error for two points can be expressed as:

$$Overlap\ Error = |1 - s^2 \frac{min(\sigma_1^2, \sigma_2^2)}{max(\sigma_1^2, \sigma_2^2)}| \qquad (6)$$

where s is the actual scale factor between images.

Here we compare the performance of three detectors; the Difference of Gaussian detector [1] proposed by Lowe[6] and referred to as *DOG*, Hessian-Laplace detector[2] proposed by Mikolajczyk and Schmid[8]) and referred to as *HL* and the Hessian-Laplace implementation proposed in this research, referred to as *SIHL* (Scale-Interpolated Hessian-Laplace). The performance of the detectors is compared using an image set consisting of 10 images where different amounts of scaling (up to a scale factor of 4.4) and rotations are applied to the reference image[3];. For comparing these different detectors we use the repeatability code provided by Mikolajczyk and Schmid on their webpage[4].

There are two parameters that can affect the repeatability score between images; first is the number of regions detected in both images and second is the size of those regions. Detecting a smaller number of points leads to regions which are distinctive and stable. A large number of points on the other hand can at times clutter the scene and lead to ambiguous matches. While evaluating the performance of affine detectors Mikolajczyk and Schmid[10] observed that for some detectors the repeatability increased with an increase in number of regions while for others it decreased. Hence in order to compensate for region density and to improve the accuracy of the results, we detect the same number of points for all detectors in each image.

Figure 1 shows the repeatability in percentage and the number of correspondences obtained for the different approaches for different scale factors. A better repeatability is obtained for SIHL detector as compared to the other detectors. The points detected using the SIHL approach are detected in close proximity which basically leads to a large number of points detected in the common region of two images which eventually results in high repeatability.

5 Feature Descriptors for Matching

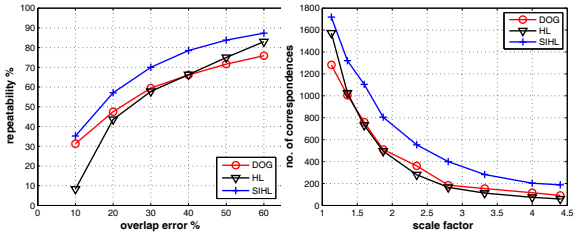
Having extracted features from an image, the next step in any matching or recognition application requires associating every feature with a unique identifier or a signature which can be used to identify the feature from a database. These identifiers or signatures used to describe features are termed as *feature descriptors*.

[1]DOG points are computed using the publicly available SIFT executable from David Lowe's webpage http://www.cs.ubc.ca/ lowe/keypoints/

[2]The executable binaries have been taken from http://www.robots.ox.ac.uk/ vgg/research/affine/detectors.html

[3]The image data sets were taken from http://www.robots.ox.ac.uk/ vgg/research/affine/. The ground truth for all images with respect to the reference image is known.

[4]http://www.robots.ox.ac.uk/ vgg/research/affine/evaluation.html



(a) (b)

Figure 1. Repeatability tests of different feature detectors . (a) Repeatability score for 40% overlap error (b) Number of repeated correspondences for 40% overlap error.

A key attribute of descriptors that is vital in determining their robustness, relates to their ability to handle different geometric and photometric transformations. Ideally, descriptors are designed such that they are invariant to changes in image scale and image rotation. In addition to these properties, the descriptor should also be robust to errors in localization of features and should not be affected by partial occlusions.

The SIFT (Scale Invariant Feature Transform) descriptor proposed by Lowe[6] has been one of the most widely used descriptors. It uses the local gradient information inside the patch to build a representation based on an histogram of orientations. In a survey done to compare the performance of different descriptors by Mikolajczyk and Schmid[9], SIFT was shown to perform better than all other local descriptors.

Principal Component Analysis-SIFT (PCA-SIFT) was proposed by Ke and Sukthankar[3] to overcome the problem of the high dimensionality of the SIFT descriptor. Rather than generating orientation histograms like SIFT, the descriptor is computed by extracting horizontal and vertical gradients from the patch. This is followed by a PCA operation which leads to the reduction in dimensionality of the descriptor.

6 Wavelet Descriptors

Wavelet transform has been frequently used for multiresolution analysis of images in order to perform compression, feature extraction and texture analysis. It is the ability of wavelets to generate a compact representation of an image, that is of particular interest when they are used to represent local image structures.

Amongst all the different wavelet basis, Haar wavelets are the ones that are most commonly used for computing descriptors. This frequent use of Haar wavelets arises from that fact that Haar basis functions are computationally very easy to implement. The basis functions for Haar can be numerically expressed as:

$$\psi_i^j(x) = 2^{j/2}\psi(2^j x - i) \qquad i = 0, ..., 2^j - 1 \quad (7)$$

where

$$\psi(x) = \begin{cases} 1 & \text{for } 0 \leq x < 1/2 \\ -1 & \text{for } 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The above notation represents the Haar basis functions in one dimension. Then given a one dimensional signal, its representation in terms of Haar basis can be written as:

$$f(x) = \sum_{i=-\infty}^{\infty} c_i^{j\prime}\phi_i^{j\prime}(x) + \sum_{i=-\infty}^{\infty}\sum_{j=j\prime}^{\infty} d_i^j\psi_i^j(x) \quad (9)$$

where

$$\phi_i^j(x) = \phi(2^j x - i) \qquad i = 0, ..., 2^j - 1 \quad (10)$$

$$\phi(x) = \begin{cases} 1 & \text{for } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Here $j\prime$ is the starting scale. The functions $\phi_i^j(x)$ are known as the scaling functions. The coefficients c_i^j associated with the scaling functions are used to represent the average values. The coefficients d_i^j of the wavelet functions represent the detail coefficients which can be used along with the average values to reconstruct the original signal.

In the context of image features, the objective of using Haar basis is to represent the image patch around a feature in terms of Haar coefficients. The idea is that some of the detail coefficients can be neglected without losing too much information about the patch. Haar basis functions are orthogonal; this property helps to preserve Euclidean distances between feature descriptors. Thus, the Euclidean distance measure can be applied directly on Haar descriptors to find the nearest neighbor.

Krishnamachari and Mottaleb[4] proposed a method to perform image retrieval and match video segments using a descriptor computed from Haar basis functions. Their method is based on extracting color histogram of an image and converting it into a 63 bit descriptor of Haar transform coefficients. This descriptor is then used as an index to perform image retrieval from a database. Utenpattanant et al.[11] proposed a method using the same descriptor along with a pruning technique for their retrieval application.

Brown et al.[2] developed a method for matching images where a 8x8 patch of sampled intensity values extracted for a point, was converted to a 64 bit descriptor of Haar transform coefficients. The first three non zero Haar coefficients were then used to represent a feature and used as an index to find nearest neighbors using a lookup table method. Our method of computing Haar descriptors is closely related to their method.

6.1 Haar Descriptor Computation

The computation of Haar descriptors is done by first selecting a patch around a point whose size is proportional to its scale value and rotated depending upon its orientation. The patch which is selected at the keypoint's scale image is then resized to a patch of size $patchsize$. We have used patches of $patchsize$ 8, 16, 64. Normalization is performed to ensure the patch has zero mean and unit standard deviation. This ensures that the descriptor is unaffected by changes in image illumination. Then the patch is converted to a descriptor of Haar coefficients using the Haar wavelet transform. We generate the final descriptor by selecting a few or all of the Haar coefficients. The number of coefficients to be selected is decided by the parameter $vectorsize$. Using different $patchsize$ and $vectorsize$ results in different configurations of Haar descriptors.

To measure the similarity between two descriptors, the ratio of Euclidean distances used in SIFT can be used. The measure is calculated by computing the ratio of distance to the closest and the second closest neighbor for a given descriptor. It is assumed that the nearest neighbor is a correct match while the second nearest is an incorrect match. It has been shown that it is easier to differentiate between correct and incorrect matches using this measure based on ratio of distances rather than using the distance of nearest neighbor alone.

7 Image Matching Experimental Results

Various evaluation metrics have been proposed in the literature for analyzing matches. Here, we evaluate our results using two evaluation metrics. For the first metric, we compute the number of correct matches obtained for various matching algorithms in both absolute and relative terms [10].

Every feature point is associated with a scale-proportional region and a descriptor. Two points are said to be matched if the overlap error between their respective regions is below 50% and the Euclidean distance between their descriptors is below a threshold. Once we have the number of correct matches, we compute the matching score, which is defined as the ratio of correct matches to the number of detected points.

In any matching application, not only are we interested in knowing the number of correct matches but also the number

of false matches obtained. The metric which is widely used for performing such analysis is based on measuring *recall* and *1-precision*. The correct and false matches are analyzed by this metric for different values of the matching threshold.

In the context of our research, a true positive corresponds to a correct match between the two images while a false positive refers to a false match. Consequently, we redefine recall and 1- precision as:

$$recall = \frac{number\ of\ correct\ matches}{number\ of\ correspondences} \quad (12)$$

$$1 - precision = \frac{number\ of\ false\ matches}{total\ number\ of\ matches} \quad (13)$$

The ideal recall vs 1-precision curve is a vertical line which starts at zero recall and goes up to a recall value of one for a zero value of 1-precision and is horizontal thereafter. However, in any scenario due to image distortions and non repeated points we will get false matches. Usually, as the threshold parameter is relaxed, the number of correct matches increases which in turn increases recall but as more false matches are introduces, 1-precision increases as well.

8 Matching Experiments

Any matching algorithm consists of two distinct entities, namely the feature detector and the feature descriptor. The feature detector is used to detect points in an image while the feature descriptor is used to generate a description for these points. Here we use the above mentioned metrics to evaluate the performance of different detectors when combined with different descriptors. The different configurations evaluated in this research are given below:

- Difference of Gaussian (DOG) detector + SIFT/PCA-SIFT descriptor[5] [6]
- Hessian-Laplace (HL) detector + SIFT/PCA-SIFT descriptor[7]
- Scale Interpolated Hessian-Laplace (SIHL) detector (our Hessian-Laplace approach) + SIFT/PCA-SIFT descriptor[8]
- Scale Interpolated Hessian-Laplace (SIHL) detector (our Hessian-Laplace approach) + Haar descriptor

[5]For computing SIFT descriptors for DOG detector we use the publicly available SIFT executable from David Lowe's webpage http://www.cs.ubc.ca/ lowe/keypoints/

[6]For computing PCA-SIFT descriptors, we use the binaries provided by Yan Ke http://www.cs.cmu.edu/ yke/pcasift/

[7]SIFT and PCA-SIFT descriptors are computed using the code provided on K.Mikolajczyk and C.Schmid's website http://www.robots.ox.ac.uk/ vgg/research/affine/descriptors.html

[8]We use our implementations of SIFT and PCA-SIFT

From the results of Figure 2, it can be observed that the size of the patch has little effect on the number of correct matches. Also, the 64 bit Haar descriptors produce more matches as they produce a more distinctive representation.

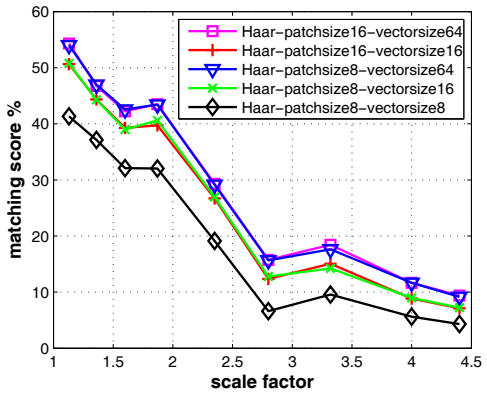


Figure 2. Matching score for different Haar descriptors combined with SIHL detector.

Figure 3 shows the matching scores for the enumerated configurations. The first observation that can be made from these graphes is that for every feature detector, a better score is obtained when the detector is combined with SIFT descriptor than PCA-SIFT. This leads to the conclusion that SIFT is a better descriptor than PCA-SIFT. The SIHL detector gives the maximum number of correct matches and the highest matching score when combined with the SIFT descriptor. A similar pattern can be observed when the three detectors combined with PCA-SIFT descriptor are considered. In this case, the SIHL detector along with PCA-SIFT gives the highest number of correct matches. Surprisingly, the 64 Haar descriptor obtains the second highest matching score and the second largest number of matches throughout the range of scales for the given image scene. This indicates that once a patch has been selected around a point which is invariant to scale change and image rotation, even a simple operation like Haar decomposition can be used to generate a sufficiently reliable descriptor.

Although these curves do indicate how good a matching technique is in finding the number of correct matches, it tells us nothing about the number of false matches obtained. In order to investigate that aspect, we look at the recall vs 1-precision curves of different strategies for both nearest neighbor matching and distance ratio matching measure (refer to Figure 4). A quick look at the these curves indicates that even though the DOG detector with SIFT descriptor doesn't give the highest number of matches, it still gives the most stable matches with the fewest mismatches. This is due to the distinctiveness of DOG points which re-

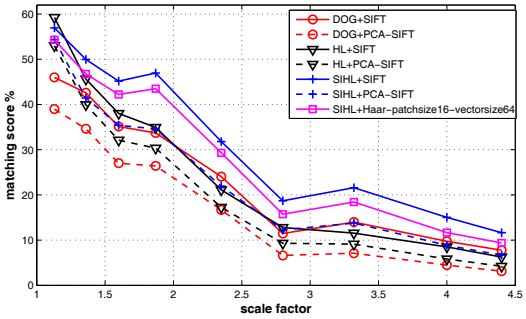


Figure 3. Results for different matching strategies for the scaling and rotation image dataset.

sults in very few ambiguous matches (point in one image being matched to two or more points in the other). The curves obtained for the two Hessian-Laplace detectors with the SIFT descriptor are quite similar. The Haar descriptor and the combination of DOG detector and PCA-SIFT descriptor also give good curves. Another thing that can be observed from the two graphs is that the SIFT descriptor obtains a higher recall for the same 1-precision value for the ratio matching measure while the PCA-SIFT descriptor obtains a higher recall for the same 1-precision for the nearest neighbor measure.

References

[1] M. Brown and D. Lowe. Invariant Features from Interest Point Groups. In *Proceedings of the 13th British Machine Vision Conference*, pages 253–262, 2002.

[2] M. Brown, R. Szeliski, and S. Winder. Multi-Image Matching using Multi-Scale Oriented Patches. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2005.

[3] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 506–513, 2004.

[4] S. Krishnamachari and M. Mottaleb. Compact Color Descriptor for Fast Image and Video Segment Retrieval. In *Proceedings of IST/SPIE Conference on Storage and Retrieval of Media Databases*, 2000.

[5] T. Lindeberg. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[6] D. Lowe. Distinctive Image Features from Scale Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[7] K. Mikolajczyk and C. Schmid. Indexing based on Scale Invariant Interest Points. In *Proceedings of the 8th International Conference on Computer Vision*, pages 525–531, 2001.

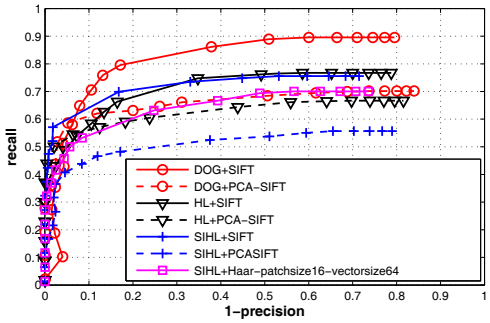
[8] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[9] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

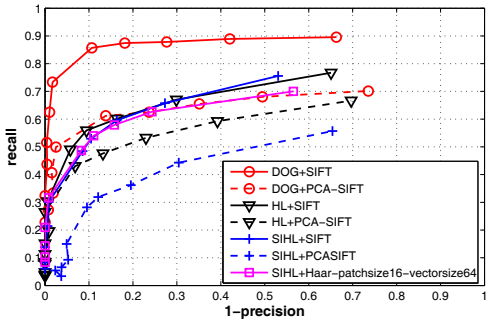
[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[11] A. Utenpattanant, O. Chitsobhuk, and A. Khawne. Color Descriptor for Image Retrieval in Wavelet Domain. In *Proceedings of 8th International Conference on Advanced Communication Technology*, pages 818–821, 2006.

[12] A. Witkin. Scale Space Filtering: A New Approach to Multi Scale Description. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 150–153, 1984.



(a)



(b)

Figure 4. Comparison of different matching strategies for two images from the scaling and rotation image dataset. (a) Nearest neighbor matching measure (b) Distance ratio matching measure