LSOD: Local Sparse Orthogonal Descriptor for Image Matching

Yiru Zhao, Yaoyi Li, Zhiwen Shao, Hongtao Lu*
Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
Department of Computer Science and Engineering, Shanghai Jiao Tong University, P.R.China
{yiru.zhao, dsamuel, shaozhiwen, htlu}@sjtu.edu.cn

ABSTRACT

We propose a novel method for feature description used for image matching in this paper. Our method is inspired by the autoencoder, an artificial neural network designed for learning efficient codings. Sparse and orthogonal constraints are imposed on the autoencoder and make it a highly discriminative descriptor. It is shown that the proposed descriptor is not only invariant to geometric and photometric transformations (such as viewpoint change, intensity change, noise, image blur and JPEG compression), but also highly efficient. We compare it with existing state-of-the-art descriptors on standard benchmark datasets, the experimental results show that our LSOD method yields better performance both in accuracy and efficiency.

Keywords

Image matching; autoencoder; local feature descriptor

1. INTRODUCTION

Local feature descriptor is basal research of many computer vision problems, such as image stitching [11], camera calibration [19], object detection [14], and so on. SIFT keypoint detector and descriptor [12], which was proposed a decade ago, has been proved effective in many image matching scenarios [18, 20], but it imposes a large computational cost, especially when used for real-time applications such as simultaneous localization and mapping (SLAM) systems. Many algorithms were proposed to improve SIFT in the following years, SURF [3] is one of them, which is faster but less accurate than SIFT. DSP-SIFT [5] raises a modification based on pooling gradient orientations. KAZE [1] introduces a feature detection and description algorithm in nonlinear scale spaces. It is accelerated in [2], by a descriptor called AKAZE.

On the other hand, machine learning and neural network are two of the rapidly growing fields in recent years and have achieved great success in many classical computer vision problems, such as image classification [9] and action recognition [8]. Inspired by sparse autoencoder, one of the well-known neural network models, we propose a new image local feature descriptor. With the orthogonal features learned from image dataset, autoencoder encodes an image patch as the descriptor. Our method is called Local Sparse Orthogonal Descriptor(LSOD), an example is shown in Figure 1. The main contributions of this paper include:

- Enhancing FAST detector with median filter scale pyramid and intensity centroid.
- Proposing a method of training a sparse orthogonal autoencoder used to describe the local image feature patch.

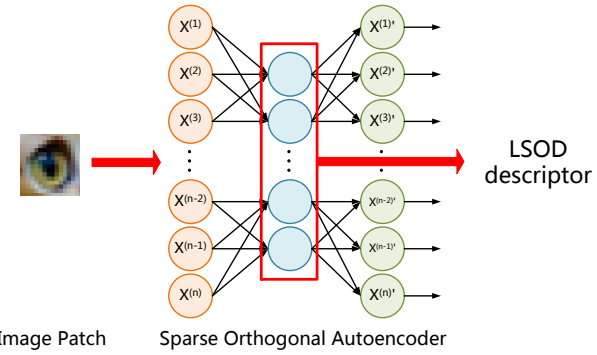

Figure 1: Illustration of calculating the LSOD descriptor for an image patch.

2. RELATED WORK

Detector: The first step in image matching is detecting interest points in the image and there have been many productive interest point detectors. Harris corner detector [7] gives a mathematical approach for determining whether an image patch is flat, edge or corner. SIFT calculates histograms of gray level gradient and chooses the peak orientation as the main direction. SURF uses approximation of block patterns, which is faster than computation of gradients. FAST and its extensions [16, 17] are good choices for keypoints detecting in real-time systems. They are stable and efficient to find corner keypoints, but sensitive to scale variance. Therefore the FAST detector is often applied with pyramid schemes for scale change.

*Corresponding author.

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2967217

Descriptor: With one interest point and nearby patch pixels, we need a discriminative descriptor to calculate the local feature vector. SIFT generates feature descriptors by calculating histograms of oriented gradients (HOG), while SURF's descriptor is based on wavelet responses in horizontal and vertical direction. Some binary descriptors, such as BRIEF[4], use simple binary tests between pixels to represent features, which could use the Hamming distance instead of the Euclidean distance to search the matched pairs and reduce the time complexity.

3. THE PROPOSED METHOD

In our method, we use FAST detector in image pyramid to find interest points at different scale levels and train a sparse orthogonal autoencoder on a large image patch dataset to encode the local feature vector.

3.1 Enhanced FAST detector

FAST detector is famous for its light computational complexity. Its segment test criterion operates by comparing the intensity between the corner candidate p and a circle of sixteen pixels around it. If there exists a set of n contiguous pixels in the circle which are all brighter than $I_p + t$ or darker than $I_p - t$, p will be classified as a corner, where I_p is the intensity of the candidate pixel p and t is a manual threshold. However, FAST detector can not resist rotation and scale changes.

An usual approach for multiscale algorithm is generating an image pyramid by filtering the original image with an appropriate function over increasing scales. Gaussian kernel is frequently-used while with the drawback of loss of localization accuracy, because the natural boundaries of objects are smoothed to the same degree with details and noise. Alcantarilla *et al.* [1] performs feature detection in nonlinear scale pyramid, in which the noise in images is blurred but the object edges remain unaffected, the resulting algorithm is called KAZE. However the improvement brings out high computation complexity. To alleviate this drawback, we use median filter instead of Gaussian filter to build the image pyramid, which could obtain the similar result as KAZE without paying the high cost in computation. See the example in Figure 2. At high levels in pyramid, the details and noise in the original image (such as the texture on the brick wall) should be ignored and the prominent structure (like the edges of building or windows) should be maintained. This is well done in median filter pyramid and KAZE's nonlinear pyramid. However, the boundaries are blurred in Gaussian filter pyramid, resulting in a loss of localization and feature describing.

At each level of the image pyramid, we need to calculate main orientation of each detected point. The concept of intensity centroid was first proposed in [15]. It defines the moments as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{1}$$

where $I(x,y)$ denotes the intensity of pixel at the position (x,y), and the intensity centroid is located in:

$$C \triangleq (C_x, C_y) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{2}$$

Denoting the geometric center of a feature patch as $O \triangleq (O_x, O_y)$, direction of the vector $\overrightarrow{OC}$ is determined as the main orientation of the local interest point patch. We can calculate an angle θ by:

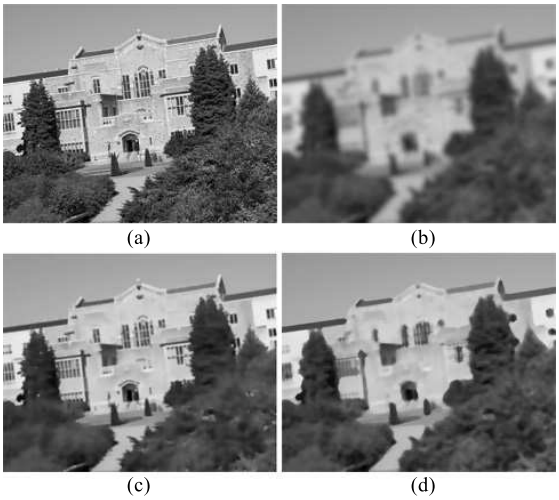

Figure 2: Comparison between the Gaussian filter, median filter and nonlinear diffusion scale pyramid. (a): original image. (b, c, d):Gaussian filter pyramid, KAZE's scale pyramid and median filter pyramid at the same scale level.

$$\theta = \begin{cases} \tan^{-1}\dfrac{O_y - C_y}{O_x - C_x}, & if \quad C_x \le O_x \\ \tan^{-1}\dfrac{O_y - C_y}{O_x - C_x} + \pi, & if \quad C_x > O_x \end{cases} \tag{3}$$

Then the patch is rotated by θ degrees before calculating description, this can help resisting the influence of image rotation.

3.2 Sparse orthogonal autoencoder descriptor

The local information used to generate feature description varies from method to method. SIFT [12] counts the histogram of gradient orientation. While it is time consuming to calculate the gray level gradient. BRIEF and some other binary descriptors [4, 10] use simple binary tests between pixels to represent features, which is fast to generate but weak in expression ability.

In order to overcome these problems, here we propose a Local Sparse Orthogonal Descriptor (LSOD) for feature description. With a trained autoencoder we can encode the descriptor of a local image patch efficiently. Learning local features needs large amounts of image patches as the training data. Sampling patches in image database randomly is a common approach. But in our case, we only care about how the autoencoder will represent the interest point patches, such as the corners. So we construct a training set with 500k colored patches sampled from the PASCAL 2006 database [6], using the enhanced FAST detector introduced in section 3.1. We train an autoencoder on the image patch data, using the neural network shown in the Figure 3 (a).

Having one image patch $x = \left[x^{(1)}, x^{(2)}, \ldots, x^{(n)}\right]^T$ as the input layer 1, with the weight matrix $W^{(1)} \in R^{h \times n}$ and the activation function $f(x)$, often sigmoid function $f(x) = \dfrac{1}{1+e^{-x}}$ is used, we can get the output of the hidden layer

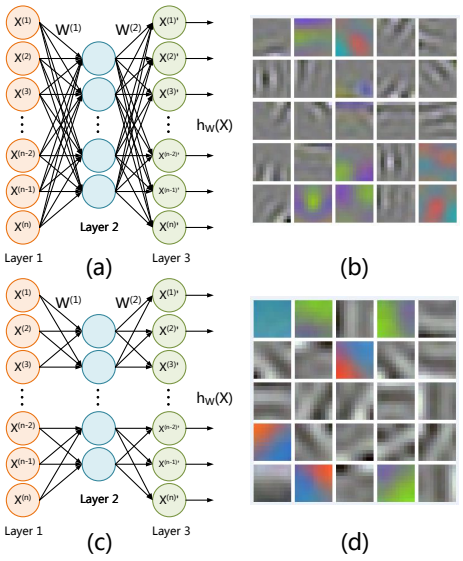

Figure 3: The neural network model and the visualization of features learned by the network. First Row: fully-connected autoencoder. Second Row: partially-connected autoencoder. Best viewed in color.

2 as: $a^{(2)} = f(W^{(1)}x)$, h is the number of hidden units. Similarly, we can calculate the output of layer 3 as: $a^{(3)} = f(W^{(2)}a^{(2)})$, $a^{(3)}$ is also denoted as $h_W(x)$.

A sparse autoencoder tries to learn a function $h_W(x) \approx x$ with some sparsity constraint. Given the training data set $X = [x_1, x_2, \ldots, x_N]$ of the interest point patches, we can calculate the average activation of the jth hidden unit as:

$$\hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N}\left[a_j^{(2)}(x_i)\right] \tag{4}$$

then enforce the constraint $\hat{\rho}_j = \rho$ to guarantee the sparsity of autoencoder, where ρ is a sparsity parameter, which is usually a small number close to zero. A relaxed penalty term is used here to replace the constraint $\hat{\rho}_j = \rho$:

$$\sum_j \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{5}$$

This term can also be written as the KL divergence:

$$\sum_j KL(\rho\|\hat{\rho}_j) \tag{6}$$

By means of gradient descent method, we can minimize the cost function and get the optimized encoding parameter $W^{(1)}$, which is visualized in the Figure 3 (b). It shows that the different hidden units have learned to detect edges in the image patch. However, in this fully-connected autoencoder, each hidden unit is only sensitive to a partial region of local image patch, and more than one units may detect the edges with same texture, color or orientation, just differing in position. This means that there exits high parameter redundancy in the fully-connected autoencoder. To cope with this problem, we propose a partially-connected autoencoder, as shown in Figure 3 (c). We split the whole network into m weight-sharing sub-networks, with the advantage that we only need to train the sub-autoencoder one time instead of training m different sub-autoencoders. Although the training data is m times bigger than the original data, we reduce the number of parameters by m^2 times. And the training time is reduced by about m times than the original autoencoder in our experiments. As shown in Figure 3 (d), the hidden units of partially-connected autoencoder detect more distinctive edge features, which can generate more discriminatory descriptions.

However, the partially-connected autoencoder still could not eliminate the similarity between the learned features totally. So we enforce another constraint that the weight matrix W must be an orthogonal matrix. In order to make the optimization tractable, we use a Frobenius norm as a relaxed penalty term added to the cost function. Finally the overall cost function of our problem is:

$$\begin{aligned} J(W) =& \frac{1}{N}\sum_{i=1}^{N}\|h_W(x_i) - x_i\|^2 + \beta\sum_j KL(\rho\|\hat{\rho}_j) \\ &+ \gamma\|WW^T - I\|_F \end{aligned} \tag{7}$$

After training the sparse orthogonal autoencoder, a weight matrix $W^{(1)}$ is obtained and we can encode a local patch x with the descriptor $W_s^{(1)}R_\theta x$, where the rotation matrix R_θ makes the patch rotate to the main direction, and $W_s^{(1)}$ denotes the parameters $W^{(1)}$ resized to the corresponding scale level.

Thus, our LSOD descriptor becomes:

$$descriptor = \frac{W_s^{(1)}R_\theta x}{\|W_s^{(1)}R_\theta x\|} \tag{8}$$

The normalization ensures that patches in different scales return same descriptors, it also helps resisting the illumination variance.

4. EXPERIMENTS

In this section we present the evaluation results of our descriptor on standard datasets and its comparisons to state-of-the-art descriptors. For SIFT, SURF we use the implementations included in OpenCV. For KAZE, AKAZE and DSP-SIFT, the original codes provided by the authors were used. We set $\rho = 0.01, \beta = 5, \gamma = 1$ to train the autoencoder. All experiments results were obtained on a Core i5 Quad 3.0GHz laptop computer.

4.1 Interest point matching

In this section, we evaluate our descriptor on the standard Oxford dataset, which contains images with different geometric and photometric transformations, such as : viewpoint change, scale change, image rotation, image blur, illumination change and JPEG compression.

We use the evaluation criterion proposed by Mikolajczyk and Schmid [13] which is based on the number of correct and false matches between image pair. We use the nearest neighbor distance ratio (NNDR) as the matching strategy, which accepts a match if the distance ratio between the first and second nearest neighbors is below a threshold. The results are presented with recall versus 1-precision curves,

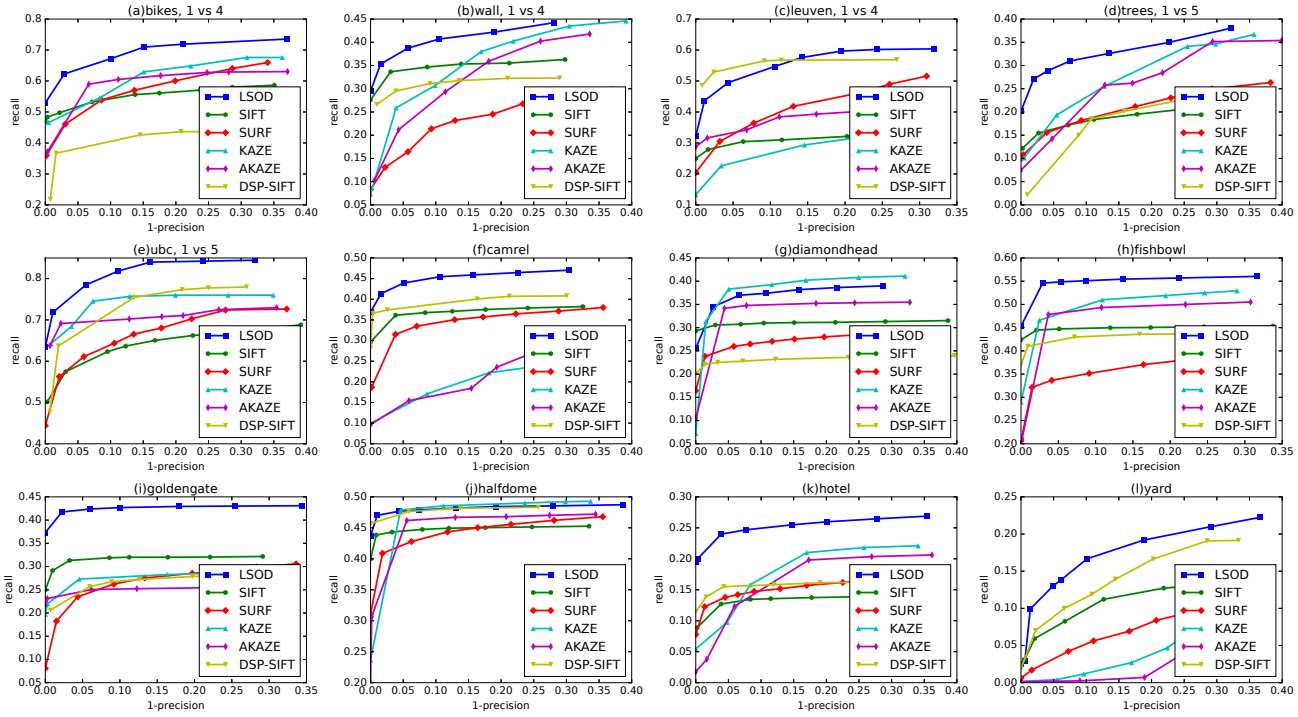

Figure 4: Interest points matching experiment results.(a)-(e): On Oxford dataset. (f)-(l): On Adobe Panorama dataset. Best viewed in color.

which is computed based on different matching thresholds:

$$recall = \frac{\#correct\,matches}{\#correspondences} \tag{9}$$

$$1 - precision = \frac{\#false\,matches}{\#allmatches} \tag{10}$$

The experiment results are plotted in Figure 4(a)-(e). Obviously, our LSOD gives the best results overall. Other methods, each with its strengths, rank 2nd to 6th on different image transformations. As can be observed, the proposed LSOD obtains a high discriminative power while staying robust to many image transformation.

4.2 Test on panorama dataset

One mainly application of image matching is panorama image stitching. Adobe Panoramas Dataset consists of several panorama image sets in different scenes, together with ground truth homographies for each overlapping image pair. We still use the same evaluation criterion as section 4.1. The results plotted in Figure 4(f)-(l) shows that our LSOD outperforms other descriptors on this panorama image dataset. It is appropriate for image stitching application and capable in variant scenes.

4.3 Timing Evaluation

In this section we perform a timing evaluation for a comparison between these feature descriptors. Considering that different methods detect different numbers of interest points, we record the total time of detecting and computing features divided by the number of detected points. The test uses the the first image of each scene in standard Oxford dataset.

Table 1 shows the average timing results in milliseconds for detecting and computing each local feature patch. As can be observed, our LSOD feature is more computational efficient than other descriptors. It is about 4 times faster than SIFT and 1.5 times faster than SURF. KAZE is computational expensive due to the calculation of nonlinear scale pyramid. AKAZE is about 3 times faster than KAZE, but it is only comparable to SIFT. DSP-SIFT is the slowest may because its implementation is based on unoptimized MATLAB code.

Table 1: Timing evaluation result.

Method	LSOD	SIFT	SURF	KAZE	AKAZE	DSP-SIFT
time(ms)	**0.038**	0.132	0.054	0.411	0.131	1.324

5. CONCLUSIONS

In this paper we proposed a Local Sparse Orthogonal Descriptor (LSOD) for image matching. Compared with the previously proposed methods, experimental results show that LSOD is superior both in discriminative expression and efficient computational process.

6. ACKNOWLEDGMENTS

This paper is supported by NSFC (No. 61272247, 61533012, 61472075), the 863 National High Technology Research and Development Program of China (SS2015AA020501), the Basic Research Project of "Innovation Action Plan" (16JC1402800) and the Major Basic Research Program (15JC1400103) of Shanghai Science and Technology Committee.

7. REFERENCES

[1] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In *Computer Vision–ECCV 2012*, pages 214–227. Springer, 2012.

[2] P. F. Alcantarilla and T. Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010.

[5] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5097–5106, 2015.

[6] M. Everingham, A. Zisserman, C. Williams, and L. Van Gool. The pascal visual object classes challenge 2006 (voc 2006) results. 2006.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[8] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011.

[9] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. *arXiv preprint arXiv:1409.5185*, 2014.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[11] Y. Li, Y. Wang, W. Huang, and Z. Zhang. Automatic image stitching using sift, 2008.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[13] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[14] P. Piccinini, A. Prati, and R. Cucchiara. Real-time object detection and localization with sift-based clustering. *Image and Vision Computing*, 30(8):573–587, 2012.

[15] P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999.

[16] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[17] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2010.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, volume 1, page 2, 2008.

[19] B. Telle, M.-J. Aldon, and N. Ramdani. Camera calibration and 3d reconstruction using interval analysis. In *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on*, pages 374–379. IEEE, 2003.

[20] X. Wang and W. Fu. Optimized sift image matching algorithm. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 843–847. IEEE, 2008.