Comparative Study Of Local Descriptors For Measuring Object Taxonomy

Baptiste Hemery, Hélène Laurent, Bruno Emile, Christophe Rosenberger

Comparative Study Of Local Descriptors For Measuring Object Taxonomy

B. Hemery[1] H. Laurent[2] B. Emile[2] C. Rosenberger[1]

[1] Laboratoire Greyc
ENSICAEN - Université de Caen - CNRS
6 boulevard du Maréchal Juin
14000 Caen - France

[2] Institut Prisme
ENSI de Bourges - Université d'Orléans
88 boulevard Lahitolle
18000 Bourges - France

Abstract

Many object descriptors have been proposed in the state of the art. For many reasons (occlusion, point of view, acquisition conditions. . .), local descriptors have a better robustness for image understanding applications. The goal of this paper is to make a comparative study of eight recent local descriptors. The objective is here to quantify their ability to generate automatically an object taxonomy. In order to answer this question, we use the Caltech256 benchmark which provides a large object taxonomy used as reference. This study shows that SIFT, differential invariants and shape context descriptors are the best ones to achieve this goal.

1. Introduction

Image processing includes many steps from image acquisition (with camera, webcam, satellite. . .) to image interpretation. This last step consists in automatically extracting information about objects present in an image (detection of objects of interest, quantitative measure. . .). Whatever the foreseen application may be (biometric systems, medical imaging, video monitoring. . .), the extracted information conditions the performances of the resulting process. It is required for this localization to be as precise as possible and with a correct recognition.

The evaluation of object classification or recognition is a crucial step for the validation of an algorithm. The classical way to conduct this evaluation is the use of ROC curves [1, 18] or Precision/Recall curves [5, 15]. Such approaches use the false recognition rate, that is to say the rate where the object is not correctly recognized. By this way, we just count how many times the algorithm did not find the correct object, independently of the coherence of the made classification.

Our objective is to distinguish among recent local descriptors from the state of the art, those that permit to distinguish objects from different categories. To achieve this goal, we have to propose a similarity measure between objects and categories of objects. Such a similarity measure could be used in an evaluation process in order to improve its quality. This similarity is computed from the well known confusion matrix. As an example, the object "Dog" should be more similar to the object "Cat" than the object "Apple". By the way, we compare local descriptors that enable the definition of such a similarity measure by considering their ability to group objects that are similar within a category.

In this paper, we first present the local descriptors tested in the comparative study. We then present an experimental study where local descriptors performances are compared using the taxonomy available in the Caltech256 database [7]. Finally, some conclusions and perspectives are given.

2. Background

In this work, we use local descriptors that are commonly used to describe pattern and objects. What is often computed to evaluate a local descriptor is its capacity to discriminate an object from others. In this paper, we are interested in their capacity first to have a similar description of objects that belong to a given category (animal, vehicles,. . .) and second, to distinguish objects that do not belong to the same category.

To compute the similarity measure, we first compute local descriptors for images of each object. The computation of local descriptors first needs a keypoints detector. Among all keypoints detectors that have been proposed in the literature [14, 3], the Hessian-Affine detector described in [13] reveals itself as interesting. The advantage of this detector is that it finds much more keypoints than the Harris-Laplace [8] or Harris-Affine ones [13], which is necessary for some samples images. Examples of detected keypoints can be found in figure 1. On the neighborhood of each keypoint, we then compute an invariant descriptor. The descriptors we tested in this work are presented below.

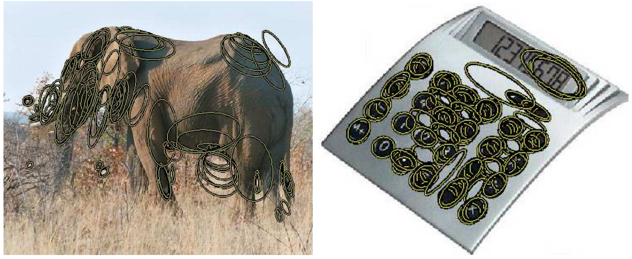


Figure 1. Examples of detected keypoints

- Complex filters (cf) [17],

  Complex filters are derived from the following equation:

  $$K_{mn}(x, y) = (x + iy)^m (x - iy)^n G(x, y) \qquad (1)$$

  where $G(x, y)$ is a Gaussian function, x and y the pixel position in the filter. Fifteen filters are used, such as $m + n < 6$, to compute the descriptor on a 41*41 pixels patch. Some of the obtained filters are presented in figure 2. The resulting descriptor is a 15-elements vector.

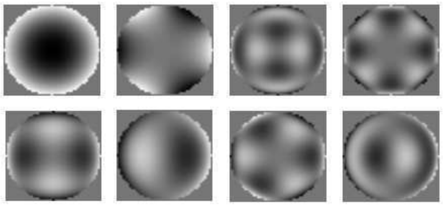


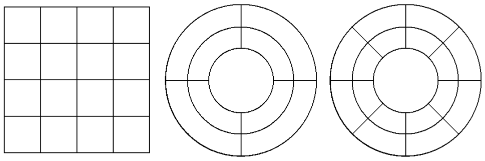
Figure 2. Complex filters

- Steerable filters (jla) [6] and differential invariants (koen) [10],

  Descriptors from steerable filters and from differential invariants use derivatives computed by convolution with Gaussian derivatives for a 41*41 pixels patch. Steerable filters are designed as the output of a three basis filter bank. The outputs are then multiplied by a set of gain maps. Each basis filter is assigned to an angle θ varying from 0° to 60° and 120°. The derivatives of steerable filters are computed up to the 4^{th} order, the final descriptor contains a 14-elements vector. The differential invariants are computed up to the 4^{th} and the final descriptor is a 12-elements vector.

- SIFT (sift) [11, 12],

  SIFT is a well known descriptor and has been largely used and studied since Lowe created it in 1999. The SIFT descriptor, as described in [12], consists of four stages: (i) scale-space peak selection, (ii) interest point localization, (iii) orientation assignment, and (iv) descriptor. The three first stages correspond to the localization of keypoints. By the way, we use only the fourth stage because we use the Hessian-Affine as keypoints detector. The descriptor is computed from a 4*4 location Cartesian grid as shown in figure 3(a). The gradient on each location bin is then quantized into 8 orientation bins, and is computed on the patch around the keypoint. This leads to a 128-elements vector.



(a) SIFT (b) Shape context (c) GLOH

Figure 3. Location for different descriptors

- PCA-SIFT (pca) [9],

  The PCA-SIFT descriptor is a variation of the SIFT one. The principal advantage of this descriptor is that the size of the representation is reduced, as it generates a 36-elements vectors. The main idea of this descriptor is to use a Principal Components Analysis (PCA) on the same patch as the one proposed by Lowe in [12], instead of computing a gradient histogram.

- Shape context (sc) [2],

  Shape context works globally like the SIFT descriptor. The difference lies in the location grid which is log-polar and it has 9 bins: 2 locations from each of the 4 directions plus the center as we can see in figure 3(b). Moreover, each location bin is quantized along 4 orientation bins. The final descriptor is a 36-elements vector and is computed on edges extracted from the Canny edge detector [4].

- Gradient location and orientation histogram (gloh) [14],

  The gradient location and orientation histogram is also a descriptor derived from the SIFT descriptor. It has been designed to increase the robustness and distinctiveness of the SIFT descriptor. It works, like the shape context, on a log-polar location grid. The location grid presented in figure 3(c) has 17 bins: 2 from each of 8 directions plus the center. Each location bins is quantized along 16 directions. This leads to a 272-elements vector, but the final descriptor is reduced to a 128-elements vector with a PCA reduction.

- Cross-correlation (cc) [14].

  Finally, the cross correlation descriptor corresponds to a smoothed and uniformly sampled image of the patch around the interest point. The patch is sampled at 9*9 pixels so the final descriptor is a 81-elements vector.

All these descriptors have been studied in [14], SIFT and SIFT-based descriptors obtain the best results in the case of object recognition. As we are in a different context, we aim to see if they are also able to measure the similarity of objects category.

3. Developed method

To compute the similarity measure between two objects or two categories, we are looking for similar parts in two sample images. We assume the fact that the more there are corresponding keypoints between the two images, the more they are similar. An example of matching result between keypoints can be found in figure 4. We can see the matching of two samples from objects “calculator” and “computer keyboard”, both in the “electronic” category. Even if these samples are from different objects, we can see some similarities as both objects have buttons.

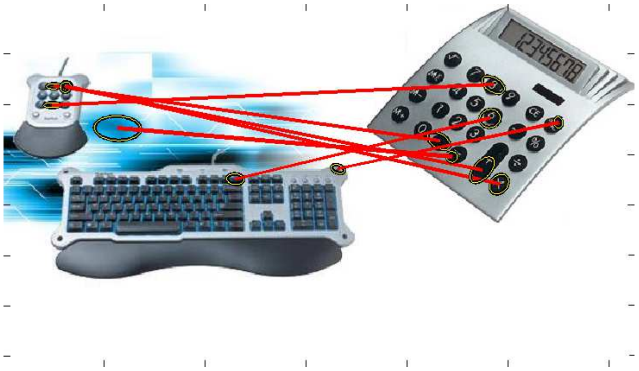


Figure 4. Example of matching between keypoints of two different samples containing different objects within the same Caltech category

The computation of a similarity measure between objects is made of 2 steps. First, we apply a matching procedure to define similar parts of different samples. Second, we compute a score quantifying if samples belong to the same object or category.

3.1. Matching procedure

We are looking for associations between keypoints from each sample image as in [16]. An association is defined as a double matching between two keypoints: for a keypoint x of image k, we are looking for the nearest neighbor keypoint y among the set $Y(I_l)$ of all points of image l. We also check that the second nearest neighbor point y' is far enough from the nearest neighbor (thanks to parameter C), otherwise, the keypoint is rejected. That is to say:

$$d(x, y) = \min_{z \in Y(I_l)} d(x, z) \tag{2}$$

and

$$d(x, y) \leq C * d(x, y') \tag{3}$$

where C is an arbitrary threshold set to $0,7$ and the computed distance $d(\cdot, \cdot)$ between keypoints descriptors is the Euclidean one between the normalized vectors. If these conditions are fulfilled, then we say that the keypoint y is matched to keypoint x. We then associate the keypoint x from image k with keypoint y from image l if y is matched to x and x is matched to y. We can see on figure 4 lines between the associated keypoints.

Once we have defined the associated keypoints between images k and l, we define the similarity measure between these images $s(k, l)$ as the number of associations. The more there are associated points, the more objects present in the images are similar.

3.2. Similarity measures

The similarity between objects is computed from several sample images from each object. To obtain it, we compute the average similarity measure S_o from the similarity measures s between several images of these two objects:

$$S_o(i, j) = \frac{\sum_{k \in i} \sum_{l \in j} s(k, l)}{Card(i) * Card(j)} \tag{4}$$

where $Card(i)$ denotes the number of images from object i. We can notice that a higher value denotes more similar objects. In a same way, we define a similarity measure for categories S_c as the average similarity measure from objects of the two categories:

$$S_c(u, v) = \frac{\sum_{i \in u} \sum_{j \in v} S_o(i, j)}{Card(u) * Card(v)} \tag{5}$$

where $Card(u)$ denotes the number of objects from category u.

For the validation process, we then gather Intra and Inter informations for objects and categories. For objects, Intra$_o$ corresponds to the set of similarities $S_o(i, i)$ and Inter$_o$ corresponds to the set of similarities $S_o(i, j)$ with $i \neq j$. For categories, Intra$_c$ corresponds to the set of similarities between objects of the same category, that is to say $S_c(u, u)$.

Inter$_c$ corresponds to the set of $S_c(u, v)$, with $u \neq v$. Finally, we define a Score as:

$$\text{Score} = \frac{\text{E[Intra]-E[Inter]}}{\sqrt{\frac{\sigma\text{[Intra]}+\sigma\text{[Inter]}}{2}}} \qquad (6)$$

where E[·] stands for the average and σ[·] stands for the standard deviation. This ratio enables us to determine which descriptors better maximize at the same time the Intra set and minimize the Inter one. The higher the ratio is, the best the descriptor separates objects or categories.

We can then define a similarity matrix such as $MS_{i,j} = S_o(i, j)$. This matrix looks like a confusion matrix, but there are two principal differences. First, we can notice that the case $S_o(i, i)$ is not interesting, as we aim to compute a similarity between different objects. Second, this matrix is symmetric because $S_o(i, j) = S_o(j, i)$. Once built, such a matrix could be used to improve the evaluation of a recognition algorithm, by taking into account the importance of the made error.

4. Experimental results

This section first presents the Caltech256 Database [7]. Second, we present the obtained results.

4.1. Image database

For this comparative study, we use the Caltech256 Database[1] [7]. This database presents several advantages. First, it contains a large number of objects: 256 different objects plus one that represents a clutter class. Objects are various and make this database very difficult. Moreover, sample images from each object are also various. The second advantage is that objects are ordered and grouped within a taxonomy. A part of the taxonomy can be seen in the figure 5. This taxonomy creates categories of objects. Each category contains objects with a common subject.

In this paper, we use one part of this database. We choose 5 categories in the taxonomy: “Household and Everyday”, “Sports”, “Electronics”, “Food” and “Animal”. For each category, we select 10 objects recapitulated in table 1. We obtain a total of 50 objects among the 256 available. For each object, we used the first 20 images of the database, and compute a maximum of 150 keypoints per image. This leads to *1.000* sample images and *509.000* comparisons for this work.

4.2. Results

In order to validate our similarity measure, we check the Intra$_o$ and Inter$_o$ similarities for all objects. Table 2 presents E(Intra$_o$), E(Inter$_o$) and Score$_o$. We can see that the sift descriptor performs the best, closely followed by the gloh, pca and cc descriptors. The performance of sift can be explained by its capacity to distinguish different objects as it obtains a very low value of the Inter similarities. These results are similar to results obtained in [14], which validates our similarity measure and the Score$_o$ indicator.

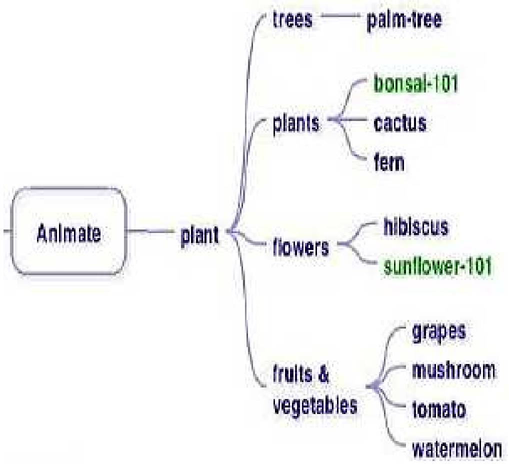


Figure 5. Extract of the taxonomy from Caltech256 database [7]: leaves of the tree are the different objects and nodes are the different categories

Table 2. Results for objects similarities for each descriptor

Descriptors	E[Intra$_o$]	E[Inter$_o$]	Score$_o$
cf	7.35	0.51	12.1
jla	7.42	0.57	11.8
koen	**8.23**	1.37	9.8
sc	6.95	0.12	12.7
sift	6.83	**0.03**	**13.2**
pca	6.87	0.07	13
gloh	6.84	0.04	13.1
cc	6.91	0.10	12.9

The question we would like to answer is how these descriptors are able to distinguish categories of objects that are different. Results, presented in Table 3, show Score$_c$ for each category. First, we can see that most descriptors mistake for the “Household and Everyday” categories, which reflects the difficulty of the Caltech256 database. The category “Animal” is the best recognized, which sounds reasonable given the taxonomy from Caltech: this category is in the group “Animate” whereas the four other category are from the group “Inanimate”. Secondly, we can notice that koen descriptors perform well for all categories, “House” category included. We can also notice that sift, pca, gloh and cc descriptors obtain approximately the same kind of results.

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

Table 1. Selected objects per category

Categories	Household and Everyday	Sports	Electronics	Food	Animal
Objects	American Flag	Baseball bat	Boom box	Beer mug	Bat
	Bathtub	Baseball glove	Breadmaker	Cake	Bear
	Birdbath	Basketball hoop	Calculator	Cereal box	Camel
	Chandelier	Billiard	CD	Chopsticks	Chimp
	Coin	Bowling ball	Computer keyboard	Coffee mug	Conch
	Desk globe	Bowling pin	Computer monitor	Drinking straw	Cormorant
	Doorknob	Boxing glove	Computer mouse	Ewer	Crab
	Fire extinguisher	Dumb bell	Flashlight	Fried egg	Dog
	Hammock	Football helmet	Head phones	Hamburger	Duck
	Hot tub	Golf-ball	Lightbulb	Ice cream cone	Elephant

Table 3. Similarity results for each category: a negative ratio implies that the descriptor is not able to distinguish the category from others

Descriptors	Household	Sports	Electronics	Food	Animal
cf	0.071	0.64	0.64	0.64	0.74
jla	0.051	0.66	0.66	0.65	0.73
koen	**0.20**	0.62	0.54	0.62	**0.83**
sc	-0.058	**0.68**	**0.71**	0.64	0.67
sift	-0.027	0.67	0.68	**0.65**	0.67
pca	-0.058	0.67	0.69	**0.65**	0.67
gloh	-0.044	0.67	0.69	**0.65**	0.67
cc	-0.088	**0.68**	0.68	**0.65**	0.67

Table 4 presents E[Intra$_c$], E[Inter$_c$] and the Score$_c$ for all categories, including the “Household and everyday” category. We can see that only two descriptors bring out: the sift descriptors still have the lower value for Inter similarities whereas the koen descriptors still have the higher value for the Intra similarities. Moreover, the koen descriptor seems to be the best for a difficult database.

Table 4. Results for categories similarities for each descriptor, in- cluding “Household and everyday” category

Descriptors	E[Intra$_c$]	E[Inter$_c$]	Score$_c$
cf	1.06	0.53	0.55
jla	1.13	0.58	0.55
koen	**1.94**	1.44	**0.56**
sc	0.67	0.11	0.53
sift	0.58	**0.02**	0.53
pca	0.61	0.05	0.53
gloh	0.58	0.03	0.53
cc	0.65	0.10	0.52

Table 5 presents the same result as table 4, but without the category “Household and Everyday”. In this case, the database is easier to recognize and this is confirmed by the fact that Intra values increase whereas Inter values decrease. We can see that three descriptors bring out: the koen descriptors have the best Intra values, the sift descriptors have the best Inter values and the sc is a good compromise. We can also remark that most of SIFT-based descriptors perform approximately like the sift descriptor.

Table 5. Results for categories similarities for each descriptor, ex- cluding “Household and everyday” category

Descriptors	E[Intra$_c$]	E[Inter$_c$]	Score$_c$
cf	1.19	0.50	0.67
jla	1.27	0.57	0.68
koen	**2.05**	1.35	0.65
sc	0.82	0.12	**0.68**
sift	0.71	**0.03**	0.67
pca	0.75	0.07	0.67
gloh	0.72	0.04	0.67
cc	0.79	0.11	0.67

Finally, the average computation time of the distance between images and objects is presented in Table 6. We can see that, except the gloh descriptor which is slower, most descriptors have approximately the same computation time between: cf, jla and koen descriptors are around 6 seconds per object, sc, sift, pca and cc descriptors are around 8-9 seconds per object, and te gloh descriptor is at 19 second per object. Moreover, the koen descriptor is the fastest one.

Table 6. Average computation time of similarity between two samples and two objects

Descriptor	Computation time for two samples (ms)	Computation time for two objects (s)
cf	17,0	6,81
jla	16,4	6,54
koen	**14,9**	**5,97**
sc	20,7	8,26
sift	24,0	9,61
pca	20,9	8,36
gloh	49,5	19,81
cc	19,3	7,70

5. Conclusions and perspectives

In this work, we compared 8 recent local descriptors from the state of the art in a particular case. Indeed, we aim to compare their ability to discriminate some object categories from each other. We can see that three descriptors bring out: sift descriptors are the best ones to compute Inter similarities as they are able to reject object from other category whereas koen descriptors are the best ones to compute Intra similarities, and the sc descriptor is a good compromise. In the case of various objects within a category, we suggest to use the koen descriptor, whereas in an easier database, we suggest to use the sift descriptor.

Finally, we aim to use a similarity matrix MS using the sift descriptor that would permit to better evaluate recognition algorithms. For example, if the objet "Dog" is recognize instead of the objet "Cat", we could penalize this error by using the similarity $MS_{\text{“Dog”,“cat”}}$.

Acknowledgment

The authors would like to thank to the French Research Ministry for the financial support.

References

[1] A. Adler and M. Schuckers. Comparing human and automatic face recognition performance. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(5):1248–1255, 2007. 1

[2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000. 2

[3] R. M. C. Schmid and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. 1

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 2

[5] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M.Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G.Dorko, et al. The 2005 pascal visual object classes challenge, 2005. 1

[6] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991. 2

[7] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, http://authors.library.caltech.edu/7694, 2007. 1, 4

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50, 1988. 2

[9] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE Computer Society Conference On Computer Vision And Pattern Recognition*, volume 2. IEEE Computer Society; 1999, 2004. 2

[10] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987. 2

[11] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Kerkyra, Greece, 1999. 2

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. 1, 2

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 27:1615–1630, 2005. 1, 2, 3, 4

[15] H. Muller, W. Muller, D. Squire, S. Marchand-Maillet, and T.Pun. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters*, 22(5):593–601, 2001. 1

[16] C. Rosenberger and L. Brun. Similarity-based matching for face authentication. In *International Conference on Pattern Recognition (ICPR)*, 2008. 3

[17] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or" how do i organize my holiday snaps?". *Lecture Notes In Computer Science*, pages 414–431, 2002. 2

[18] N. Thacker, A. Clark, J. Barron, J. Ross Beveridge, P. Courtney, W. Crum, V. Ramesh, and C. Clark. Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding*, 109(3):305–334, March 2008. 1