SO-BRIEF: FAST RECOGNITION OF RECTANGULAR OBJECTS

Philippe Métais[⋆] *Jian-Jiun Ding*[†]

[⋆] ENSEEIHT, Toulouse, France. philippe.metais@etu.enseeiht.fr
[†] Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan.
jjding@ntu.edu.tw

ABSTRACT

Much research has been conducted in computer vision about feature extraction. In recent years, binary descriptors have been proved to be extremely fast and yet highly discriminative. So-BRIEF aims at bringing the benefits of this kind of local descriptors to the recognition of rectangular objects, such as books, CD covers, boxes, paints, boards and cell phones. It takes advantage of the special geometry of these objects to recognize very efficiently an image, even rotated or distorted. Our main contribution is this new descriptor along with the search for the optimal values of parameters leading to an extremely fast image matching process. We compare it to 2D-DCT based description techniques. This paper also encompasses a discussion about a new method for efficient detection of rectangular structures.

Index Terms— Feature points, real-time matching, descriptor, computer vision, object recognition.

1. INTRODUCTION

Most of today's recognition processes begin with a feature extraction step which completely ignores the shape of the considered objects. This information is only considered during the clustering step, when methods like RANSAC or Least Median of Squares aim at minimizing the distance between experimental and theoretical feature coordinates.

Our research study firstly starts from the basic observation that some image recognition applications only deal with real-life non-deformable rectangular objects, such as books, CD covers [1], paintings [2][3], buildings or furniture [4][5]. A descriptor could thus speed up the recognition process for such objects by taking advantage of their shape similarity.

The second starting point of this research lies in new processors bitcounting instructions (see Intel POPCNT [6] and ARM VCNT [7] instructions) achieving super-fast Hamming distance calculation, which makes them particularly suitable for binary strings comparison.

So-BRIEF is a binary descriptor for rectangular objects designed to be as fast as possible to both build and compare. When efficient recognition tools often rely on innovative data structures [8][9], so-BRIEF tries to improve the efficiency of description itself. Bringing the speed of feature binary descriptors to the field of object recognition while taking advantage of the special geometry of rectangular objects, it may enable new real-time applications for both computers and portable devices with lower computing power.

2. OVERVIEW OF THE METHOD

A primary study case could be as simple as a smartphone user photographing a book cover to get more information about it. If the picture is not taken from a particular point of view, a projective transformation is first required. Our method makes the underlying assumption that it is possible to quickly extract the cover from the background of the image. As discussed at the end of this paper, thanks to the extreme speed of the so-BRIEF descriptor, this assumption may not be as unrealistic as it seems, and our method may even help in achieving such a segmentation.

2.1. 2D Projective Transformation

More precisely, let us imagine that we know the location of the four corners of the cover in the source image. We would first like to correct perspective effects from the photograph in order to make opposite edges parallel. Without more data, we cannot recover the height to width ratio of the cover, nor can we infer which of the four locations corresponds to the upper left corner of the real object. Thus we may only transform the original segmented cover into a normalized square, without presuming its orientation yet (see Fig.1).

As we are working on digital images, some interpolation is needed to compute the intensity of each pixel on the square. To guarantee good execution time, the interpolation is performed linearly using values at neighboring grid points in each dimension, and one will try to reduce the square dimensions as far as possible.

2.2. Building a Descriptor

If our database is made of rectangular objects, these can easily be interpolated along one dimension and turned into squares. A very fast way to build a descriptor for each of these squared

(a) Original image. (b) Segmented cover. (c) Squared cover.

Fig. 1. Transformation of the detected real-life rectangular object into a square. To correct perspective we make opposite edges parallel, but as the rectangle height to width ratio is unknown, we transform the quadrilateral into a square. We don't know its orientation neither. In this example, the book cover is thus turned to the left.

images is then to compare pairs of points, as maid with BRIEF [10]. While the original BRIEF describes an image patch, a so-BRIEF descriptor will code for an entire image. In short, we define n_d descriptor bits as the results of the following τ test on a set of n_d $(\mathbf{x}, \mathbf{y})$-location pairs and an image $\mathbf{I}$:

$$\tau(\mathbf{I}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{I}(\mathbf{x}) < \mathbf{I}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

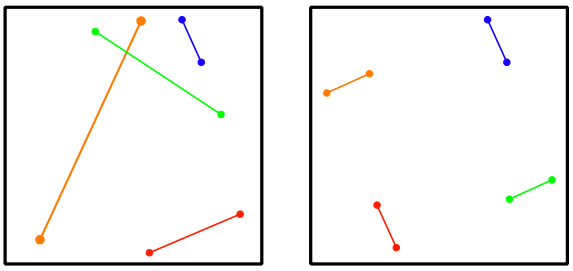
with $\mathbf{I}(\mathbf{x})$ the pixel intensity at $\mathbf{x} = (u, v)^{\mathrm{T}}$ in the smoothed version of $\mathbf{I}$. The descriptor is then defined as a vector of n_d binary tests:

$$f_{n_d}(\mathbf{I}) := \sum_{1 \le i \le n} 2^{i-1} \tau(\mathbf{I}; \mathbf{x}_i, \mathbf{y}_i) \quad (2)$$

Given an images size, a smoothing kernel and a spatial arrangement of the binary tests, each image of the database can thus easily be described by a n_d-dimensional bitstring once converted to grayscale. As for the squared query image, it can be described by four descriptors, one for each of the four polar directions. Indeed, as we don't know the orientation of the query image, four descriptors have to be built to ensure that one of them corresponds to the proper orientation of the related image in the database.

2.3. Retrieving the Best Answer

From that point, dissimilarity between images of the database and the queried book cover can be computed really efficiently as the Hamming distance between the corresponding bitstrings. This calculation is done extremely efficiently on modern CPUs that provide specific instructions to perform bitXOR and bitcounting operations. The couple of images that minimizes this distance tells us both which image of the database is the most similar to the query image, and with which orientation.



(a) Four independently chosen location pairs. (b) Four location pairs invariant under rotations in 90° increments.

Fig. 2. Comparison on random and well-chosen location pairs.

3. SO-BRIEF: IMPROVED BRIEF DESCRIPTOR FOR SQUARE IMAGES

Features described by BRIEF descriptors are different in many ways from entire images that we are trying to describe in this paper. Thus some work has to be done to adapt BRIEF to specific characteristics of quite wide square objects.

This way we define so-BRIEF, a BRIEF descriptor suitable to Squared Objects.

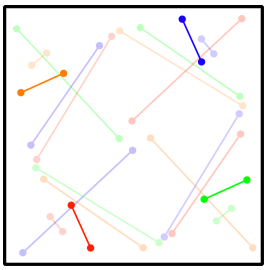
3.1. Taking Advantage of Invariants of Squares

Contrary to features studied by BRIEF, square images cannot be rotated to every possible angle, but only to the four polar directions : north, east, south and west. Therefore it only needs four descriptions of the query image to make the matching process rotation invariant. Nonetheless, computing four different descriptors for a query image is not necessary in practice. This step can indeed be sped up thanks to the specific square shape of the images to be described.

When deciding on the spatial arrangement of the binary tests, instead of independently choosing each location pairs (Fig.2a) we can take advantage of the invariants of squares by choosing quadruplets of pairs, making these locations invariant under rotations by 90° increments. As explained by Fig. 3, this well-chosen spatial arrangement enables the fast computation of the four polar directions descriptors, each one being in this case a simple known permutation of another.

3.2. Learning Good Binary Features

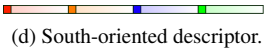
As initiated by the ORB descriptor [11], it is useful to choose binary tests which show not only high variance but also low correlation. However, the greedy search algorithm described in the paper [11] needs to be adapted. First, it has to take into account the previously introduced quadruplets of location pairs. Then we may also eventually consider squared images way bigger than the pixel patches used for BRIEF



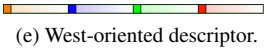
(a) Quadruplets of well-chosen location pairs.

(b) North-oriented descriptor.

(c) East-oriented descriptor.



(d) South-oriented descriptor.



(e) West-oriented descriptor.

Fig. 3. Well-chosen quadruplets of location pairs enable to compute the four descriptors in one. Indeed the n_d-dimensional descriptor computed for the north-oriented image (see Fig.3b) is composed of $n_d/4$ bits related to as many primary binary tests (the blue ones on Fig.3a), successively followed by $n_d/4$ bits related to the primary locations rotated by $90°$ (green pairs), by $180°$ (red pairs) and by $270°$ (orange pairs).
The descriptors for east, south and west-oriented images are then simple permutations of the north one.

and ORB descriptors. Using 256×256 pixel squared images means over one billion possible tests (only 200000 with 31×31 pixel patches used by ORB). Therefore, we cannot search among all possible binary tests, and have to sparingly adapt the greedy search algorithm into the following:

1: Randomly choose a large number of pair tests (typically $100 * n_d$).
2: Run each binary test against all images of the training database[1].
3: For each quadruplet of pair tests, compute a total variance as the sum of their four variances.
4: Sort feature quadruplets by decreasing total variance, forming the vector $\mathbf{T}$.
5: Initialize the n_d-dimensional result vector $\mathbf{R}$.
6: **while** the result vector $\mathbf{R}$ does not contain n_d features **do**
7: **if** $\mathbf{T}$ is empty **then**
8: Repeat the algorithm with a higher threshold.
9: **else**
10: Remove the first quadruplet $\mathbf{Q}$ from $\mathbf{T}$.
11: **end if**
12: **if** [the absolute correlation inside $\mathbf{Q}$ is below the threshold] **and** [the absolute correlation between $\mathbf{Q}$ and $\mathbf{R}$ is below the threshold] **then**
13: Add $\mathbf{Q}$ to $\mathbf{R}$.
14: **end if**
15: **end while**

This algorithm ensures that selected features are some of the most variant ones on the learning database, and that no couple of features is correlated above a given threshold. We measure the importance of this learning step later in this paper.

[1]For this research, we built up a personal 1.1GB training database balanced between 500 book covers, 500 movie posters, 500 CD covers and 500 concert posters.

4. IMPLEMENTATION AND RESULTS

Until now, we have presented the main ideas of so-BRIEF but have not discussed the actual values of different parameters yet.

In this section, we will discuss the effects of each parameter and then try experimentally to find their optimal values. In order to do this, we will compute a discriminancy score for a set of 40 images consisting in four versions of 10 images: the original rectangular image, a blurred version, a noised one, and a poorly cut one (simulating a bad corner detection). A so-BRIEF descriptor is built for each of them and Hamming distance is computed against all images of the previously described database. For each of the 40 test images, a discriminancy score is calculated:

$$discriminancy = \frac{m_1 - m_0}{\sigma_1} \tag{3}$$

where m_0 is the distance[2] between the test image and its related image in the database, and (m_1, σ_1) are the mean and standard deviation of the distances between the test image and all the other images in the database.

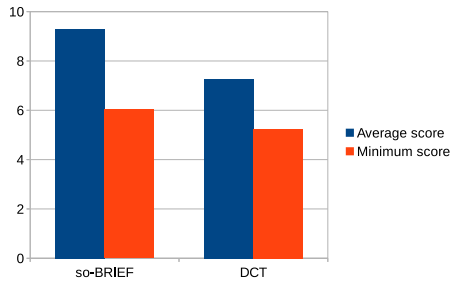
4.1. Finding Good Parameter Values

Firstly, the width w_p of squared images should be settled. A usual database is made of high resolution images, which consequently don't limit the value of this parameter. However, the query image could impose some restrictions, because of a limited camera sensor resolution for example.

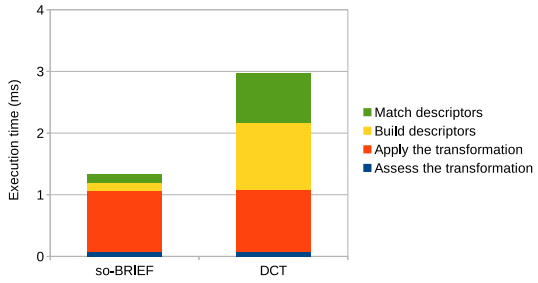
Secondly, the size of the smoothing kernel for binary tests should also be discussed. To speed up the process, smoothing is performed using integral images, as done by ORB [11]. The width w_t of the sub-window is of course related to the size of the image. Comparable discriminancy results are indeed obtained for equal w_p/w_t ratios. Low values of w_t would result in high noise sensitivity, whereas high values would increase descriptor features correlation. An optimum can thus be experimentally approached.

Finally the number n_d of features is also variable. In term of memory and efficiency, low values of n_d are obviously preferred. On the other hand, it seems at first sight that high values of n_d would imply a better discriminancy. Yet, for a given image size, there is a value of n_d from which features become too correlated to increase the descriptor distinctiveness. That is why for a given (w_p, w_t) couple there is an optimal number of features.

[2]Let us remember that the distance between an image in the database and a query image is defined as the minimum Hamming distance between the descriptor of the first and all of the four descriptors of the later.



(a) Average and minimum discriminancy scores.



(b) Execution time on a 960×540px image.

Fig. 4. Comparison of so-BRIEF and 2D-DCT discriminancy and efficiency.
For so-BRIEF: $n_d = 512$, $w_p = 64$ and $w_t = 8$.
For DCT, the 8×8 first coefficients of the 64×64 2D-DCT are stored.

Prohibiting high values of w_p, we for example find for $w_p = 64$ that the best sub-window size is $w_t = 10$ with an optimal number of features between 500 and 1000. Thus, henceforth we would use $n_d = 512$, $w_p = 64$ and $w_t = 10$.

4.2. Results

In this subsection, we assess the robustness of our method by computing the discriminancy score defined by Eq.3 on a set of 10 real-life test images against our database of 2000 images. Matlab scripts are launched on a 2.50GHz Intel Core i5-42000M, running a x86_64 Ubuntu distribution.

4.2.1. Evaluation of the learning step

Experimentation proved how essential it is to learn good binary features on the training dataset. Indeed, with learned uncorrelated features, the average discriminancy score is increased by exactly 50% in our experiments, and all the real-life test images are correctly matched, whereas this is not the case without the learning step.

4.2.2. Evaluation against other methods

Many different techniques can be used for image recognition. Applications are usually based on feature descriptors such as the SIFT [12], or on eigenvectors [13] but these techniques are way slower than binary descriptors and thus do not take place in the context of our study. The computation time of the methods based on the Harris corner and the SIFT is indeed more than 0.03s and 0.37s respectively. One could also try color indexing methods but these have been shown to inevitably often retrieve false matches [14]. Finally, 2D Discrete Cosine Transform (2D-DCT) techniques have been proved to be very efficient [15] and fast, as the 2D-DCT can be computed with $O(n \log(n))$ time complexity. It is thus a perfect competitor to our descriptor.

The DCT descriptor of an image is composed of the first low frequency coefficients of its 2D-DCT. On our real-life test images, best results have been obtained by keeping the 8×8 first coefficients. Storing them with single-precision floating-point format leads to 256-byte descriptors. In Fig. 4, we compare its performance to so-BRIEF. The latter is used with $n_d = 512$, leading to shorter 64-byte descriptors, and yet shows to be more discriminative, with a gain of 28% on average and 15% in worst cases (Fig.4a). These better performances can be explained by the specific construction of so-BRIEF descriptors. Indeed, while the DCT-based technique only captures low frequency data, so-BRIEF includes information from low frequency (with features based on close pixel pairs) to high frequency (with features based on distant pixel pairs).

4.2.3. Discussion about a content-driven object detector

One of our first assumption in this paper was that we were able to extract the object from the background of the original image. With such a configuration, the entire so-BRIEF recognition process runs more than 700 times a second, making this assumption plausible. One could indeed rapidly try several possible quadrilaterals and select the one most likely to correspond to an object, creating this way a new kind of content-driven detector for real-life rectangular objects.

5. CONCLUSION

In this paper, we introduced the so-BRIEF descriptor that represents rectangular objects as binary strings. It adapts BRIEF to image description, uses ideas from ORB and takes advantage of the specific shape of studied objects. It shows very good distinctiveness compared to DCT-based descriptors while running faster. This is very promising because it opens the door to a new range of applications, possibly running in real time on devices with limited computational power. Future possible work could aim at developing new applications making the most of so-BRIEF possibilities.

6. REFERENCES

[1] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, vol. 2, pp. 2161–2168, 2006.

[2] T. Cai, "Recognizing art pieces in subway using computer vision," 2011.

[3] Q.F. Tan and D. Lau, "Modified eigenimage algorithm for painting image retrieval," Available in http://web.stanford.edu/class/ee368/Project_07/reports/ee368group07.pdf.

[4] J. Košecká and W. Zhang, "Extraction, matching, and pose recovery based on dominant rectangular structures," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 274–293, 2005.

[5] B. Micusik, H. Wildenauer, and J. Kosecka, "Detection and matching of rectilinear structures," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–7, 2008.

[6] Intel, "Sse4 programming reference," *D91561-003*, pp. 156–158, 2007.

[7] ARM, "Realview compilation tools," p. 5.63, 2010.

[8] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, vol. 2, pp. 775–781, 2005.

[9] L. Chang, M.M. Duarte, L.E. Sucar, and E.F. Morales, "A bayesian approach for object classification based on clusters of sift local features," *Expert Systems With Applications*, vol. 39, no. 2, pp. 1679–1686, 2012.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, pp. 778–792, 2010.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. IEEE, pp. 2564–2571, 2011.

[12] D. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, vol. 2, pp. 1150–1157, 1999.

[13] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[14] M.A. Stricker and M. Orengo, "Similarity of color images," in *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*. International Society for Optics and Photonics, pp. 381–392, 1995.

[15] Š. Obdržálek and J. Matas, "Object recognition using local affine frames on maximally stable extremal regions," in *Toward Category-Level Object Recognition*, pp. 83–104. Springer, 2006.