# From Bits to Images: Inversion of Local Binary Descriptors

Emmanuel d'Angelo, Laurent Jacques, Alexandre Alahi and Pierre Vandergheynst

**Abstract**—Local Binary Descriptors are becoming more and more popular for image matching tasks, especially when going mobile. While they are extensively studied in this context, their ability to carry enough information in order to infer the original image is seldom addressed. In this work, we leverage an inverse problem approach to show that it is possible to directly reconstruct the image content from Local Binary Descriptors. This process relies on very broad assumptions besides the knowledge of the pattern of the descriptor at hand. This generalizes previous results that required either a prior learning database or non-binarized features. Furthermore, our reconstruction scheme reveals differences in the way different Local Binary Descriptors capture and encode image information. Hence, the potential applications of our work are multiple, ranging from privacy issues caused by eavesdropping image keypoints streamed by mobile devices to the design of better descriptors through the visualization and the analysis of their geometric content.

**Index Terms**—Computer Vision, Inverse problems, Image reconstruction, BRIEF, FREAK, Privacy

## 1 Introduction

How much, and what type of information is encoded in a keypoint descriptor? Surprisingly, the answer to this question has seldom been addressed directly. Instead, the performance of keypoint descriptors is studied extensively through several image-based benchmarks following the seminal work of Mikolajczyk and Schmid [1] using Computer Vision and Pattern Recognition task-oriented metrics. These stress tests aim at measuring the stability of a given descriptor under geometric and radiometric changes, which is a key to success in matching templates and real world observations. While precision/recall scores are of primary interest when building object recognition systems, they do not tell much about the intrinsic quality and quantity of information that are embedded in the descriptor. Indeed, these benchmarks are informative about the context in which a descriptor performs well or poorly, but not why. As a consequence, descriptors were mostly developed empirically by benchmarking new ideas against some image matching datasets.

Furthermore, there is a growing trend towards the use of image recognition technologies from mobile handheld devices such as the smartphones combining high quality imaging parts and a powerful computing platform. Application examples include image search and landmark recognition [2] or augmented media and advertisement [3]. To reduce the amount of data exchanged between the mobile and the online knowledge database, it is tempting to use the terminal to extract image features and send only these features over the network. This data is obviously sensitive since it encodes what the user is viewing. Hence it is legitimate to wonder if its interception could lead to a privacy breach.

Recently, two papers addressed the task of reconstructing image parts from their descriptors. First, the inspirational work in [4] showed that ubiquitous interest points such as SIFT [5] (but SURF [6] could be used as well) suffice to reconstruct plausible source images. This method is based on an image patch database indexed by their SIFT descriptors and then proceeds by successive queries, replacing each input descriptor by the corresponding patch retrieved in the reference database. Finally, reconstructed images are obtained from these smaller parts using Poisson interpolation [7]. This algorithm produces attracting results and clearly answers the initial question of the authors: the privacy of cloud-based image recognition application users is not protected by the sole fact that feature points are sent to a server instead of images. However, this algorithm tells us little about the information embedded in the descriptor: retrieving an image patch from a query descriptor leverages the matching capabilities of SIFT which are now well established by numerous benchmarks and were actually key for its wide adoption.

In the second and most recent paper [8], the authors consider state-of-the-art object recognition pipelines that compute Histograms of Oriented Gradient (HOG) [9] from sliding windows applied on the image to analyze. While HOG descriptors are closely related to the descriptor part of SIFT, Vondrick and co-authors study four different and more complex descriptor inversion techniques where the correspondence between a descriptor and the reconstructed image patch is not obtained from a single

- *E. d'Angelo is with Advanced Silicon S.A., 1004 Lausanne, Switzerland. E-mail: emmanuel.dangelo@advancedsilicon.com. This work was done while the author was at EPFL/LTS2.*
- *P. Vandergheynst is with the Signal Processing Labs (LTS2), Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. E-mail: pierre.vandergheynst@epfl.ch*
- *Alexandre Alahi is with the Stanford Vision Lab, Stanford University, CA 94305-9020, USA. E-mail: alahi@stanford.edu*
- *Laurent Jacques is with the ICTEAM institute, ELEN Department, Université catholique de Louvain (UCL), 1348 Louvain-la-Neuve, Belgium. E-mail: laurent.jacques@uclouvain.be*

nearest neighbor in a patch database but from a type of sparse regression in this database instead.

Thus, both related work [4, 8] rely on the prior existence of an image patch collection that is *large enough* to be generic. In this paper, we propose instead two algorithms that aim at reconstructing image patches from local descriptors that exploit only the values contained in the descriptors and with very little additional constraints. Besides the practical interest of getting rid of a possibly huge patch database, our approach provides a different information about the patch descriptors: since only one descriptor is available at a time of reconstruction, the proposed algorithms only rely on the information intrinsically contained in the said descriptor and not on its specificity inside any given database.

We consider descriptors made of local image intensity differences, which are increasingly popular in the Computer Vision community, for they are not very demanding in computational power and hence well suited for embedded applications. The first algorithm that we describe works on *real-valued* difference descriptors, and addresses the reconstruction process as a regularized deconvolution problem. The second algorithm leverages some recent results from 1-bit Compressive Sensing (CS) [10, 11] to reconstruct image parts from *binarized* difference descriptors, and hence is of great practical interest because these descriptors are usually available as bitstrings rather than as real-valued vectors.

### Contributions

The contributions of this paper are twofold. First, we extend the seminal work of [4] by showing that an inverse problem approach suffices to invert a local image patch descriptor provided that the descriptor is a local difference operator, thus avoiding the need to build an external database beforehand. Second, conversely to most literature on the subject of 1-bit Compressed Sensing still focused on theoretical issues and tested on synthetic signals [10, 11, 28], we present a real application where an algorithm can efficiently estimate image features from a collection of binary descriptors.

An earlier version of this work appeared in [12]. However, it was limited to real-valued descriptors, hence we greatly extend it by proposing an algorithm for 1-bit measurements. We also replace the Total Variation prior of [12] by another *analysis sparsity prior* relying on wavelets projections. This allows us to design two reconstruction algorithms (for real and binarized descriptors) that optimize over similar quantities, indeed easing the reading. Furthermore, we had to drop the detailed derivation of the real-valued algorithm therein for brevity concerns, and take advantage of the current paper to make the technical steps more explicit.

### Notations

In this paper, we make extensive use of the following notations. Matrices and vectors are denoted by bold letters or symbols (e.g., $\boldsymbol{\Phi}$, $\boldsymbol{x}$) while light letters are associated to scalar values (e.g., scalar functions, vector components or dimensions). The scalar product between two $N$-length vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is written $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{N} x_i y_i$, while their Hadamard product $\boldsymbol{x} \odot \boldsymbol{y}$ is such that $(\boldsymbol{x} \odot \boldsymbol{y})_i = x_i y_i$ for $1 \leq i \leq N$. Since we work only with real matrices, the adjoint of a matrix $\boldsymbol{A}$ is $\boldsymbol{A}^* = \boldsymbol{A}^T$. The vector of ones is written $\mathbf{1} = (1, \cdots, 1)^T$ and the identity matrix is denoted $\mathbf{Id}$.

Most of the time, we will "vectorize" 2-D images, *i.e.*, an image or a patch image $\boldsymbol{x}$ of dimension $N_1 \times N_2$ is represented as a $N$-dimensional vector $\boldsymbol{x} \in \mathbb{R}^N$ with $N = N_1 N_2$. This allows us to represent any linear operation on $\boldsymbol{x}$ as a simple matrix-vector multiplication. One important linear operator is the 2-D wavelet analysis operator $\boldsymbol{W}$ with $\boldsymbol{W}^T$ the corresponding synthesis operator. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$, $\boldsymbol{W}\boldsymbol{x}$ is then a vector of wavelet coefficients and $\boldsymbol{W}^T\boldsymbol{y}$ a patch with the same size as $\boldsymbol{x}$.

We denote by $\|\boldsymbol{x}\|_p = (\sum_i |x_i|^p)^{1/p}$ with $p \geq 1$ the $\ell_p$-norm of $\boldsymbol{x} \in \mathbb{R}^N$, reserving the notation $\|\cdot\|$ for $p = 2$ and with $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$. The $\ell_0$ "norm" of $\boldsymbol{x}$ is $\|\boldsymbol{x}\|_0 = \#\{i : x_i \neq 0\}$. Correspondingly, for $1 \leq p \leq +\infty$, a $\ell_p$-ball of radius $\lambda$ is the set $B_p(\lambda) = \{\boldsymbol{x} \in \mathbb{R}^N : \|\boldsymbol{x}\|_p \leq \lambda\}$.

We use also the following functions. We denote by $(\boldsymbol{x})_+$ the non-negativity thresholding function, which is defined componentwise as $(\lambda)_+ = (\lambda + |\lambda|)/2$, and $(\boldsymbol{x})_- = -(-\boldsymbol{x})_+$. The sign function $\operatorname{sign} \lambda$ is equal to 1 if $\lambda > 0$ and $-1$ otherwise.

In the context of convex optimization, we denote by $\Gamma^0(\mathbb{R}^N)$ the class of proper, convex and lower-semicontinuous functions of the finite dimensional vector space $\mathbb{R}^N$ to $(-\infty, +\infty]$ [13]. The indicator function $\imath_{\mathcal{S}} \in \Gamma^0(\mathbb{R}^N)$ of a set $\mathcal{S}$ maps $\imath_{\mathcal{S}}(\boldsymbol{x})$ to 0 if $\boldsymbol{x} \in \mathcal{S}$ and to $+\infty$ otherwise. For any $F \in \Gamma^0(\mathbb{R}^N)$ and $\boldsymbol{z} \in \mathbb{R}^N$, the *Fenchel-Legendre* conjugate function $F^*$ is

$$F^*(\boldsymbol{z}) = \max_{\boldsymbol{x} \in \mathbb{R}^N} \ \langle \boldsymbol{z}, \boldsymbol{x} \rangle - F(\boldsymbol{x}),$$

while, for any $\lambda > 0$, its *proximal operator* reads:

$$\operatorname{prox}_{\lambda F} \boldsymbol{z} = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^N} \ \lambda F(\boldsymbol{x}) + \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|^2.$$

For $F = \imath_{\mathcal{S}}$ for some convex set $\mathcal{S} \subset \mathbb{R}^N$, the proximal operator of $\operatorname{prox}_{\lambda F}$ simply reduces to the orthogonal projection operator on $\mathcal{S}$ denoted by $\operatorname{proj}_{\mathcal{S}}$.

## 2 Local Binary Descriptors

In this paper, we are interested in reconstructing image patches from binary descriptors obtained by quantization of local image differences, such as BRIEF [14] or FREAK [15]. Hence, we will refer to these descriptors as Local Binary Descriptors (LBDs) in the sequel. In a standard Computer Vision and Pattern Recognition application, such as object recognition or image retrieval, an interest point detector such as Harris corners [16], SIFT [5] or FAST [17] is first applied on the images to locate interest points. The regions surrounding these keypoints are then described by a feature vector, thus replacing the raw light intensity values by more meaningful information such as histograms of gradient orientation or Haar-like analysis coefficients. In the case of LBDs, the feature vectors are made of local binarized differences computed according to the generic process described below.

### 2.1 Generic Local Binary Descriptor model

A LBD of length $M$ describing a given image patch of $\sqrt{N} \times \sqrt{N} = N$ pixels can be computed by iterating $M$ times the following three-step process:

1) compute the Gaussian average of the patch at two locations $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ with variance $\sigma_i$ and $\sigma'_i$ respectively;
2) form the difference between these two measurements;
3) binarize the result by retaining only its sign.

Reshaping the input patch as a column vector $\mathbf{p} \in \mathbb{R}^N$, the first two steps in the above procedure can be merged into the application of a single linear operator $\boldsymbol{\mathcal{L}}$:

$$\begin{aligned} \boldsymbol{\mathcal{L}} &: \mathbb{R}^N \to \mathbb{R}^M \\ &\boldsymbol{p} \mapsto \left(\langle \boldsymbol{\mathcal{G}}_{\boldsymbol{q}_i,\sigma_i}, \boldsymbol{p}\rangle - \langle \boldsymbol{\mathcal{G}}_{\boldsymbol{q}'_i,\sigma'_i}, \boldsymbol{p}\rangle\right)_{1 \leqslant i \leqslant M}, \end{aligned} \quad (1)$$

where $\boldsymbol{\mathcal{G}}_{\boldsymbol{q},\sigma} \in \mathbb{R}^N$ denotes a (vectorized) two-dimensional Gaussian of width $\sigma$ centered in $\boldsymbol{q} \in \mathbb{R}^2$ (Fig. 1, top) with $\|\boldsymbol{\mathcal{G}}_{\boldsymbol{q},\sigma}\|_1 = 1$. As illustrated in Fig. 1-bottom, since $\boldsymbol{\mathcal{L}}$ is a linear operator, it can be represented by a matrix $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{M \times N}$ multiplying $\boldsymbol{p}$ and whose each row $\boldsymbol{\mathcal{L}}_i$ is given by

$$\boldsymbol{\mathcal{L}}_i = \boldsymbol{\mathcal{G}}_{\boldsymbol{q}_i,\sigma_i} - \boldsymbol{\mathcal{G}}_{\boldsymbol{q}'_i,\sigma'_i}, \quad 1 \leq i \leq M. \quad (2)$$

We will take advantage of this decomposition interpretation to avoid explicitly writing $\boldsymbol{\mathcal{L}}$ later on.

The final binary descriptor is obtained by the composition of this sensing matrix with a component-wise quantization operator $\mathcal{B}$ defined by $\mathcal{B}(\boldsymbol{x})_i = \operatorname{sign} x_i$, so that, given a patch $\boldsymbol{p}$, the corresponding LBD reads

$$\bar{\boldsymbol{p}} := \mathcal{B}(\boldsymbol{\mathcal{L}}\boldsymbol{p}) \in \{-1,+1\}^M.$$

Note that we have chosen this definition of $\mathcal{B}$ to be consistent with the notations of [18]. Implementations of LBDs will of course use the binary space $\{0,1\}^M$ instead, since it fits naturally with the digital representation found in computers.

From the description of LBDs, it is clear that they involve only simple arithmetic operations. Furthermore, the distance between two LBDs is measured using the Hamming distance, which is a simple bitwise exclusive-or (XOR) instruction [14, 15]. Hence, computation and matching of LBDs can be implemented efficiently, sometimes even using hardware instructions (XOR), allowing their use on mobile platforms where computational power and electric consumption are strong limiting constraints. Since they also provide good matching performances, LBDs are getting more and more popular over SIFT and SURF: combined with FAST for the keypoint detection, they provide a fast and efficient feature extraction and matching pipe-line, producing compact descriptors that can be streamed over networks.

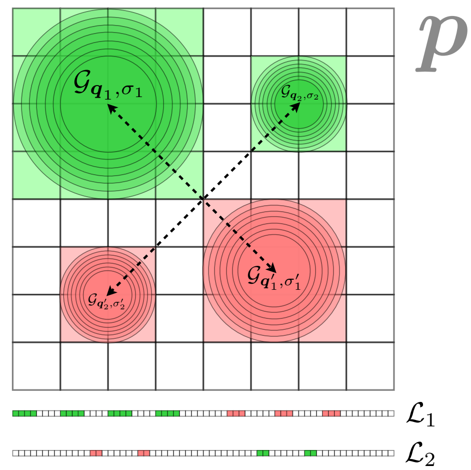



Fig. 1: Example of a local descriptor for an $8 \times 8$ pixels patch and the corresponding sensing matrix. Only two measurements of the descriptor are depicted; each one is produced by subtracting the Gaussian mean in the lower (red) area from the corresponding upper (green) one. All the integrals are normalized by their area to have values in $[0, 1]$. Below this, the corresponding vectors.

Typically, a 32-by-32 pixels image patch (1024 bytes in 8 bit grayscale format) can be reduced to a vector of only 256 measurements [14] coded with 256 *bits*. A typical floating-point descriptor such as SIFT or SURF would require instead 64 float values, *i.e.*, 256 *bytes* for the same patch, eight times the LBD size, and the distances would be measured with the $\ell_2$-norm using slower floating-point instructions.

### 2.2 LBDs, LBPs, and other integral descriptors

Unlike [4], we use LBDs in this work instead of SIFT descriptors. As we will see in Sec. 3, it is actually the knowledge of the spatial measurement pattern used by an LBD that allows us to properly define the matrix of the operator $\mathcal{L}$ in (1) as a convolution matrix. SIFT and SURF use histograms of gradient orientation instead, thus losing the precise localization information through an integration step. Hence, it seems very unlikely that our approach could be extended to these descriptors. On the other hand, it is possible to reproduce most of the algorithm described in [4] by replacing SIFT with a correctly chosen LBD to index the reference patch database, but this would bring only minor novelty.

Note also that we have coined the descriptors used here as LBDs, which are not the same as the Local Binary Patterns (LBPs) popularized by [19] for face detection. Although both LBDs and LBPs produce bit string descriptors, LBPs are obtained after binarization of image direction histograms. As such, LBPs are integral descriptors and suffer from the same lack of spatial awareness as SIFT and SURF.

### 2.3 The BRIEF and FREAK descriptors

Given two LBDs, the differences reside in the pattern used to select the size and the location of the measurement pairs $(\mathcal{G}_{q_i,\sigma_i}, \mathcal{G}_{q'_i,\sigma'_i})_{i=1}^M$. The authors of the pioneering BRIEF [14] chose small Gaussians of fixed width to bring some robustness against image noise, and tested different spatial layouts. Among these, two random patterns outperformed the others: the first one corresponds to a normal distribution of the measurement points centered in the image patch, and the second one to a uniform distribution.

Working on improving BRIEF, the authors of ORB [20] introduced a measurement selection process based on their matching performance and retained pairs with the highest selectivity. On the other hand, BRISK [21] introduced a concentric pattern to distribute the measurements inside the patch but retained only the innermost points for the descriptor, keeping the peripheral ones to estimate the orientation of the keypoint.

Eventually, the FREAK descriptor was proposed in [15] to leverage the advantages of both approaches: the learning procedure introduced with ORB and the concentric measurement layout of BRISK. The pattern was modified to resemble retinal sampling and can be seen in Fig. 2. Note that it allows for a wider overlap than the BRISK pattern. All the rings were allowed to contribute in the training phase. Consequently, the FREAK descriptor implicitly captures the image details at a coarser scale when going away from the center of the patch.

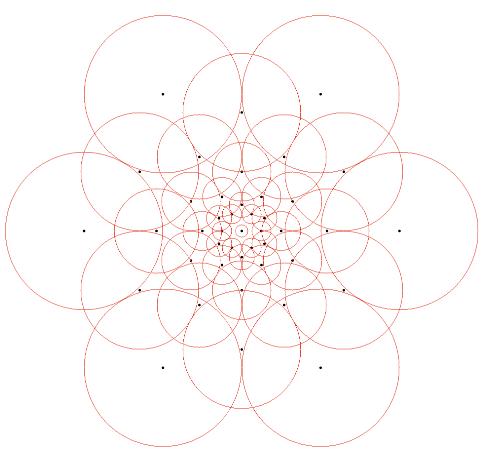


Fig. 2: The retinal pattern used by FREAK. The further a point from the center, the wider the averaging area. Hence, FREAK captures the image variations at a coarser scale on the border of the patch than in the center.

## 3 Reconstruction as an inverse problem

In this work, our goal is to demonstrate that the knowledge of the particular measurement layout of an LBD is sufficient to infer the original image patch *without any external information*, using only an inverse problem approach. Typically, a $32 \times 32$ pixels patch (1024 values) will be represented by a descriptor with 512 components. Hence, the reconstruction task is ill-posed: even without binarization of the features, there are half less measurements than unknowns. Assuming that this feature vector is represented with floating-point values, the binarization will then divide by an additional factor of 32 (the standard size of a float in bits) the amount of available information! Classically, to make this problem tractable we introduce a regularization constraint that should be highly generic since we do not know a priori the type of image that we need to reconstruct. Thus, the sparsity of the reconstructed patch in some wavelet frame appeared as a natural choice: it only requires that a patch should have few nonzero coefficients when analyzed in this wavelet frame, which is quite general.

### 3.1 Real-valued descriptor reconstruction with convex optimization

Ignoring first the quantization operator by replacing $\mathcal{B}$ with the identity function, we choose the $\ell_1$-norm to penalize the error in the data fidelity term and the $\ell_1$-norm of the wavelet coefficients as a sparsity promoting regularizer. The $\ell_1$-norm is more robust than the usual $\ell_2$-norm to the actual value of the error and it is more connected with its sign. Hence, it is hopefully a better choice when dealing with binarized descriptors. The problem of reconstructing an image patch $\hat{\boldsymbol{p}} \in \mathbb{R}^N$ given an observed binary descriptor $\bar{\boldsymbol{p}} \in \mathbb{R}^M$ then reads:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{x} \in \mathbb{R}^N}{\arg\min}\, \lambda \|\boldsymbol{\mathcal{L}} \boldsymbol{x} - \bar{\boldsymbol{p}}\|_1 + \|\boldsymbol{W}\boldsymbol{x}\|_1 + \imath_{\mathcal{S}}(\boldsymbol{x}), \qquad (3)$$

which is a sparse $\ell_1$ deconvolution problem [22]. In Eq. (3), $\|\boldsymbol{\mathcal{L}}\boldsymbol{x} - \bar{\boldsymbol{p}}\|_1$ is the *data fidelity term* that ties the solution to the observation $\bar{\boldsymbol{p}}$, $\|\boldsymbol{W}\boldsymbol{x}\|_1$ is the regularizer that constrains the patch candidate $\boldsymbol{x}$ to have a sparse representation, and $\imath_{\mathcal{S}}(.)$ is the indicator function of the *validity domain* of $\boldsymbol{x}$ that we will make explicit later.

While the objective function in (3) is convex, it is not differentiable since the $\ell_1$-norm has singular points on the axes of $\mathbb{R}^N$. Hence, we chose the *primal-dual* algorithm presented in [23] to solve this minimization problem. Instead of using derivatives of the objective which may not exist, it relies on *proximal calculus* and proceeds by alternate minimizations on the primal and dual unknowns.

The generic version of this algorithm aims at solving minimization problems of the form:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^N}{\arg\min}\, F(\boldsymbol{K}\boldsymbol{x}) + G(\boldsymbol{x}), \qquad (4)$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the *primal* variable, $\boldsymbol{K} \in \mathbb{R}^{D \times N}$ is a linear operator, and $F \in \Gamma^0(\mathbb{R}^D)$ and $G \in \Gamma^0(\mathbb{R}^N)$ are convex (possibly non-smooth) functions. The algorithm proceeds by restating (4) as a *primal-dual* saddle-point problem on both the *primal* $\boldsymbol{x}$ and its *dual* variable $\boldsymbol{u}$:

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{u}} \langle \boldsymbol{K}\boldsymbol{x}, \boldsymbol{u} \rangle + G(\boldsymbol{x}) - F^*(\boldsymbol{u}).$$

For the problem at hand, we start by decoupling the data fidelity term and the sparsity constraint in Eq. (3) by introducing the auxiliary unknowns $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^N$ such that:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{y},\boldsymbol{z}\in\mathbb{R}^N}{\arg\min}\, \lambda\|\boldsymbol{\mathcal{L}}\boldsymbol{y} - \bar{\boldsymbol{p}}\|_1 + \|\boldsymbol{W}\boldsymbol{z}\|_1 + \imath_{\mathcal{S}}(\boldsymbol{y}) + \imath_{\{\boldsymbol{0}\}}(\boldsymbol{y} - \boldsymbol{z}).$$

The term $\imath_{\{\boldsymbol{0}\}}(\boldsymbol{y} - \boldsymbol{z})$ guarantees that the optimization occurs on the bisector plane $\boldsymbol{y} = \boldsymbol{z}$. Then, defining $\boldsymbol{K} = \left(\begin{smallmatrix} \boldsymbol{\mathcal{L}} & 0 \\ 0 & \boldsymbol{W} \end{smallmatrix}\right) \in \mathbb{R}^{(M+N)\times 2N}$, we can perform our optimization (3) in the product space $\boldsymbol{x} = (\boldsymbol{y}^T, \boldsymbol{z}^T)^T \in \mathbb{R}^{2N}$ [24, 25]. In this case, (3) can be written as (4) by setting $F(\boldsymbol{K}\boldsymbol{x}) = F_1(\boldsymbol{\mathcal{L}}\boldsymbol{y}) + F_2(\boldsymbol{W}\boldsymbol{z})$, where:

$$F_1(\cdot) = \lambda\|\cdot - \bar{\boldsymbol{p}}\|_1, \qquad F_2(\cdot) = \|\cdot\|_1,$$
$$G\left(\begin{smallmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{smallmatrix}\right) = \imath_{\mathcal{S}}(\boldsymbol{y}) + \imath_{\{\boldsymbol{0}\}}(\boldsymbol{y} - \boldsymbol{z}).$$

Introducing $\boldsymbol{r} \in \mathbb{R}^M$ and $\boldsymbol{s} \in \mathbb{R}^N$ the dual counterparts of $\boldsymbol{y}$ and $\boldsymbol{z}$ respectively, and taking the Fenchel-Legendre transform of $F$, yields the desired primal-dual formulation of (3):

$$\min_{\boldsymbol{y},\boldsymbol{z}} \max_{\boldsymbol{r},\boldsymbol{s}} \langle \boldsymbol{\mathcal{L}}\boldsymbol{y}, \boldsymbol{r}\rangle + \langle \boldsymbol{W}\boldsymbol{z}, \boldsymbol{s}\rangle + G\left(\begin{smallmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{smallmatrix}\right) - F_1^*(\boldsymbol{r}) - F_2^*(\boldsymbol{s}). \quad (5)$$

An explicit formulation of $F_1^*(\boldsymbol{r})$ can be obtained by noting that, for $\varphi(\boldsymbol{r}) = \varphi'(\boldsymbol{r} - \boldsymbol{u})$, $\varphi^*(\boldsymbol{r}) = \varphi'^*(\boldsymbol{r}) + \langle \boldsymbol{r}, \boldsymbol{u}\rangle$, and that the conjugate of the $\ell_1$-norm is $\imath_{B_\infty(1)}$ [13]. $F_2^*$ is derived in [23], eventually yielding:

$$F_1^*(\boldsymbol{r}) = \imath_{B_\infty(\lambda)}(\boldsymbol{r}) + \langle \boldsymbol{r}, \bar{\boldsymbol{p}}\rangle, \quad F_2^*(\boldsymbol{s}) = \imath_{B_\infty(1)}(\boldsymbol{s}).$$

The algorithm presented in [23] requires explicit solutions for the proximal mappings of $F_1^*$, $F_2^*$ and $G$. The first two are easily computed pointwise [13, 23] as:

$$(\mathrm{prox}_{\sigma F_1^*}\, \boldsymbol{r})_i = \mathrm{sign}(r_i - \sigma\bar{p}_i) \cdot \min(\lambda, |r_i - \sigma\bar{p}_i|),$$
$$(\mathrm{prox}_{\sigma F_2^*}\, \boldsymbol{s})_i = \mathrm{sign}(s_i) \cdot \min(1, |s_i|).$$

The function $G$ is formed by the indicator of the set $\mathcal{S}$ and the indicator of the bisector plane $\{\left(\begin{smallmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{smallmatrix}\right) \in \mathbb{R}^{2N} : \boldsymbol{y} = \boldsymbol{z}\}$. An easy computation provides [25]:

$$\mathrm{prox}_{\sigma G}\left(\begin{smallmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{smallmatrix}\right) = \begin{pmatrix} \mathrm{proj}_{\mathcal{S}}\, \frac{1}{2}(\boldsymbol{y} + \boldsymbol{z}) \\ \mathrm{proj}_{\mathcal{S}}\, \frac{1}{2}(\boldsymbol{y} + \boldsymbol{z}) \end{pmatrix}.$$

Thus, its proximal mapping does not depend on any parameter.

Let us now precisely define our *validity domain* $\mathcal{S}$, in order to remove ambiguities in the definition of the program (3) that could lead to a non-uniqueness of the solution. They are due to the differential nature of $\boldsymbol{\mathcal{L}}$, *i.e.*, the descriptor of any constant patch is zero. This involves both that $\bar{\boldsymbol{p}}$ does not include any information about the average of the initial patch $\boldsymbol{p}$, and the average of $\boldsymbol{x}$ in (3) cannot be determined by the optimization.

This problem is removed by defining $\mathcal{S}$ as the intersection of two convex sets $\mathcal{S}_1$ and $\mathcal{S}_2$. The first set $\mathcal{S}_1$ arbitrarily constrains the minimization domain to stay in the set of patches whose pixel dynamic lies in $[0, h_{\mathrm{pix}}]$, *i.e.*, $\mathcal{S}_1 = \{\boldsymbol{x} \in \mathbb{R}^N : 0 \le x_i \le h_{\mathrm{pix}}\}$. In our experiments, we simply consider pixels with real values in $[0, 1]$ and consequently fix $h_{\mathrm{pix}} = 1$. The second domain $\mathcal{S}_2$ is associated to the space of patches whose pixel mean is equal to $0.5$, *i.e.*, $\mathcal{S}_2 = \{\boldsymbol{x} \in \mathbb{R}^N : \frac{1}{N}\sum_i x_i = 0.5\}$.

This gives us a first set $\mathcal{S}_1$ whose proximal mapping $\mathrm{proj}_{\mathcal{S}_1}$ is a simple clipping of the values in $[0, 1]$, while $\mathcal{S}_2$ is an hyperplane in $\mathbb{R}^N$ whose corresponding proximal mapping $\mathrm{proj}_{\mathcal{S}_2}$ is the projection onto the simplex of $\mathbb{R}^N$ of vectors with mean $0.5$. This projection can be solved efficiently using [26]. While the desired constraint set $\mathcal{S}$ is the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$, we approximate the projection on $\mathcal{S}$ by $\mathrm{proj}_{\mathcal{S}} \simeq \mathrm{proj}_{\mathcal{S}_1} \circ \mathrm{proj}_{\mathcal{S}_2}$. The correct treatment of $\mathrm{proj}_{\mathcal{S}}$ would normally require to iteratively combine $\mathrm{proj}_{\mathcal{S}_1}$ and $\mathrm{proj}_{\mathcal{S}_2}$ (e.g., running Generalized Forward-Backward splitting [24] until convergence). In our experiments, this approximation did not lead to differences in the estimated patches.

Alg. 1 summarizes the different steps involved in the resolution scheme. It requires a bound $\Gamma$ on the operator norm $\boldsymbol{K} \in \mathbb{R}^{(M+N)\times 2N}$, *i.e.*, on $\|\boldsymbol{K}\| = \max_{\boldsymbol{x}:\, \|\boldsymbol{x}\|=1} \|\boldsymbol{K}\boldsymbol{x}\|$. This is obtained by observing that $\|\boldsymbol{W}\|^2 = 1$ with a proper rescaling due to energy conservation constraints, leaving:

$$\|\boldsymbol{K}\|^2 = \|\left(\begin{smallmatrix} \boldsymbol{\mathcal{L}} & 0 \\ 0 & \boldsymbol{W} \end{smallmatrix}\right)\|^2 \leqslant \|\boldsymbol{\mathcal{L}}\|^2 + 1,$$

where $\|\boldsymbol{\mathcal{L}}\|$ can be efficiently estimated without any spectral decomposition of $\boldsymbol{\mathcal{L}}$ by using the power method [22].

While (5) may seem unnecessarily complicated at first because it involves both a minimization and a maximization subproblem, the resolution scheme is actually very efficient: it is a first-order method that involves mostly pointwise normalization and thresholding operations. The convergence rate of Alg. 1 is $\mathcal{O}(1/k)$ [23]: since the $\ell_1$-norm is not smooth, accelerated schemes with a quadratic convergence rate are not applicable. Furthermore, it involves only simple operations on images that have the size of a patch: pointwise thresholding, LBD computation and wavelet transform, that can easily be parallelized. The simplex projection [26] converges after at most $N$ iterations, where $N$ is the number of pixels in a patch. The LBD can be implemented using integral images (see Section 4.1). Thus, given a fixed number of iterations, the overall complexity of Alg. 1 is dominated by the one of the wavelet transform, *i.e.*, $O(N \log_2 N)$, which is almost linear in the patch dimension $N$.

**Algorithm 1** Primal-dual $\ell_1$ sparse patch reconstruction.

Take $\Gamma \geqslant \|\boldsymbol{K}\|_2$, choose $\tau, \sigma, \theta$ such that $\Gamma^2\sigma\tau \leqslant 1$, $\theta \in [0, 1]$ and $n$ the number of iterations
Initialize: $\boldsymbol{x}^{(0)}, \tilde{\boldsymbol{x}}^{(0)} \leftarrow \boldsymbol{0}$, and $\boldsymbol{r}^{(0)}, \boldsymbol{s}^{(0)} \leftarrow \boldsymbol{0}$
**for** $i = 0$ to $n - 1$ **do**
  $\boldsymbol{r}^{(i+1)} \leftarrow \mathrm{prox}_{\sigma F_1^*}(\boldsymbol{r}^{(i)} + \sigma\boldsymbol{\mathcal{L}}\tilde{\boldsymbol{x}}^{(i)})$
  $\boldsymbol{s}^{(i+1)} \leftarrow \mathrm{prox}_{\sigma F_2^*}(\boldsymbol{s}^{(i)} + \sigma\boldsymbol{W}\tilde{\boldsymbol{x}}^{(i)})$
  $\boldsymbol{x}^{(i+1)} \leftarrow \mathrm{proj}_{\mathcal{S}}(\boldsymbol{x}^{(i)} - \frac{\tau}{2}\boldsymbol{\mathcal{L}}^T\boldsymbol{r}^{(i+1)} - \frac{\tau}{2}\boldsymbol{W}^T\boldsymbol{s}^{(i+1)})$
  $\tilde{\boldsymbol{x}}^{(i+1)} \leftarrow \boldsymbol{x}^{(i+1)} + \theta\,(\boldsymbol{x}^{(i+1)} - \boldsymbol{x}^{(i)})$
**end for**
**return** $\hat{\boldsymbol{p}} \leftarrow \boldsymbol{x}^{(n)}$.

### 3.2 Iterative binary descriptor reconstruction

To our surprise, when implementing and testing Alg. 1 it turned out that it was able to reconstruct not only real-valued descriptors but also binarized ones, *i.e.*, it still worked without modifications for some $\bar{\boldsymbol{p}} \in \{-1, 1\}^M$ instead of $\mathbb{R}^M$. This is probably due to the choice of the $\ell_1$-norm in the data fidelity term, which tends to attach more importance to the sign of the error than to its actual value. However, the behavior of our solver in the binarized descriptor case was unstable and it consistently failed to reconstruct some image patches, yielding a null solution. Hence, we chose to leverage some recent results from 1-bit Compressive Sensing [18] to work out a dedicated binary reconstruction scheme.

In 1-bit CS [10], a sparse (or compressible) signal $\boldsymbol{x}$ is observed through the non-linear compressive model $\boldsymbol{y} = \mathcal{B}(\boldsymbol{\Phi x}) \in \{-1, 1\}^M$ where $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ is a certain sensing matrix performing measurements of $\boldsymbol{x}$ before recording their *sign* in the measurement vector $\boldsymbol{y}$. In this model, the signal amplitude is obviously lost since $\boldsymbol{y}$ is unchanged if $\boldsymbol{x} \to \lambda\boldsymbol{x}$ for any $\lambda > 0$. Despite this missing information, it has been recently shown that when the sensing matrix is a random Gaussian matrix, *i.e.*, $\boldsymbol{\Phi}_{ij} \sim_{\rm iid} \mathcal{N}(0, 1)$, any $K$-sparse vectors $\boldsymbol{x}'$ *consistent* with a $K$-sparse vector $\boldsymbol{x}$, *i.e.*, whose 1-bit CS observations match those of $\boldsymbol{x}$, is angularly close to this one. In particular, $\angle(\boldsymbol{x}, \boldsymbol{x}') := \arccos[(\|\boldsymbol{x}\|\|\boldsymbol{x}'\|)^{-1}\langle\boldsymbol{x}, \boldsymbol{x}'\rangle] = O(K/M)$. This particular fact, which stems from the loss of precision induced by $\mathcal{B}$, makes 1-bit CS really different from standard CS where two consistent and sparse vectors are shown to be truly identical if $\boldsymbol{\Phi}$ respects the *restricted isometry property* (RIP) [27].

Additionally, as expressed by the Binary $\epsilon$ Stable Embedding property (B$\epsilon$SE) [11], in case of inconsistency, the Hamming distance between $\mathcal{B}(\boldsymbol{\Phi x})$ and $\mathcal{B}(\boldsymbol{\Phi x}')$ for any pair of $K$-sparse vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ is $\epsilon$-close to their angular distance with $\epsilon = O(\sqrt{K/M})$. These observations are moreover extendable to compressible signal sets [28].

Practically, a few algorithms have been proposed to estimate a signal from its 1-bit CS observations. Amongst those, the *binary iterative hard thresholding* (BIHT) [18] is a fast and efficient greedy method which aims at iteratively enforcing the consistency of a signal estimate with the available 1-bit CS observations in the set of sparse signals.

Our main objective is to estimate an image from a given collection of LBDs, *i.e.*, following the constrained sensing methods prescribed by BRIEF and FREAK, without accessing side information (e.g., a database of LBDs of known images). This distances our methodology from CS theory which, as summarized above, relies on (random) sensing for reaching better reconstruction quality. Nevertheless, inspired by the non-linear inverse-problem approach developed for 1-bit CS, we substantially modified the functional of (3) in two ways:

1) the data fidelity term was changed to enforce bitwise consistency between the LBD computed from the reconstructed patch and the input binary descriptor;
2) to apply the same BIHT algorithm, we take as sparsity measure the $\ell_0$-norm of the wavelet coefficients instead of the relaxed version obtained with the $\ell_1$-norm.

Driven by the BIHT approach, we are interested by the solution of this new Lasso-type program [29]:

$$\hat{\boldsymbol{p}} = \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^N} \mathcal{J}(\boldsymbol{x}) \text{ s.t. } \|\boldsymbol{W x}\|_0 \leqslant K \text{ and } \boldsymbol{x} \in \mathcal{S}, \quad (6)$$

where the constraints enforces both the validity and the $k$-sparsity of $\boldsymbol{x}$ in the wavelet domain.

As in [18], we set our data fidelity term as

$$\mathcal{J}(\boldsymbol{x}) = \|[\bar{\boldsymbol{p}} \odot \mathcal{B}(\mathcal{L}\boldsymbol{x})]_-\|_1.$$

Qualitatively, $\mathcal{J}$ measures the LBD consistency of $\boldsymbol{x}$ with $\boldsymbol{p}$, with $\mathcal{J}(\boldsymbol{x}) = 0$ iff $\mathcal{B}(\boldsymbol{\mathcal{L}x}) = \bar{\boldsymbol{p}}$. Each component of the Hadamard product in the definition of $\mathcal{J}$ is either positive (both signs are the same) or negative. Since the negative function sets to 0 the consistent components, the $\ell_1$-norm finally adds the contribution of each inconsistent entry. Note that at the time of writing we do not know a solution for the proximal mapping associated to this data fidelity term $\mathcal{J}$, which explains our choice for BIHT over the primal-dual solver used in the previous section.

Similarly to the way Iterative Hard Thresholding aims at solving an $\ell_0$-Lasso problem [30], BIHT finds one solution of (6) by repeating the three following steps until convergence:

1) computing a step of gradient descent of the data fidelity term;
2) enforcing sparsity by projecting the intermediate estimate to the set of patches with at most $K$ non-zero coefficients;
3) enforcing the mean-value constraint on the result.

This last operation was already studied in the previous section for the real-valued case: it is the projection onto the set $\mathcal{S}$. The $\ell_0$-norm constraint is applied by Hard Thresholding and amounts to keeping the $K$ biggest coefficients of the wavelet transform of the estimate and discarding the others. We write this operation $\mathcal{H}_K$. Finally, unlike in the primal-dual algorithm, the gradient descent of $\mathcal{J}$ has to be computed. Lemma 5 in [18] applies in our case providing the subgradient:

$$\partial\mathcal{J}(\boldsymbol{x}) \ \ni \ \tfrac{1}{2}\boldsymbol{\mathcal{L}}^T\big(\mathcal{B}(\boldsymbol{\mathcal{L}x}) - \bar{\boldsymbol{p}}\big),$$

*i.e.*, the back-projection of the binary error.

Putting everything together, we obtain Alg. 2 that is the adaptation of BIHT to the reconstruction of image patches from their LBD representation. Again, this algorithm is made of simple elementary steps. The parameter $\tau = 1/M$ guarantees that the current solution $\boldsymbol{x}^{(i)}$ and the gradient step $\frac{\tau}{2}\boldsymbol{\mathcal{L}}^T\big(\bar{\boldsymbol{p}} - \mathcal{B}(\boldsymbol{\mathcal{L}x}^{(i)})\big)$ have comparable amplitudes [18]. Since $M$ is determined from the LBD size, only the patch sparsity level $K$ in the wavelet basis must be tuned (see Sec. 4.1). In our experiments, however, the algorithm was not very sensitive to the value of $K$. Similarly to Alg. 1, this BIHT-derived optimization process involves only simple operations on small images (the input patches): computation of the LBD response, wavelet transform and point wise thresholding. Thus, the complexity of the algorithm remains low for a given number of iterations (typically, $n = 200$) and, as for Alg. 1, it is mainly bounded by the complexity of the wavelet transform. Furthermore, these operations can be parallelized in a computationally efficient implementation.

**Algorithm 2** BIHT patch reconstruction.

1: Take $\tau = 1/M$, choose $K$ the number of non-zero coefficients and $n$ the number of iterations
2: Initialize: $\boldsymbol{x}^{(0)} \leftarrow 0$ and $\boldsymbol{a}_0 \leftarrow 0$.
3: **for** $i = 0$ to $n-1$ **do**
4: $\quad \boldsymbol{a}^{(i+1)} \leftarrow \boldsymbol{x}^{(i)} + \frac{\tau}{2}\boldsymbol{\mathcal{L}}^T(\bar{\boldsymbol{p}} - \mathrm{sign}(\boldsymbol{\mathcal{L}}\boldsymbol{x}^{(i)}))$
5: $\quad \boldsymbol{b}^{(i+1)} \leftarrow \mathcal{H}_K(\boldsymbol{W}\boldsymbol{a}^{(i+1)})$
6: $\quad \boldsymbol{x}^{(i+1)} \leftarrow \mathrm{proj}_{\mathcal{S}}\left(\boldsymbol{W}^T\boldsymbol{b}^{(i+1)}\right)$
7: **end for**
8: **return** $\hat{\boldsymbol{p}} \leftarrow \boldsymbol{x}^{(n)}$.

# 4 Results and discussion

## 4.1 Implementation details

For the reconstruction tests presented in this section, we re-implemented two of the different LBDs: BRIEF and FREAK. For BRIEF, we chose a uniform distribution for the location of the Gaussian measurements, whose support was fixed to $3 \times 3$ pixels, following the original paper [14]. For FREAK, we did not take into account the orientation of the image patches (see [15], Sec. 4.4) but we also implemented two variants:

- EX-FREAK, for EXhaustive-Freak, computes all the possible pairs from the retinal pattern;
- RA-FREAK, for RAndomized-FREAK, randomly selects its pairs from the retinal pattern.

All the operators were implemented in C++ with the OpenCV library[1] and used the same codebase for fair comparisons, varying only in the measurement pair selection. The code used to generate the examples in this paper is available online and can be retrieved from the page http://lts2www.epfl.ch/software/.

The sensing operator was implemented in the following way:

- the forward operator $\boldsymbol{\mathcal{L}}$ is obtained through the use of integral images for a faster computation of the Gaussian weighted integrals. This approximation has become standard in feature descriptor implementations since it allows a huge acceleration of the computations, see for example [6];
- the backward operator $\boldsymbol{\mathcal{L}}^T$ is the combination of the frame vectors of the considered LBD, weighted by the input vector of coefficients. Hence, we avoid explicitly forming $\boldsymbol{\mathcal{L}}^T$ by computing on the fly the image representation of each vector $\boldsymbol{\mathcal{L}}_i$ of (2).

In the previous sections, we have proposed two algorithms that aimed at reconstructing an image patch from the corresponding local descriptor. In order to assess their relevance and the quality of the reconstructions we have applied them on whole images according to the following protocol:

1) an image is divided into patches of size $\sqrt{N} \times \sqrt{N}$ pixels, with an horizontal and vertical offset of $N_{\text{off}}$ pixels between each patch;
2) each patch is reconstructed independently from its LBD representation using the additional constraint that its mean should be $0.5$, *i.e.*, the mean of the input dynamic range;
3) reconstructed patches are back-projected to their original image position. Wherever patches overlap, the final result is simply the average of the reconstructions.

Hence, the experiments introduce an additional parameter which is the offset between the selected patches.

Note that in contrast with [4] our methods do not require the use of a seamless patch blending algorithm. Simple averaging does not introduce artifacts. Also, we do not require the knowledge of the scale and orientation of the keypoint to reconstruct: we assume it is a patch of fixed size aligned with the image axes. This is in line with the 1-bit feature detection and extraction pipeline: the genuine FAST detector does not consider scale and orientation, and the BRIEF descriptor is not rotation or scale-invariant. Later descriptors such as FREAK were trained in an affine-invariant context; hence by considering only fixed width and orientation our algorithm is suboptimal.

We used patches of $32 \times 32$ pixels, LBDs of 512 measurements and ran Alg. 1 and Alg. 2 for $n = 1000$ and $n = 200$ iterations respectively. In Alg. 1, the trade-off parameter $\lambda$ was set to $0.1$. We tried different values for the sparsity $K$ in Alg. 2 (retaining between 10% and 40% of the wavelet coefficients) but the results did not vary in a meaningful way. Thus we fixed $K$ throughout all the experiments to keep the 40% greater coefficients of $\boldsymbol{W}\boldsymbol{p}$. As noted previously, LBPs are actually differential operators and they are very likely to encode the boundaries between flat image areas. Hence, we chose the Haar wavelet as analysis operator for its ties with the Total Variation image model[2] [31].

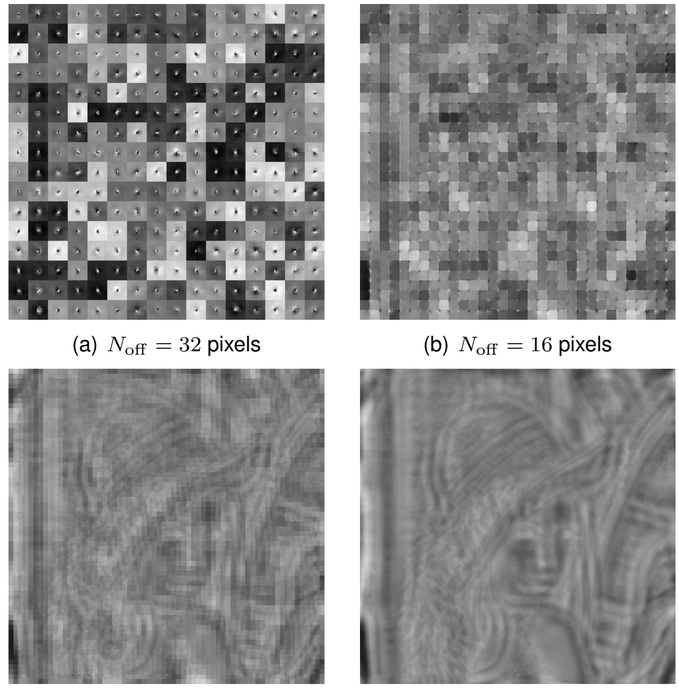
The original `Lena`, `Barbara` and `Kata` images can be seen in Fig. 3.

1. Freely available at http://opencv.org

2. Optimizing the analysis operator for reaching the best reconstruction performances (e.g., using other wavelet bases) is beyond the scope of the current paper and should be the object of future research.

(a) Lena (b) Barbara (c) Kata

Fig. 3: Original images and their designated names.



(a) $N_{\text{off}} = 32$ pixels (b) $N_{\text{off}} = 16$ pixels

(c) $N_{\text{off}} = 8$ pixels (d) $N_{\text{off}} = 1$ pixel

Fig. 5: Reconstruction of Lena from binary FREAKs. The size of the patches is kept fixed at $32 \times 32$ pixels, while their spacing is gradually reduced. We start with an offset of 32 pixels, *i.e.*, no overlap, until a dense reconstruction. In the limit when each pixel is reconstructed from its neighborhood the individual edge bits chain up and one can clearly distinguish the original image contours, like after the application of a Laplacian filter.

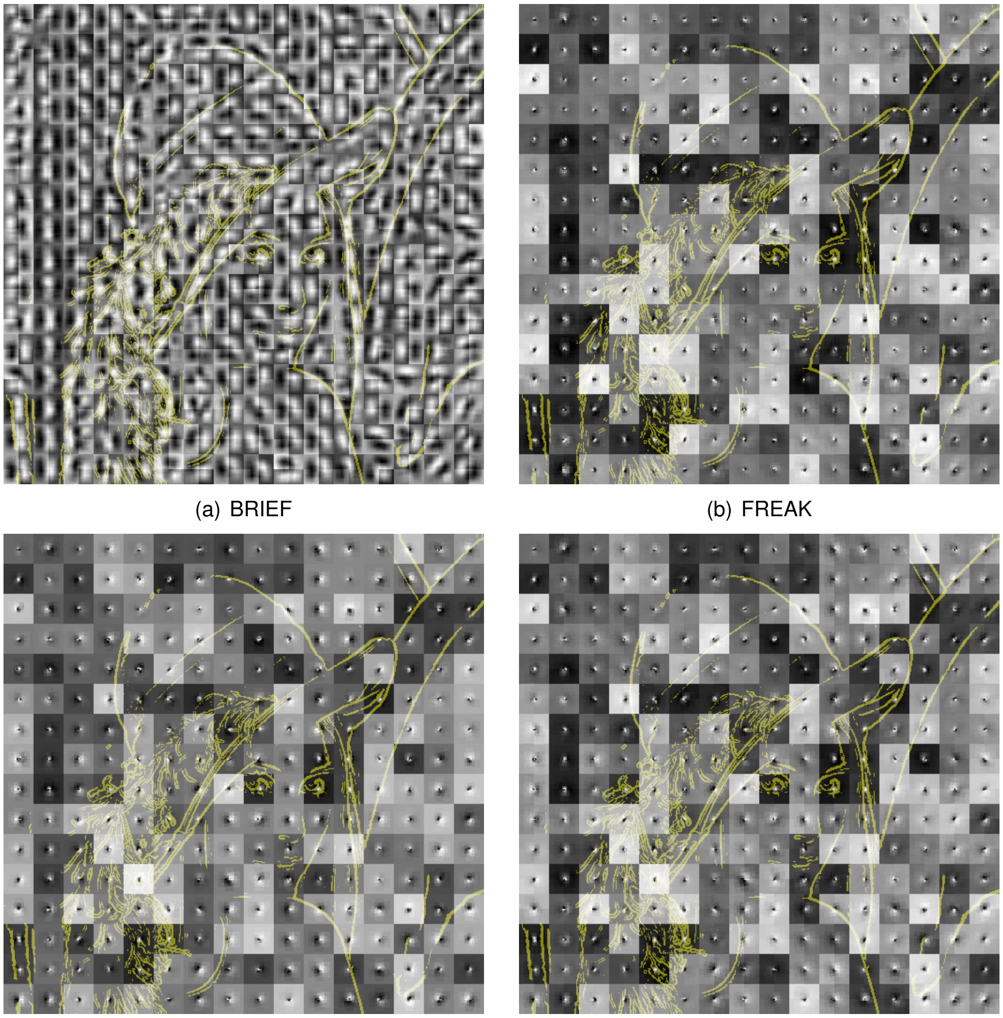
### 4.2 Reconstruction results

At first glance, the reconstruction results for non-overlapping patches visible in Fig. 4 seem very weird and have sometimes very little in common with the original image. However, if we overlay the original edges on top of the reconstructed images, one can see that each estimated patch contains a correct version of the original gradient direction. This shows that all the experimented LBDs capture the local gradient and that this information is enough for Alg. 2 to infer the original value. Even curved lines and cluttered areas are encoded by the binary descriptors: see the shoulder of Lena and the feathers of her hat (Fig. 4). While there is a significant difference between the reconstruction from BRIEF and FREAK, the variants RA-FREAK and EX-FREAK lead to results which are almost identical with the original FREAK.

Keeping the patch size constant at $32 \times 32$ pixels, some results for various offsets between the patches can be seen in Fig. 5. An increased number of overlapping patches dramatically improves the quality of the reconstruction using FREAK. This can be understood easily by considering the peculiar shape of the reconstruction without overlap: each estimated patch contains the correct gradient information at its center only. By introducing more overlap between the patches these small parts of contour sum up to recreate the original objects.

Instead of computing patches at fixed positions, an experiment more relevant with respect to privacy concerns consists in reconstructing only the patches associated with an interest point detector. For the results shown in Fig. 6, we have first applied the FAST detector of OpenCV with its default parameters and discarded the remaining part of the image, hence the black areas, and used real-valued descriptors. Fig. 6 shows the results of the same experiment with binary descriptors. Since FAST points tend to aggregate near angular points and corners, this leads to a relatively dense reconstruction. In each of the images, the original content can be clearly recognized and a large part of the background clutter has been removed. Thus, one can add as a side note that FAST points are a good indicator of image content saliency. The results in Fig. 7 and Fig. 11 extend to binary descriptors an important privacy issue raised before in [4] for SIFT: if one can intercept keypoint data sent over a network, then it is possible to find out what the legitimate user was seeing. Finally, Fig. 12 compares our results with [4]. This earlier work is able to reconstruct globally the images thanks to Poisson interpolation but lacks the finer details found in our approach.

Reconstruction results from BRIEF and FREAK are strikingly different (Fig. 4). While BRIEF leads to large, blurred edge estimates that almost entirely occupy the original patch, FREAK produces small accurate edges almost confined in the center of the patch. This allows us to point at a fundamental difference between BRIEF and FREAK. While the former does randomly sample a rough estimate of the dominant gradient in the neighborhood, the latter concentrates its finest measurements and allows more bits (Fig. 9) to the innermost part. Thus, the inversion of BRIEF leads to a fuzzy blurred edge dividing two areas since the information is spread spatially over the whole patch, while the reconstruction of FREAK produces a small but accurate edge surrounded by a large low-resolution area. This is confirmed by the experiments shown in Fig. 8. One can especially remark the eyes of `Lena` and `Barbara` and the crossed pattern of the table blanket from `Barbara` which exhibit fine details using FREAK that are missing with BRIEF. In the `Kata` image, one can almost recognize the face of the characters with FREAK, while the fingers holding the



(a) BRIEF (b) FREAK

(c) RA-FREAK (d) EX-FREAK

Fig. 4: Reconstruction of Lena from binary LBDs using Alg. 2. There is no overlap between the patches used in the experiment, thus giving a blockwise aspect. We have overlaid some edges from the original image. In each case, the orientation selected for the output patch is consistent with the original main gradient direction. Note also the difference between BRIEF (random measurements spread over an image patch) and FREAK and its derivatives (fine measurements with higher density near the center of the patch): the former gives large blurred edges covering the whole patch, while the latter affect the dominant gradient direction to the central pixel, leaving the periphery almost untouched.

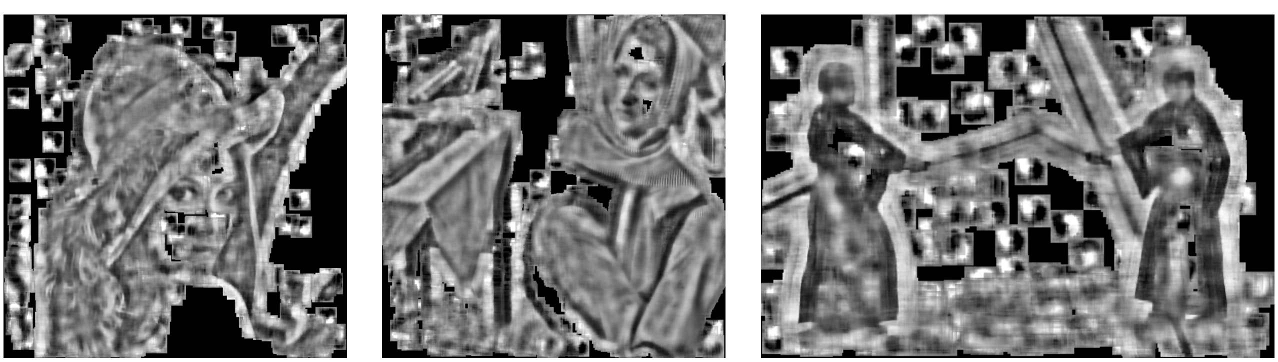


Fig. 6: Reconstruction of floating-point (non binarized) BRIEFs centered on FAST keypoints.

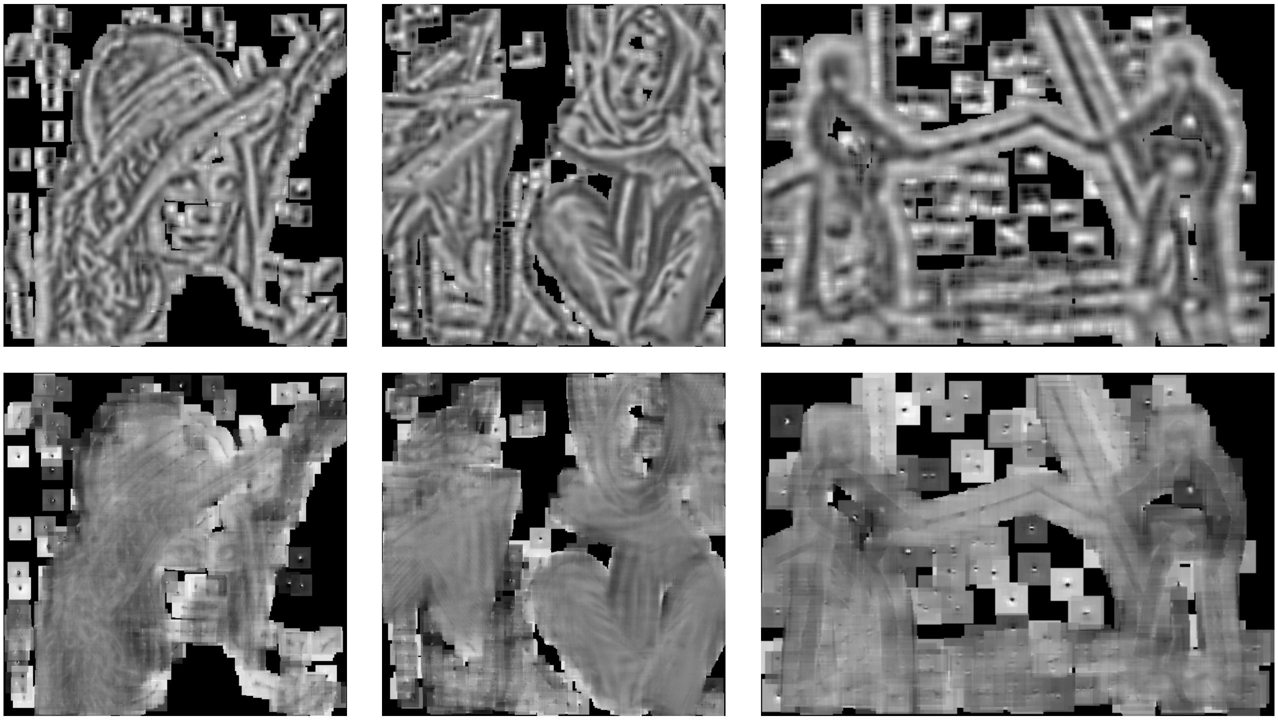


Fig. 7: Reconstruction of LBDs centered on FAST keypoints only. *Top row:* using BRIEF. *Bottom row:* using FREAK. Since the detected points are usually very clustered there is a dense overlap between patches, yielding a visually plausible reconstruction. The original image content has been correctly recovered by Alg. 2 from binary descriptors, and eavesdropping the communications of a mobile camera (e.g., embedded in a smartphone) could reveal private data.

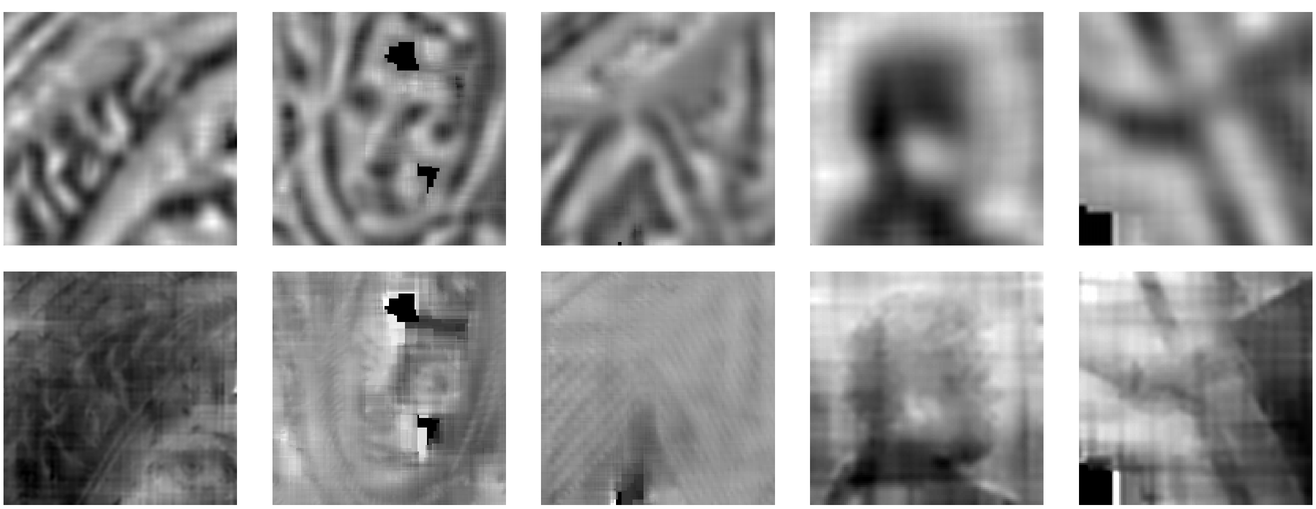
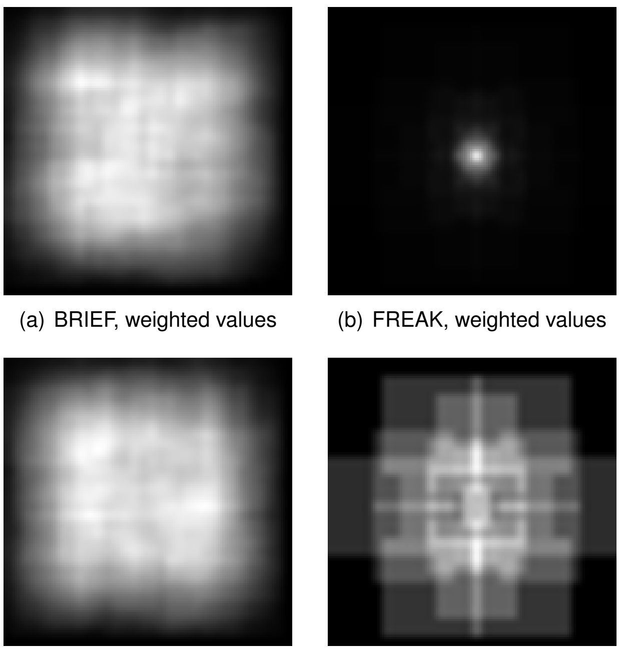


Fig. 8: Details of the reconstructions from Fig. 7. *Top row:* using BRIEF as LBD. *Bottom row:* using FREAK. The reconstructed patches were selected by the application of the FAST detector with identical parameters. While BRIEF is successful at capturing large gradient orientations, hence giving pleasant results when the image is seen from a distance, FREAK captures more accurate orientations in the center of the patches. Thus finer details are recovered: notice for example the eyes in the pictures of Lena and Barbara, the textures from Barbara or the face and the fingers in the kata image. For this figure, a linear remapping of the resulting images to the range $[0, 255]$ was applied to extend the contrast and emphasize the point.

sword are clearly distinguishable.

Fig. 9 compares the measurement strategies of BRIEF and FREAK for 512 measurements. The top row displays the sum of the absolute values of the weights applied to a pixel when computing the descriptor, *i.e.*, $\sum_{i=1}^{M} |(\mathcal{G}_{\boldsymbol{q}_i,\sigma_i})_j| + |(\mathcal{G}_{\boldsymbol{q}'_i,\sigma'_i})_j|$ in (1) for the $N$ pixels $1 \leq j \leq N$. We clearly see that BRIEF measures patch intensity almost uniformly over its domain, while FREAK focuses its patch observation on the patch domain center. Yet, when plotting in how many LBD measurements a pixel contributed (Fig. 9, bottom row) one can see that FREAK also uses peripheral pixels. Since both the weight and occurrence patterns are similar with BRIEF, it means that this LBD is democratic and gives all the pixels a similar importance.

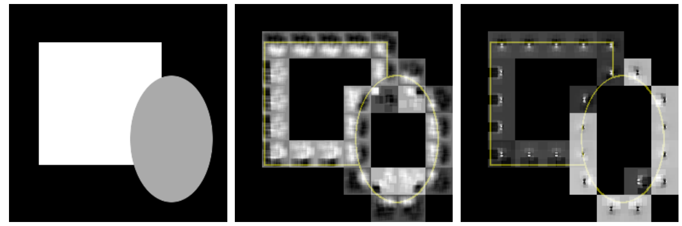


(a) BRIEF, weighted values (b) FREAK, weighted values (c) BRIEF, occurrence count (d) FREAK, occurrence count

Fig. 9: Comparison of the spatial weights in BRIEF and FREAK basis functions. In the top row, we display the sum of the absolute values of the weight of each pixel when computing a descriptor. Brighter means higher importance. One can see that BRIEF considers pixels almost equally all over the patch, while FREAK gives a very high weight to the center. The bottom row shows how many times a pixel value was read to generate the description vector. Here, brighter means often retained. This shows that FREAK uses peripheral values, but with a low ponderation. BRIEF is clearly more democratic since the weight pattern is similar to the occurrence pattern.

### 4.3 Quality and stability of the reconstruction

Because of the very peculiar structure of the LBD operators, establishing strong mathematical properties on these matrices is a very arduous task, especially in the 1-bit case. As a consequence, finding indubitable theoretical grounds to the success of our BIHT reconstruction algorithm still remains to be investigated. However, one can remark that the conditions used to ensure the existence of a reconstruction in Compressive Sensing, like the Restrictive Isometry Property (RIP) or the Binary $\epsilon$ Stable Embedding (B$\epsilon$SE) for 1-bit CS, are only *sufficient* conditions and are by no means necessary conditions. Since LBDs were designed to accurately describe some image content, they are probably more efficient than random sensing matrices. Hence, they can capture more information from an input patch with very few measurements and with a more brutal quantization at the cost of a loss in genericity: they are specialized sensors tuned to image keypoints. This follows for instance the approach of [34] where near-optimal *bandwise* random projections, in which more projections are allocated to low spatial frequencies, better capture natural images than common compressive sensing strategies.



(a) Original (b) BRIEF (c) FREAK

Fig. 10: Zero-overlap reconstruction of a synthetic image using LBDs of $M = 128$ measurements instead of 512 as in the other experiments (and 256 in most image matching softwares). Note that in spite of the huge information loss (compression ratio of $256:1$ for each patch) the directions of the edges are correctly estimated.

An important parameter with respect to the expected quality of the reconstructions is of course the length $M$ of the LBDs. Since we lack a reliable quality metric to assess the reconstructions, we have proceeded to visual comparisons between the original image contours and the reconstructed gradient directions on a synthetic image. As can be seen in Fig. 10, dominant orientations are reconstructed correctly until $M = 128$ measures. Smaller sizes yield blocky estimated patches where it is hard to infer any edge direction.

## 5 Conclusion and Future Work

In this work, we have presented two algorithms that can successfully reconstruct small image parts from a subset of local differences without requiring external data or prior training. Both algorithms leverage an inverse problem approach to tackle this task and use as regularization constraint the sparsity of the reconstructed image patches in some wavelet frame. However, they rely on different frameworks to solve the corresponding problem.

The first method relies on proximal calculus to minimize a convex non-smooth objective function, adopting a deconvolution-like approach. While this functional was


(a) Original image (b) FAST + binarized BRIEF (c) FAST + binarized FREAK

Fig. 11: Reconstruction of book covers. The leftmost image is the input image. The middle and right images are close-up views of the content of the red rectangle after its reconstruction from BRIEF and FREAK descriptors respectively. Patches of interest were selected using the FAST detector. This experiment confirms the difference between BRIEF and FREAK: while the former extracts salient shapes such as the butterfly, the latter is more successful at reconstructing the text. Note that FREAK allows the reading of the titles, hence demonstrating the potential privacy breach in case of communications eavesdropping.

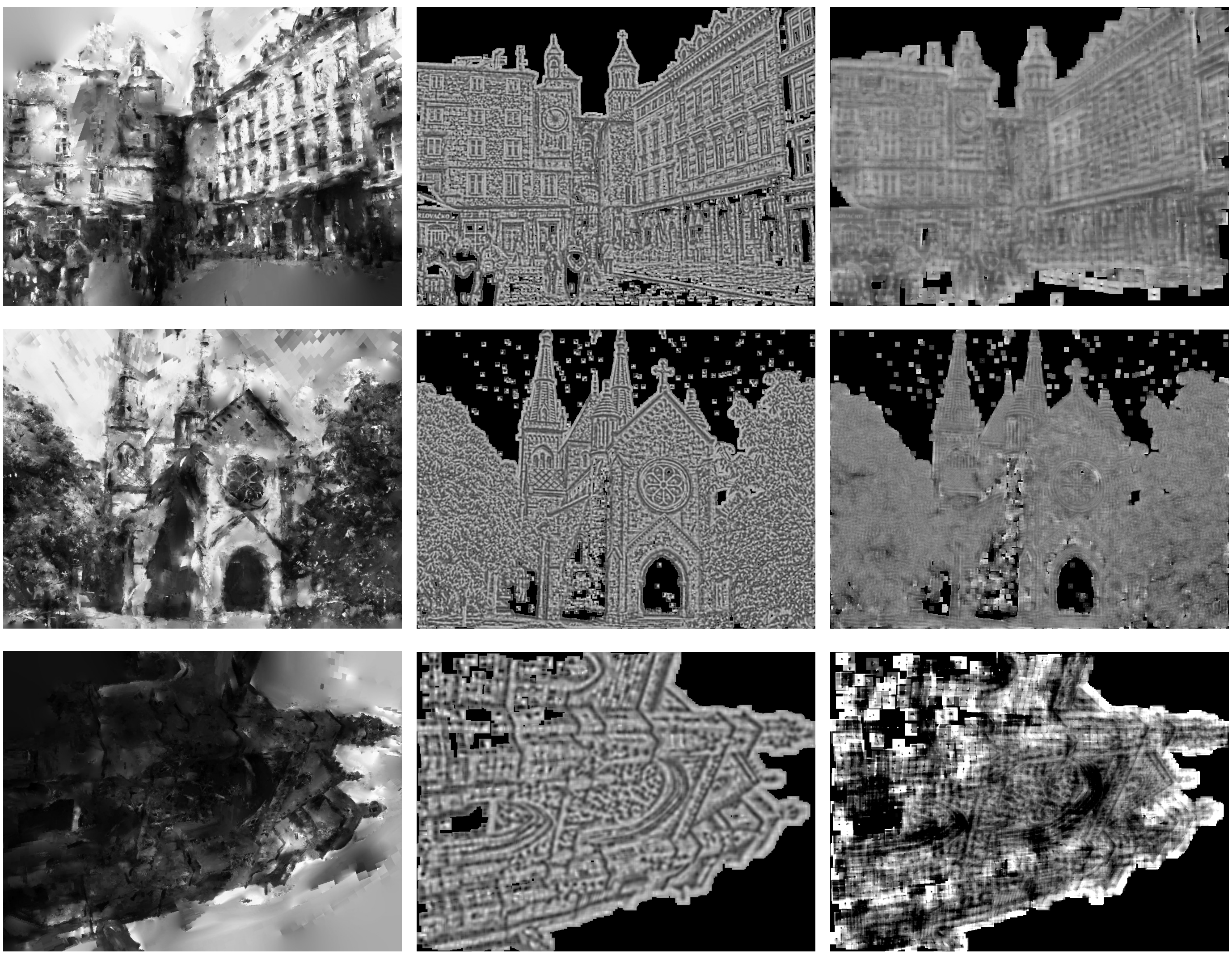


Fig. 12: Comparison with [4]. From left to right: image reconstructed with the method of [4], our binary reconstruction algorithm using FAST+BRIEF (middle) and FAST+FREAK (right). We used patches of $32 \times 32$ pixels in our algorithm. In this figure, the results of our algorithm were linearly remapped to the maximum dynamic range $[0, 255]$ for readability. This figure is best viewed online or in electronic version.

not specifically designed for 1-bit LBDs, it has proved to be robust enough to provide some 1-bit reconstructions, but it does lack stability in this case. On the other hand, the second method was built from the ground up to handle 1-bit LBDs, and thus provides stable results. The reconstruction process is guided by a hard sparsity constraint in the wavelet domain.

There are several levels on which to exploit and interpret our results. First, they can have an important industrial impact. Since it is possible to easily invert LBDs without additional information, mobile application developers cannot simply move from SIFT to LBDs in order to avoid the privacy issues raised by [4]. Hence, they need to add an additional encryption tier to their feature point transmission process if the conveyed data is either sensitive or private.

Second, the differences in the reconstruction from different LBDs can help researchers to design their own LBDs. For example, our experiments have pointed out that BRIEF encodes information at a coarser scale than SIFT, and maybe both descriptors could be combined in some way to create a scale-aware descriptor taking advantage of both patterns. Hence, our work can be used as a tool to study and compare binary descriptors providing different information than standard matching benchmarks. Furthermore, the fact that real-value differences yield comparable results as binarized descriptors legitimates *a posteriori* the performance of LBDs in matching benchmarks: they encode most of the originally available information.

Finally, our framework for 1-bit contour reconstruction could be combined with the concept of Gradient Camera [32], leading to the development of a 1-bit Compressive Sensing Gradient Camera. This disruptive device would ally the qualities of both worlds with an extended dynamic range and low power consumption. Exploiting the retinal pattern of FREAK and our reconstruction framework could also yield neuromorphic cameras mimicking the human visual system that could be useful for medical and physiological studies.

Of course, the reconstruction algorithms still need to be improved before reaching this application level. Among the possible improvements, we believe that an interesting extension would be to depart from the patch-based reconstruction analyzed in this paper and to inverse a larger problem where the whole image, assumed sparse or compressible in a certain basis, would have to be reconstructed from its sensing from a collection of LBDs observed on different patches. Another possibility is to make our framework *scale-sensitive*. While some feature point detectors provide a scale space location of the detected feature, we discarded the scale coordinate and used patches of fixed width instead. This does not depreciate our experiments with FAST points because we used an implementation that is not scale aware, but reconstructions of better quality can probably be achieved by mixing smooth coarse scale patches with finer details. Additionally, this work did not investigate the issues linked to the geometric transformation invariance enabled by most descriptors. Our model can be interpreted in terms of reconstruction of canonical image patches that correspond to a reference orientation and scale. As far as we have seen, this omission did not create artifacts in our results. This absence by itself is worth of investigating.

## Acknowledgment

The authors would like to thank the reviewers for their thorough reading of this manuscript, and also Jérôme Gilles (UCLA) who contributed to the implementation of the wavelet transforms. The `Kata` image comes from the iCoseg dataset[3] [33]. The results obtained with the algorithm [4] were kindly provided by its authors. Corresponding original images are courtesy of INRIA through the INRIA Copydays dataset. Laurent Jacques is funded by the Belgian Fund for Scientific Research - F.R.S-FNRS. Alexandre Alahi is funded by the Swiss National Science Foundation under the fellowship PBELP2_141078.

## References


[1] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[2] “Googles Goggles.” http://www.google.com/mobile/goggles

[3] “Kooaba Image Recognition.” http://www.kooaba.com

[4] P. Weinzaepfel, H. Jegou, and P. Pérez, “Reconstructing an image from its local descriptors,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 337–344.

[5] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] H. Bay, T. Tuytelaars, and L. van Gool, “Surf: Speeded up robust features,” *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[7] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH ’03. New York, NY, USA: ACM, 2003, pp. 313–318.

[8] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, “HOGgles: Visualizing Object Detection Features,” *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2013.

[9] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, 2005.

[10] P. T. Boufounos and R. G. Baraniuk, “1-Bit compressive sensing,” in *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pp. 16–21, 2008.

[11] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *Information Theory, IEEE Transactions on*, vol. 59, no. 4, pp. 2082–2102, 2013.

[12] E. d’Angelo, A. Alahi, and P. Vandergheynst, “Beyond Bits: Reconstructing Images from Local Binary Descriptors,” *International Conference On Pattern Recognition (ICPR)*, pp. 1–4, Sep. 2012.

[13] P. L. Combettes and J. C. Pesquet, “Proximal splitting methods in signal processing,” *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212, 2011.

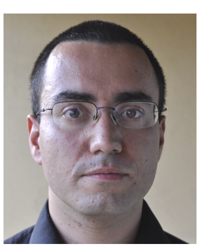
[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features,” *Computer Vision–ECCV 2010*, pp. 778–792, 2010.

[15] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast Retina Keypoint,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 510–517.

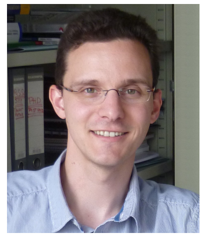
[16] C. Harris and M. Stephens, “A combined corner and edge detector,” *Alvey vision conference*, vol. 15, p. 50, 1988.


3. Link: http://chenlab.ece.cornell.edu/projects/touch-coseg/

[17] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” *Computer Vision–ECCV 2006*, pp. 430–443, 2006.
[18] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors,” *Information Theory, IEEE Transactions on*, vol. 59, no. 4, pp. 2082-2102, 2013.
[19] T. Ahonen, A. Hadid, and M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2006.
[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2564–2571.
[21] S. Leutenegger, M. Chli, and R. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2548–2555.
[22] E. Sidky, J. Jørgensen, H. Jakob and X. Pan, “Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm,” *Physics in Medicine and Biology*, vol. 57, no. 10, pp. 3065, 2012.
[23] A. Chambolle and T. Pock, “A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2010.
[24] H. Raguet, J. Fadili, and G. Peyré, “Generalized forward-backward splitting,”, *SIAM Journal on Imaging Sciences*, 2013.
[25] A. Gonzalez, L. Jacques, C. De Vleeschouwer, and P. Antoine, “Compressive Optical Deflectometric Tomography: A Constrained Total-Variation Minimization Approach,” *Submitted, preprint, arXiv:1209.0654*, 2012.
[26] C. Michelot, “A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$,” *Journal of Optimization Theory and Applications*, vol. 50, no. 1, pp. 195–200, 1986.
[27] E. Candès and T. Tao, “Decoding by linear programming,” *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.
[28] Y. Plan and R. Vershynin, “One-bit compressed sensing by linear programming,” *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1275–1297, 2013.
[29] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1 pp. 267–288, 1996.
[30] T. Blumensath and M. Davies, “Iterative hard thresholding for compressed sensing,” *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.
[31] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
[32] J. Tumblin, A. Agrawal, and R. Raskar, “Why I want a gradient camera,” *Computer Vision and Pattern Recognition, CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 103–110, 2005.
[33] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, “Interactively Co-segmentating Topically Related Images with Intelligent Scribble Guidance,” *International Journal of Computer Vision*, vol. 93, no. 3, pp. 273–292, 2011.
[34] H. S. Chang, Y. Weiss and W. T. Freeman, “Informative sensing of natural images,” *Image Processing (ICIP), IEEE International Conference on*, pp. 3025–3028, 2009.

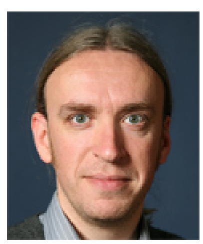


**Emmanuel d’Angelo** Emmanuel d’Angelo received the M.Sc.(Eng) in Information Processing and Remote Sensing from the ISAE (Supaero), Toulouse, France, and the M.Sc. degree in Applied Mathematics for Machine Learning and Vision from ENS, Cachan, France, in 2003 and 2005 respectively. He worked for several years at the French ministry of defense on the development of image processing systems for military and homeland security applications. In 2008, he joined the Signal Processing Laboratory (LTS2) of the Swiss federal Institute of Technology (EPFL), Lausanne, Switzerland, where he was awarded the Ph. D. degree in 2013. His research interests focus on patch-based and non-local methods in Image Processing, and parallel implementations.



**Laurent Jacques** Laurent Jacques received the B.Sc. in Physics, the M.Sc. in Mathematical Physics and the PhD in Mathematical Physics from the Université Catholique de Louvain (UCL), Belgium. Postdoctoral Researcher with the Communications and Remote Sensing Laboratory of UCL in 2005 to 2006, he obtained in Oct. 2006 a four-year (3+1) Postdoctoral funding from the Belgian FRS-FNRS in the same lab. He was a visiting Postdoctoral Researcher, in spring 2007, at Rice University (DSP/ECE, Houston, TX, USA), and from Sep. 2007 to Jul. 2009, at the Swiss Federal Institute of Technology (LTS2/EPFL, Switzerland). Formerly funded by Belgian Science Policy (Return Grant, BELSPO, 2010-2011), and as a F.R.S.-FNRS Scientific Research Worker (2011-2012) in the ICTEAM institute of UCL, he is a FNRS Research Associate since Oct. 2012. His research focuses on Sparse Representations of signals (1-D, 2-D, sphere), Compressed Sensing theory (reconstruction, quantization) and applications, Inverse Problems in general, and Computer Vision.

**Alexandre Alahi** Alexandre Alahi received the M.S.degree in Electrical and Computer engineering and Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne (Swiss Federal Institute of Technology). During his studies, he earned a one-year exchange fellowship to study at Carnegie Mellon University, Pittsburgh, PA, USA. In the past few years, he has worked with various companies in the field of image processing and computer vision. He is now a Postdoctoral researcher with the vision laboratory at Stanford University. His research focuses on sparse approximation, compressed-sensing, inverse problems, real-time vision, large-scale analysis, and bio-inspired vision. He holds several awards (e.g. CVPR 2012 open source award) and patents.



**Pierre Vandergheynst** Pierre Vandergheynst received the M.S. degree in Physics and the Ph.D. degree in Mathematical Physics from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 1995 and 1998, respectively. From 1998 to 2001, he was a Postdoctoral Researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He was Assistant Professor at EPFL (2002-2007), where he is now an Associate Professor. His research focuses on harmonic analysis, sparse approximations and mathematical data processing in general with applications covering signal, image and high dimensional data processing, sensor networks, computer vision. He was co-Editor-in-Chief of Signal Processing (2002-2006) and is Associate Editor of the IEEE Transactions on Signal Processing (2007-2011), the flagship journal of the signal processing community. He has been on the Technical Committee of various conferences, serves on the steering committee of the SPARS workshop and was co-General Chairman of the EUSIPCO 2008 conference. Pierre Vandergheynst is the author or co- author of more than 60 journal papers, one monograph and several book chapters. He has received two IEEE best paper awards. Professor Vandergheynst is a laureate of the Apple 2007 ARTS award and of the 2009-2010 De Boelpaepe prize of the Royal Academy of Sciences of Belgium.