IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 21, 2021, accepted July 7, 2021, date of publication July 12, 2021, date of current version July 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3096643

Representation Learning via Cauchy Convolutional Sparse Coding

PERLA MAYO, (Student Member, IEEE), OKTAY KARAKUŞ, (Member, IEEE), ROBIN HOLMES, AND ALIN ACHIM, (Senior Member, IEEE)

Visual Information Laboratory, University of Bristol, Bristol BS1 5DD, U.K.

Corresponding author: Perla Mayo (pm15334@bristol.ac.uk)

This work was supported in part by a Consejo Nacional de Ciencia y Tecnología (CONACyT) PhD studentship under grant 461322 (to Mayo), in part by the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/R009260/1 (AssenSAR), and in part by a Leverhulme Trust Research Fellowship (to Achim).

ABSTRACT In representation learning, Convolutional Sparse Coding (CSC) enables unsupervised learning of features by jointly optimising both an ℓ_2-norm fidelity term and a sparsity enforcing penalty. This work investigates using a regularisation term derived from an assumed Cauchy prior for the coefficients of the feature maps of a CSC generative model. The sparsity penalty term resulting from this prior is solved via its proximal operator, which is then applied iteratively, element-wise, on the coefficients of the feature maps to optimise the CSC cost function. The performance of the proposed Iterative Cauchy Thresholding (ICT) algorithm in reconstructing natural images is compared against algorithms based on minimising standard penalty functions via soft and hard thresholding as well as against the Iterative Log-Thresholding (ILT) method. ICT outperforms the Iterative Hard Thresholding (IHT), Iterative Soft Thresholding (IST), and ILT algorithms in most of our reconstruction experiments across various datasets, with an average Peak Signal to Noise Ratio (PSNR) of up to 11.30 dB, 7.04 dB, and 7.74 dB over IST, IHT, and ILT respectively. The source code for the implementation of the proposed approach is publicly available at `https://github.com/p-mayo/cauchycsc`

INDEX TERMS Cauchy-based penalty function, convolutional sparse coding, proximal splitting.

I. INTRODUCTION

Representation learning seeks to understand the underlying patterns and structures that give raise to the data of interest. This often involves using generative models to describe the processes involved in the formation of this data, using known or assumed priors [1]. In some of these models it is assumed that data arises from a linear operation among the elements of a set of basic or canonical features. Thus, in addition to selecting this set of features, it is also necessary to obtain their respective coefficients for generating any given sample. Computing the coefficients for such a sample effectively transforms it into the chosen feature domain. This transformation process is referred as encoding, whilst decoding corresponds to the reverse action of transforming back into the original domain [1], [2].

Establishing an effective choice of features for the generative model can aid in understanding the nature of the data. Furthermore, the resulting coefficients can be used in-place of the raw data for many tasks including source separation [3], [4], image compression [5]–[7], image denoising [8], [9], image super-resolution [10], [11], image and audio [12], [13] [14] classification, or anomaly detection [15]. Algorithms following these approaches have been successfully employed in a variety of applications such as medical imaging [11], [16], [17] and remote sensing [3], [18]–[20], to name but a few.

Thus, determining a suitable feature set for the data in question is a crucial task, but how should this be accomplished? Early on, it was common to employ a set of predefined or fixed basis features. Sets such as Wavelets and ones obtained from the Discrete Cosine Transform (DCT) have been used with success for image denoising and compression respectively. For instance, to denoise images, a common practice is to transform them to the wavelet domain, in which a threshold can be applied to the wavelet coefficients [9]. Reversing the coefficients to the original domain after thresholding results in a cleaner image. Feature sets such as these are in

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang.

some sense universal. As such they are not always effective in capturing specific traits of a particular dataset. This can sometimes result in vital particularities of the data being lost. For this reason different approaches enabling the unveiling of new meaningful data specific information have become more prevalent.

As previously mentioned, the assumptions made will guide the design of the model to utilise for these purposes. Principal Component Analysis (PCA) [21] and Independent Component Analysis (ICA) [22] provide the means to determine the underlying components comprising the data of interest. The difference between these two being that ICA makes a further assumption regarding the independence among these components. In either case, the generative model corresponds to a dot product that meets some orthogonality conditions on the (squared) matrix of features. There is no strict requirement however, for the feature matrix to be square. For instance, in AutoEncoders (AE) [23] the goal is to train a network in an unsupervised fashion such that its weights both encode and decode the data in a lower dimensional space. Alternatively, there is dictionary learning and sparse coding [24], in which it has been suggested that overcomplete sets are capable of describing as well as (if not better than) complete ones as they are able to unveil a bigger number of underlying features [24]. Since the features belong to an overcomplete matrix, the model to solve is underdetermined and an infinity of solutions becomes available. This can be remedied by assuming the data representation is sparse, meaning that only a few elements of the feature matrix take part in the formation of the data and hence most of the elements in the vector of coefficients are set to zero. This is modelled by the addition of a penalty term known to enforce this behaviour. The assumption of sparsity has been motivated by the way in which the V1 cells from the visual cortex work [25].

With this addition, the model not only learns the features to represent the data but also the coefficients describing their contribution. This is commonly achieved by splitting the learning into two tasks, i.e., a step is devoted to learn the elements of the dictionary and the other to learn the elements in the vector of coefficients. There is evidence showing that the latter step can be the most critical for the model to succeed in the representation task [2]. In [2] the power of encoding was demonstrated regardless the choice of learning (or lack of it) for the features. This motivates efforts on the design of novel approaches aiming to learn the coefficients in the encoding stage of the algorithm.

The core contribution of this work is the derivation and effective demonstration of a new regularisation term used during the encoding step of Convolutional Sparse Coding (CSC). This term arises from the assumption that feature maps of coefficients follow a Cauchy distribution. To make use of this new regularisation we propose the Cauchy proximal operator, which when implemented iteratively follows in the vein of shrinkage algorithms [26]–[28] and gives raise to an algorithm, which we refer to as Iterative Cauchy Thresholding (ICT). Unlike existing previous approaches this algorithm does not perform explicit thresholding, resulting in values approaching 0 but not necessarily locking to it. Following the theoretical guarantees for the Cauchy proximal operator provided in [29], in this manuscript we devoted efforts to demonstrating the efficiency of this new type of regularisation in a 2D image reconstruction task via CSC. We evaluate the performance of the proposed approach against the common choices of soft, hard, and log thresholding algorithms.

The remaining of this manuscript is organised as follows. The backbone and derivation of the proposed algorithm is reviewed in detail in section III along with related work that inspired our approach. In section IV the algorithm used for the reconstruction task is shown. The experiments conducted are found in section V along with their results. Lastly, section VI offers a discussion, conclusion, and future lines of work.

II. THEORETICAL PRELIMINARIES

In a basic generative model, it is assumed the observation vector $y \in \mathbb{R}^P$ can be estimated from a linear combination of the column vectors (also referred as atoms, codes or features) of the dictionary matrix $A = [a_1, a_2, \ldots, a_N] \in \mathbb{R}^{P\times N}$. The contribution of each one of these elements is given by the coefficients in the vector $x \in \mathbb{R}^N$ such that there is one coefficient for each column in A. The estimation of the observed data y is thus given by:

$$\hat{y} = Ax. \tag{1}$$

where $\hat{y} \in \mathbb{R}^P$. Since $\hat{y}$ does not retrieve exactly the original sample y, this is $y \approx \hat{y}$, there exists a vector $\boldsymbol{\varepsilon}$ such that $\boldsymbol{\varepsilon} = y - \hat{y}$ or, equivalently

$$y = \hat{y} + \boldsymbol{\varepsilon}, \tag{2}$$

with $\boldsymbol{\varepsilon} \in \mathbb{R}^P$. For dictionary learning and sparse coding, the dictionary matrix A is overcomplete, i.e. $N > P$. In addition, there is a one-to-one spatial correspondence between the features and the data to reconstruct, in other words, the size of the features has to be the same as that of the data, which can be impractical for large size signals. This can be alleviated by using patches extracted from the original signal instead, reducing the size of the dictionary atoms to the one of these patches. Thus, the observed data is now a set of vectors $y_i \in \mathbb{R}^M, i = 1, 2, \ldots, I$ containing the I patches extracted from the original signal of original dimension P. Moreover, from the N available features, only a much smaller number L are used for the generation of the data ($L \ll N$). By using patches it becomes necessary to perform pre- and post-processing of the data to extract the patches and then bring them together to reconstruct the sample. For this to work, it is assumed these patches are independent, which is not necessarily true. There are two main ways to extract patches from the data, one of them is restricting them to not overlap. This, in addition to the independence assumption, is later on reflected in blocking artifacts when the samples are reconstructed. On the other hand, when the patches are overlapped, an average operation is performed when building the final image, which also results in a degraded version of the original sample as there is now a smoothing effect present in it. Furthermore, the learned features are often translated versions of other atoms within the set (they are not shift invariant).

The use of the convolution operator in the generative model helps to address the aforementioned limitations of dictionary learning [14], [30]. Thus, it evolves to CSC. Eq. 3 describes this model.

$$\hat{y} = \sum_{k=1}^{K} f_k * z_k, \tag{3}$$

where the signal of interest $y \in \mathbb{R}^P$ is now modelled as a sum of K filters $f_k \in \mathbb{R}^M$ convolved with their respective feature map $z_k \in \mathbb{R}^Q$, with $P = M + Q - 1$ for $k = 1, 2, \ldots, K$. Note that y is the complete original signal of size P. We would like to highlight that for ease of reading the equations are expressed purely using one dimensional data. Nevertheless, the extension to higher dimensional data is straightforward.

In either of the two mentioned generative models, the learning of the features can be done by minimising the error between the estimated and the observed data. For instance, for CSC:

$$\begin{aligned} f^* &= \arg\min_f \mathcal{L}(f, z), \\ &= \arg\min_f ||y - \hat{y}||_2^2, \\ &= \arg\min_f ||y - \sum_{k=1}^{K} f_k * z_k||_2^2, \end{aligned} \tag{4}$$

where $f = [f_1, f_2, \ldots, f_K]$ and $z = [z_1, z_2, \ldots, z_K]$ in $\mathcal{L}(\cdot, \cdot)$. In the traditional dictionary learning approach, the optimisation would be carried over the matrix A instead of f. From now on, the generative model considered in the paper is the CSC.

To seek for sparsity, it suffices to add a regularisation term to the optimisation function as

$$\begin{aligned} f^*, z^* &= \arg\min_{f,z} \mathcal{G}(f, z), \\ &= \arg\min_{f,z} \mathcal{L}(f, z) + \lambda\varphi(z), \\ &\quad \text{s.t. } ||f_k||_2 = 1, \quad k = 1, 2, \ldots, K, \end{aligned} \tag{5}$$

in which it is now required to learn, in addition to the set of features in f, the set of coefficients in the feature maps in z. The constraint on the filters prevents them from absorbing most of the energy during the learning.

The learning is carried on by iteratively alternating the optimisation of the cost function over f and z. This means that in a first step (z-step), the cost function will be minimised by assuming f is fixed or constant. The opposite happens during the f-step.

In addition to Gradient Descent, several approaches have emerged aiming to solve Eq. 5 w.r.t. f to learn features from the data, such as K-SVD [31] and the more image statistic-adapted SparseDT [32]. Similarly, optimising the cost function w.r.t. z will result in the learning of the sparse coefficients. Such optimisation depends on the choice of penalty function. If one is to seek for the sparsest solution, then the penalty term chosen is the ℓ_0-norm. Hence, finding the set of coefficients that optimise Eq. 5 is a combinatorial (NP-hard) problem. Broadly speaking, there are two main approaches to solve said regularisation term: greedy and relaxed algorithms. The first category focuses on solving the ℓ_0-norm whilst the second one considers its relaxed version (the ℓ_1-norm). For the former, Matching Pursuit is one of the most common solvers in which coefficients are chosen one by one in a greedy fashion until a stopping criteria is met. If, on the other hand, one chooses the Least Absolute Shrinkage and Selection Operator (LASSO), given by the ℓ_1-norm, the function to optimise is now non-smooth convex.

In these circumstances, the choice of penalty term is based on the known behaviour (shape) of the function. Thus, as long as one knows the function has a shape that can enforce sparsity, such function can be used for $\varphi(\cdot)$. Some alternatives are the non-convex ℓ_p-norm (with $p \leq 1$) or the (also non-convex) log regulariser among others. A comprehensive review of these sparsity-enforcing functions can be found in [33] and references therein.

It is important to note at this point that sparsity can be perceived from (at least) two perspectives: in one sense, the aim is to set most elements to zero as enforced by penalties based on the ℓ_1- and ℓ_0-norms. The alternative option is to enforce sparsity from a statistical point of view, whereby the coefficients follow a heavy-tailed distribution. The Cauchy distribution underpins ICT, whereas with a similar reasoning one could also say that Iterative Shrinkage Thresholding (IST) is based on the Laplace distribution.

Iterative algorithms have come along with an associated sparsity enforcing penalty term. The IST algorithm is a common choice when the function to optimise makes use of the ℓ_1-norm whilst the Iterative Hard Thresholding (IHT) algorithm solves the ℓ_0-norm. More recently the use of the Iterative Log Thresholding (ILT) [28] has been proposed to optimise the log regulariser. Table 1 summarises the equations involved in these algorithms.

These algorithms can also be derived via surrogate functions in which one seeks to separate the terms involved in the cost function. Regardless of the chosen algorithm to use, these thresholding operators are applied in an element-wise fashion. These three approaches suppress any value below some threshold but it is only IST and ILT that update values higher than said threshold. In the case of IST this has a direct impact on the results as they often exhibit blurring.

III. ITERATIVE CAUCHY THRESHOLDING

The Cauchy assumption in the field of image processing is not new as it has previously been used to model the noise corrupting the images of interest [34], [35]. Nonetheless, in this work it is not the noise but the coefficient values involved in the generative models the ones that are assumed to follow this distribution. In fact, the assumption of a Cauchy prior for the model has been done with success in the past [36]–[40].

TABLE 1. **Penalty terms that promote sparsity and their proximal operators.**

Algorithm	Penalty term	Optimising function
IHT	$\lvert x_i \rvert^0$	$x_i = \begin{cases} x_i, & \lvert x_i \rvert > \lambda \\ 0, & \lvert x_i \rvert \leq \lambda \end{cases}$
IST	$\lvert x_i \rvert$	$x_i = \begin{cases} x_i - \lambda/2, & x_i > \lambda/2 \\ x_i + \lambda/2, & x_i < -\lambda/2 \\ 0, & otherwise \end{cases}$
ILT	$\lambda \log(\delta + x_i)$	$x_i = \begin{cases} \frac{1}{2}\left((x_i - \delta) + \sqrt{(x_i + \delta)^2 - 2\lambda)}\right), & x_i \geq \sqrt{2\lambda} - \delta \\ \frac{1}{2}\left((x_i + \delta) - \sqrt{(x_i - \delta)^2 - 2\lambda)}\right), & x_i < -\sqrt{2\lambda} + \delta \\ 0, & otherwise \end{cases}$

The encoding step depends on the regularisation term in the optimisation model. This term could fall into the non-smooth convex functions, such as the ℓ_1-norm; non-smooth non-convex, such as the log regulariser or the ℓ_0-pseudo norm; or smooth non-convex penalty terms, such as the one explored in this paper. This function is derived from a statistical assumption on the coefficients, serving as prior in a Maximum a Posteriori (MAP) approach. The resultant learning algorithm corresponds to a function which, despite the non-convexity of its regularisation term, is guaranteed to convergence under a certain condition. Specifically, it is the Cauchy distribution the one assumed to drive the learning framework. The use of this prior enables the learning of the coefficients by iteratively applying its proximal operator on the coefficients, achieving shrinkage around a non-explicit threshold that emerges naturally from the equations involved in this process. In addition, the parameters shaping the distribution of the coefficients can be estimated from the observations, facilitating the use of this method.

A. THE CAUCHY DISTRIBUTION

In this work, we make use of a penalty function based on the Cauchy distribution, which is known to be heavy-tailed, hence ideal to model sparsity [14]. From a purely theoretical viewpoint, our preference for the Cauchy model over other candidate models stems from its membership of the α-Stable family of distributions. Specifically, unlike other empirical distributions able to faithfully fit distributions with heavy-tails, α-Stable distributions are motivated by the generalised Central Limit Theorem (CLT) similarly to the way Gaussian distributions are motivated by the classical CLT. However, although the (symmetric) α-Stable density behaves approximately like a Gaussian density near the origin, its tails decay at a lower rate than the Gaussian density tails. Indeed, let X be a non-Gaussian symmetric α-Stable random variable. Then, as $x \to \infty$

$$P(X > x) \sim c_\alpha x^{-\alpha}, \tag{6}$$

where $c_\alpha = \Gamma(\alpha)(\sin \frac{\pi\alpha}{2})/\pi$, $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is the Gamma function, and the statement $h(x) \sim g(x)$ as $x \to \infty$ means that $\lim_{x\to\infty} h(x)/g(x) = 1$. Hence, the tail probabilities are asymptotically power laws.

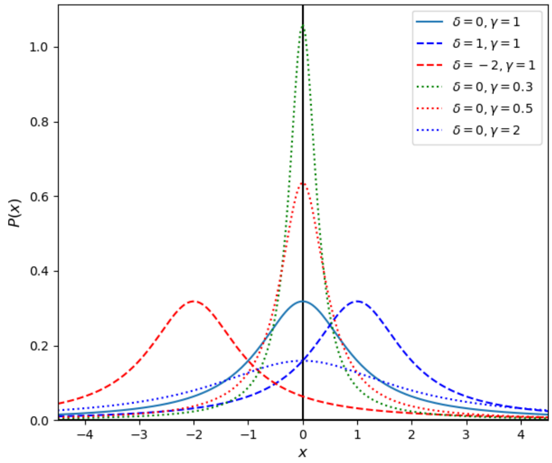


FIGURE 1. **Cauchy p.d.f. with different values for the parameters δ and γ.**

The location and dispersion of the Cauchy distribution are described by the parameters δ and γ respectively, and its p.d.f. is defined by

$$p(x) = \frac{\gamma}{\pi(\gamma^2 + (x - \delta)^2)}, \tag{7}$$

whilst Figure 1 illustrates their role on the distribution.

For the aim of the proposed work (enforcing sparsity), it is required that $\delta = 0$, so that the distribution is peaked at the origin. This results in a simplification of the expression to work with. On the other hand, the parameters of the distribution can be estimated from the data itself using maximum likelihood estimation as

$$\gamma = \arg\min_{\gamma} - \sum_{t=1}^{T} \log(p(x) + \epsilon), \tag{8}$$

where ϵ is a very small value.

Thus, the overall cost function to be optimised is composed of two functions, as illustrated in Eq. 5. The data fidelity term $\mathcal{L}(\cdot, \cdot)$ being convex with $\varphi(\cdot)$ possibly non-smooth or non-convex or both. The properties of the penalty term are determined by the chosen function, for instance, the ℓ_1-norm is non-smooth convex, the log penalty term is non-smooth and non-convex, and the Cauchy penalty term is smooth non-convex.

The optimisation of Eq. 5 is done by following the proximal splitting approach, in which the functions that present challenges during conventional optimisation techniques are projected into a convex set via their proximal operators. Note that the number of functions involved in the optimisation can be ≥ 2. The proximal operator of a given function $\varphi(\cdot)$ can be obtained by solving:

$$\mathrm{prox}_{\lambda\varphi}(x) = \arg\min_{z} (z-x)^2 + \lambda\varphi(z) \tag{9}$$

B. THE CAUCHY PROXIMAL OPERATOR

Following a MAP approach, the penalty term on the coefficients in z is then defined as $\varphi(\cdot) = -\log(p(\cdot))$, with $p(\cdot)$ as defined in Eq. 7 and setting $\delta = 0$. This penalty term is applied individually on every element of z. Thus, by using $\varphi(z) = -log(p(z))$ in Eq. 9 it is possible to derive the Cauchy proximal operator as

$$\begin{aligned}\mathrm{prox}_{\lambda\varphi}(x) &= \arg\min_{z} \quad (z-x)^2 + \lambda\varphi(z)\\ &= \arg\min_{z} \; (z-x)^2 - \lambda \log\left(\frac{\gamma}{\pi\left(\gamma^2+z^2\right)}\right).\end{aligned} \tag{10}$$

To find the stationary points, we just need to take the derivative of the above expression and setting it equals to zero. After doing this and rearranging the terms, we obtain:

$$z^3 - xz^2 + (\gamma^2+\lambda)z - \gamma^2 x = 0. \tag{11}$$

Using the Cardano's method to find the roots of the previous cubic equation with $a = 1$, $b = -x$, $c = \gamma^2 + \lambda$ and $d = -\gamma^2\, x$, one finally gets to:

$$z = \frac{x}{3} + t, \tag{12}$$

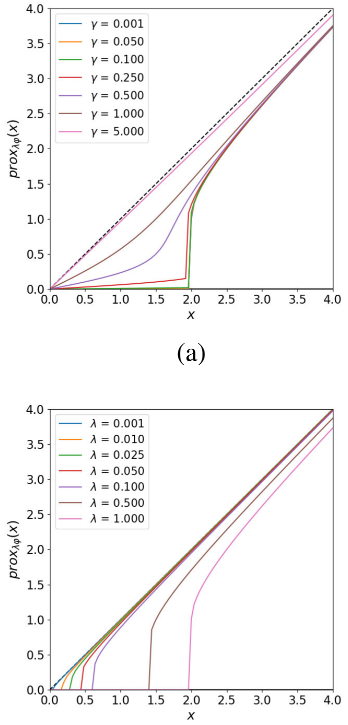
where

$$\begin{aligned} t &= \sqrt[3]{-\frac{q}{2} + \sqrt[2]{\Delta}} + \sqrt[3]{-\frac{q}{2} - \sqrt[2]{\Delta}},\\ \Delta &= \frac{q^2}{4} + \frac{p^3}{27},\\ p &= \lambda + \gamma^2 - \frac{x^2}{3},\\ q &= -\frac{2}{27}x^3 + \frac{1}{3}\left(\lambda - 2\gamma^2\right)x. \end{aligned}$$

The Cauchy proximal operator, thus, requires the selection values for the parameters γ and λ, for which it becomes ideal to understand their function in the operator. By fixing γ to a specific value and vary λ and vice-versa it is possible to gain an intuition of their roles. In fact, it is found that the value of γ shapes the thresholding function and smaller values contribute to a more aggressive shrinkage near the threshold, whilst λ shifts the threshold location. Fig. 2a and 2b illustrate this behaviour.

In fact, when $\gamma \to 0$ the threshold $\to 2\lambda$ and the shape of the function approximates the ILT. On the other hand, when $\lambda \to 0$, the roots of Eq. 11 $\to \gamma i, -\gamma i$, and x, which would keep the values unchanged, i.e. no shrinkage would be performed. Nonetheless, in this work we do not treat γ as a tunable parameter. Instead, this value is estimated from the data following the approach mentioned in Section III-A.

One could compare the Cauchy and log penalty terms (third row in Table 1) since both are shaped by the logarithm function and some parameter (δ for ILT and γ for ICT). However, the corresponding proximal operators are considerably different. A major difference between ICT and the rest of the algorithms presented in this manuscript so far is the lack of an explicit threshold. The coefficients are still shrunk according to the Cauchy proximal operator in an iterative manner, reaching values closer to zero but not necessarily locking on it. This occurs since, as we can see in Fig. 2a, there are curves that do not set any value to zero at all. The actual shape of the function describing the proximal operator will depend on the data itself.



(a)

(b)

FIGURE 2. Behaviour of ICT for varying (a) γ's and (b) λ's.

The penalty term derived from the Cauchy distribution is a smooth non-convex function, which makes the optimisation of $\mathcal{G}(\cdot,\cdot)$ challenging. However, the cost function defined in Eq. 5, as a whole, can be guaranteed to converge to a minimum, if the following condition is met [29]:

$$\lambda \leq 8\gamma^2. \tag{13}$$

Specifically, the condition given in Eq. 13 ensures that the cost function in the Cauchy proximal operator (Eq. 10) converges. This condition guarantees convergence in the scenario in which the proximal operator needs to be applied in an iterative manner for inverse problems [29] as it ensures

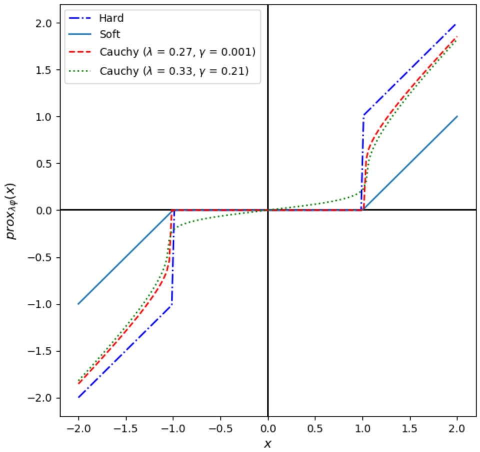


FIGURE 3. Behaviour of the different thresholding algorithms and, in the case of Cauchy, using different parameters.

Algorithm 1 Iterative Cauchy Thresholding

```
Initialise $x$ with 0's
Set $\eta$, $\gamma$ and $\lambda$
Choose stopping criterion. In this work this corresponds to a max number of iterations
while Stopping criterion has not been met do
    Compute $z \leftarrow z - \eta\nabla_z\mathcal{L}(f, z)$
    Shrink every element in $z$ using Eq. (12).
end while
```

Algorithm 2 Cauchy Convolutional Sparse Coding

```
Initialise $z_k$ with 0's, $k = 1, 2, .., K$
Initialise randomly $f_k$, $k = 1, 2, .., K$
Estimate $\gamma$ from the data using Eq. 8
Choose stopping criterion
while Overall stopping criterion has not been met do
    Set $z^{old} \leftarrow z$
    while Stopping criterion for $z$-step has not been met do
        Set $C_O \leftarrow \mathcal{G}(f, z)$
        For every $z_k$ compute:
        $z_k \leftarrow z_k - \eta_z\nabla_{z_k}\mathcal{L}(f, z)$
        Shrink $z_k$ using Eq. (12).
        Set $C_N \leftarrow \mathcal{G}(f, z)$
        if $C_N > C_O$ then
            Set $z \leftarrow z^{old}$
            Set $\eta_z = \eta_z/2$
        else
            Set $z^{old} \leftarrow z$
        end if
    end while
    Set $f^{old} \leftarrow f$
    while Stopping criterion for $f$-step has not been met do
        Set $C_O \leftarrow \mathcal{G}(f, z)$
        For every $f_k$ compute GD on the filers:
        $f_k \leftarrow f_k - \eta_f\nabla_{f_k}\mathcal{L}(f, z)$
        if $C_N > C_O$ then
            Set $f \leftarrow f^{old}$
            Set $\eta_f = \eta_f/2$
        else
            Set $f^{old} \leftarrow f$
        end if
    end while
end while
```

that the Hessian of the expression to minimise in Eq. 10 is positive semidefinite (see Lemma 2 and Theorem 2 in [29] for the details of the mathematical derivation). It is this iterative process the one that gives rise to our proposed ICT algorithm, whose pseudocode is presented in Algorithm 1. Note that an additional parameter η is present as it accounts for the learning rate, thus, the original equation contains $\eta\lambda$, and since $\lambda = 1$ the algorithm has η only, which also affects the convergence condition to $\eta \leq 8\gamma^2$ instead.

This condition is easily applied when the generative model corresponds to CSC. As shown by [29], this arises from the condition being derived by taking the second derivative where the generative model is no longer involved and the resultant expression is dependant only on the hyperparameters.

IV. CAUCHY CONVOLUTIONAL SPARSE CODING

In this section, our new CSC algorithm for representation learning is introduced. It is based on the use of the Cauchy proximal operator through an iterative process in order to encode the data for the z-step. The cost function is derived via MAP. The prior knowledge employed and which then translates into the penalty function corresponds to the assumed statistical distributions of the coefficients [14].

By using the Cauchy distribution in the generative model it is now required to perform the z-step via the Cauchy proximal operator defined in Section III-B. Our goal is thus to solve Eq. (5) for the feature maps using the Cauchy penalty function. Specifically, the cost function is now:

$$\mathcal{G}(f, z) = ||y - \hat{y}||_2^2 - \lambda \sum_{k=1}^{K}\sum_{q=1}^{Q} \log\left(\frac{\gamma}{\pi(\gamma^2 + z_{k,q}^2)}\right), \quad (14)$$

with $\hat{y}$ as defined in Eq. 3. Thus, the full algorithm aims to learn the set of filters f and the feature maps z associated to the data from a dataset of size T. Note that extending the cost function defined in Eq. 5 to learn from more than one sample (i.e. dataset size > 1) is straightforward and hence not detailed in here.

As it is common in similar algorithms, the proposed approach works by alternating optimisation between the learning of the features and the learning of the coefficients, until a convergence criterion is met. This can consist in reducing the reconstruction error below some predefined value or in a maximum number of iterations to be reached. Gradient descent is used as learning approach for the features (f-step) in conjunction with a chosen thresholding algorithm for the coefficients (z-step). In each of the learning steps,

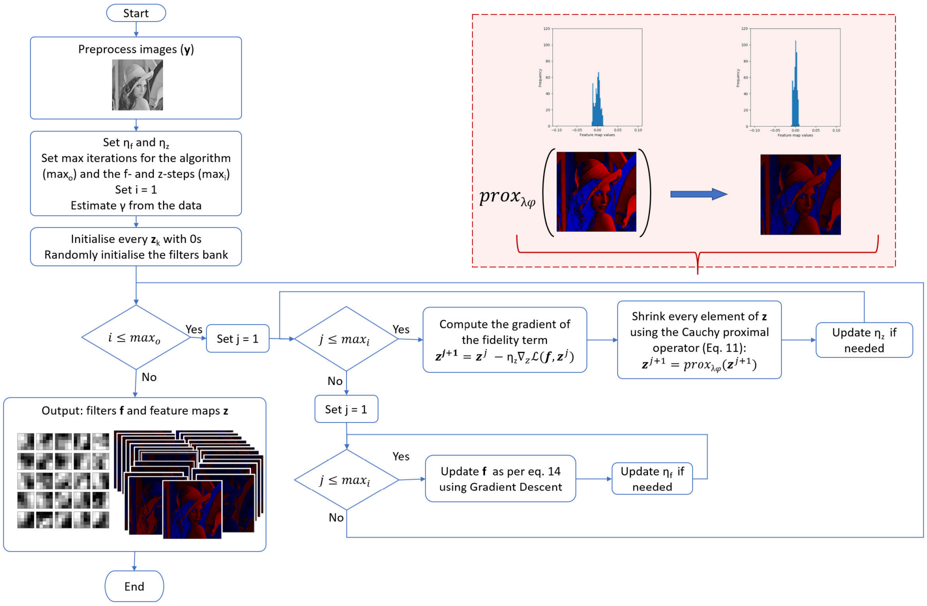

Start
Preprocess images (**y**)
Set η_f and η_z
Set max iterations for the algorithm (max_o) and the f- and z-steps (max_i)
Set i = 1
Estimate γ from the data
Initialise every $\mathbf{z}_k$ with 0s
Randomly initialise the filters bank
$i \leq max_o$
Yes
No
Set j = 1
$j \leq max_i$
Yes
No
Compute the gradient of the fidelity term
$\mathbf{z}^{j+1} = \mathbf{z}^j - \eta_z \nabla_z \mathcal{L}(\mathbf{f}, \mathbf{z}^j)$
Shrink every element of **z** using the Cauchy proximal operator (Eq. 11):
$\mathbf{z}^{j+1} = prox_{\lambda\varphi}(\mathbf{z}^{j+1})$
Update η_z if needed
Set j = 1
$j \leq max_i$
Yes
No
Update **f** as per eq. 14 using Gradient Descent
Update η_f if needed
Output: filters **f** and feature maps **z**
End
$prox_{\lambda\varphi}$
Frequency
Feature map values
−0.10
−0.05
0.00
0.05
0.10
0
20
40
60
80
100
120

FIGURE 4. Block diagram of the Cauchy Convolutional Sparse Coding algorithm. After a few iterations the coefficients within the feature maps get closer to zero.

their respective learning rate is reduced when the cost function increases, so that overshooting over the local minima is prevented. This is achieved by halving the learning rate for the current step following an observed increase in value subsequent to an update.

The f-step is solved by minimising Eq. 14 over f, which can be written compactly as

$$f^* = \arg\min_f \quad ||y - \sum_{k=1}^{K} f_k * z_k||_2^2. \tag{15}$$

This requires taking the gradient over f and choosing a step size η_f for updating the features iteratively. This guarantees convergence since it involves the optimisation of the ℓ_2-norm, which is a smooth convex function.

The key to implementing the Cauchy Convolutional Sparse Coding (CCSC) method consists in using ICT in the encoding phase of the algorithm. This is achieved by solving:

$$z^* = \arg\min_z \quad ||y - \sum_{k=1}^{K} f_k * z_k||_2^2 \\ - \lambda \sum_{k=1}^{K} \sum_{q=1}^{Q} \log\left(\frac{\gamma}{\pi(\gamma^2 + z_{k,q}^2)}\right). \tag{16}$$

In addition to the regularisation parameter λ, one requires also to choose a learning rate η_z. Similarly to what has been done for the f-step, η_z is updated whenever the cost function increases as result of the previous coefficient update. The pseudocode of the whole approach is presented in Algorithm 2 whereas its block diagram is depicted in Figure 4. For reproducibility of the results, the source code of the presented algorithms is provided at https://github.com/p-mayo/cauchycsc

V. SIMULATION RESULTS

In order to quantify the performance of our proposed method, we compared results obtained when using CSC in conjunction with ICT, IHT, IST, and ILT, in a 2D image reconstruction task. The data employed were classical images such as Lena and the Shepp-Logan phantom, as well as the MNIST and AT&T faces[1] datasets. Before applying the representation learning algorithm, independently of the regulariser used, the data have been pre-processed firstly with MinMax normalisation followed by the subtraction of the mean, such that the intensity distribution is centred at the origin (zero-mean). There is no pre-processing done to enforce the dataset to have unit variance since this could affect the estimation of the γ parameter required by the ICT algorithm. Since MNIST and the faces dataset are considerably large, a sample composed of $T = 500$ and $T = 30$ random images therein were used in the respective experiments.

The complete approach was performed 100 times for each dataset using different random initialisation for the filters. For the MNIST and AT&T datasets a random set of samples was also chosen at the beginning of each of their experiments. The maximum number of iterations was fixed to 100 per experiment.

Note that the hyperparameter λ absorbs the learning rate during learning for IHT, IST and ILT, whereas for ICT it was set to 1 in order to leave it as close to the original cost function

[1] https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/

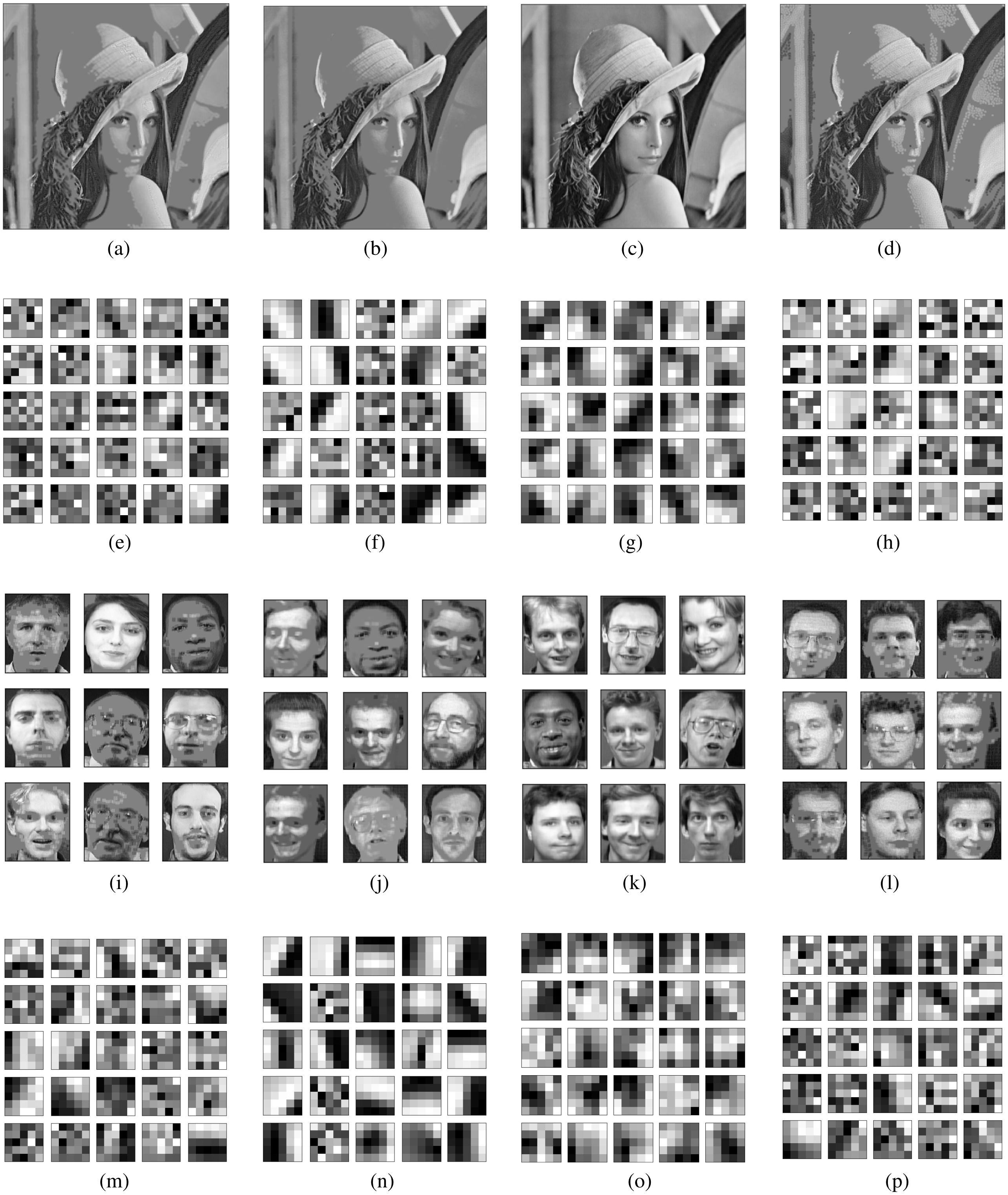


FIGURE 5. Reconstructions of image Lena (first row) and AT&T Faces dataset (third row) using the algorithms (a,i) IHT, (b,j) IST, (c,k) ICT, and (d,l) ILT and the filters learned (second and fourth row, respectively) using (e,m) IHT, (f,n) IST, (g,o) ICT, and (h,p) ILT on second and fourth row.

derived from MAP as possible. Hence, only the estimation of γ is required.

For ICT, the learning rate needs to meet the condition in Eq. 13. The additional tunable parameters employed were $K = 25$ and a filter size of 5×5 for all the experiments.

For an initial qualitative assessment, Figures 5 and 6 show the filters learned for the different datasets, along with the reconstructed images. We show samples from the experiments with the highest Peak Signal-to-Noise Ratio (PSNR) for each algorithm. By visually inspecting Figures 5 and 6,

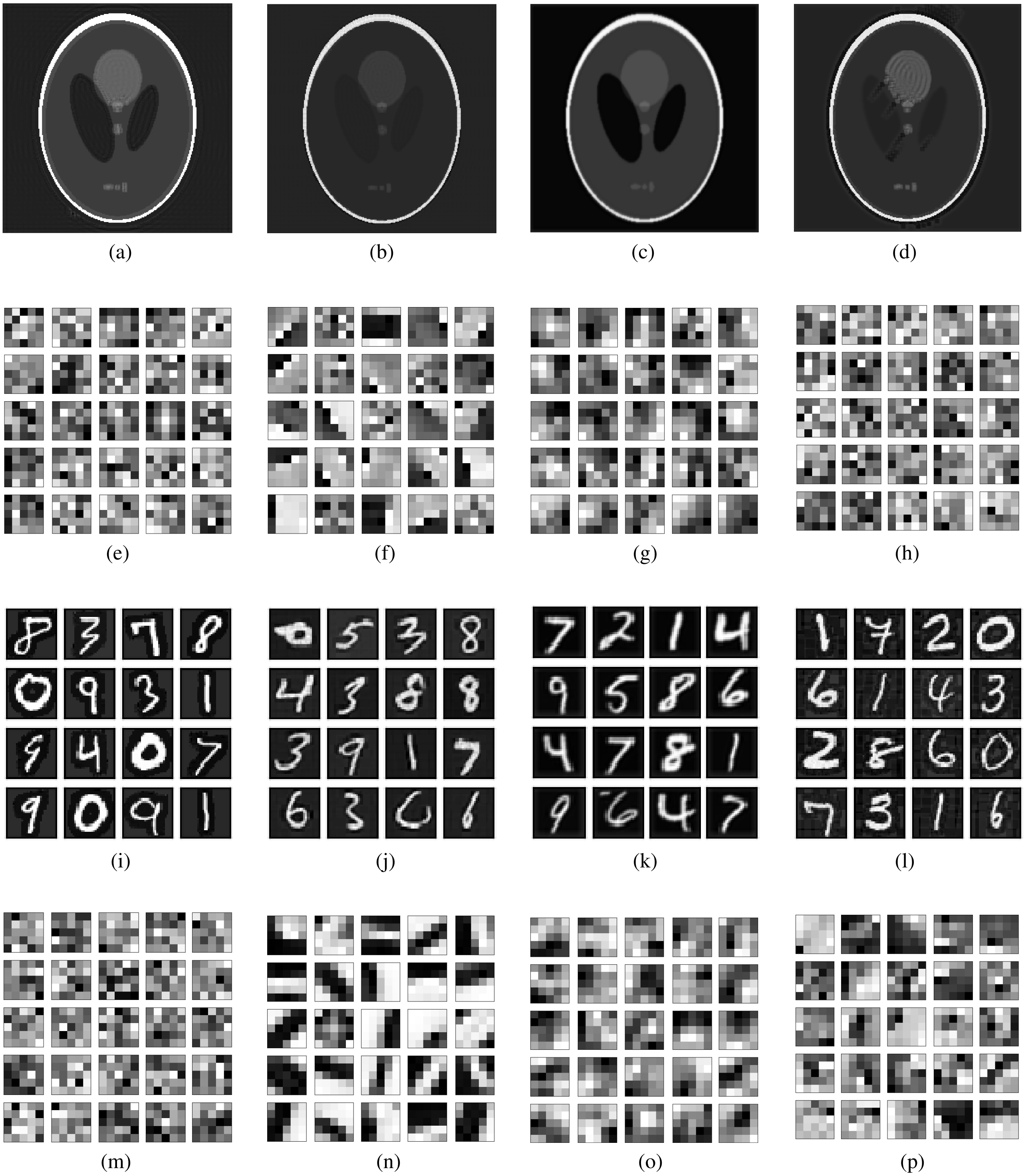


FIGURE 6. Reconstructions of image Shepp-Logan Phantom (first row) and MNIST dataset (third row) using the algorithms (a,i) IHT, (b,j) IST, (c,k) ICT and (d,l) ILT and the filters learned (second and fourth row, respectively) using (e,m) IHT, (f,n) IST, (g,o) ICT and (h,p) ILT.

it can be seen that of the four algorithms assessed, it is the IST algorithm that provides clearer and sharper features. On the other hand, we also notice that ICT can learn a larger number of meaningful filters as observed in the less random patterns they exhibit. By contrast, for IST, IHT and ILT, some of these bases failed to be meaningfully updated. In fact, we noticed that the initialisation of the filters plays an important role in their learning as sometimes there seem to be no learning at all for IHT as the filters have a noisy appearance. This is in spite of their relatively good reconstruction performance with high PSNR values achieved and this confirms the dependence of reconstruction performance on the encoding step [2].

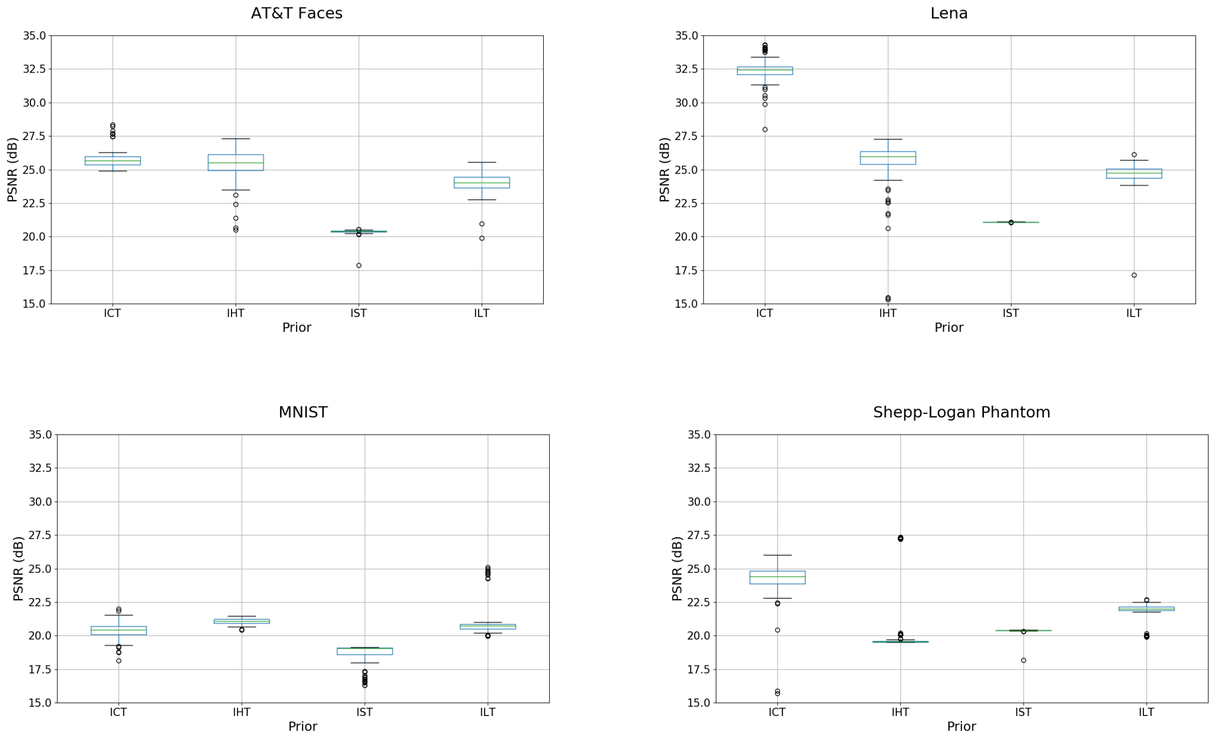


FIGURE 7. Boxplots of PSNR (dB) for CSC using different algorithms for z-step.

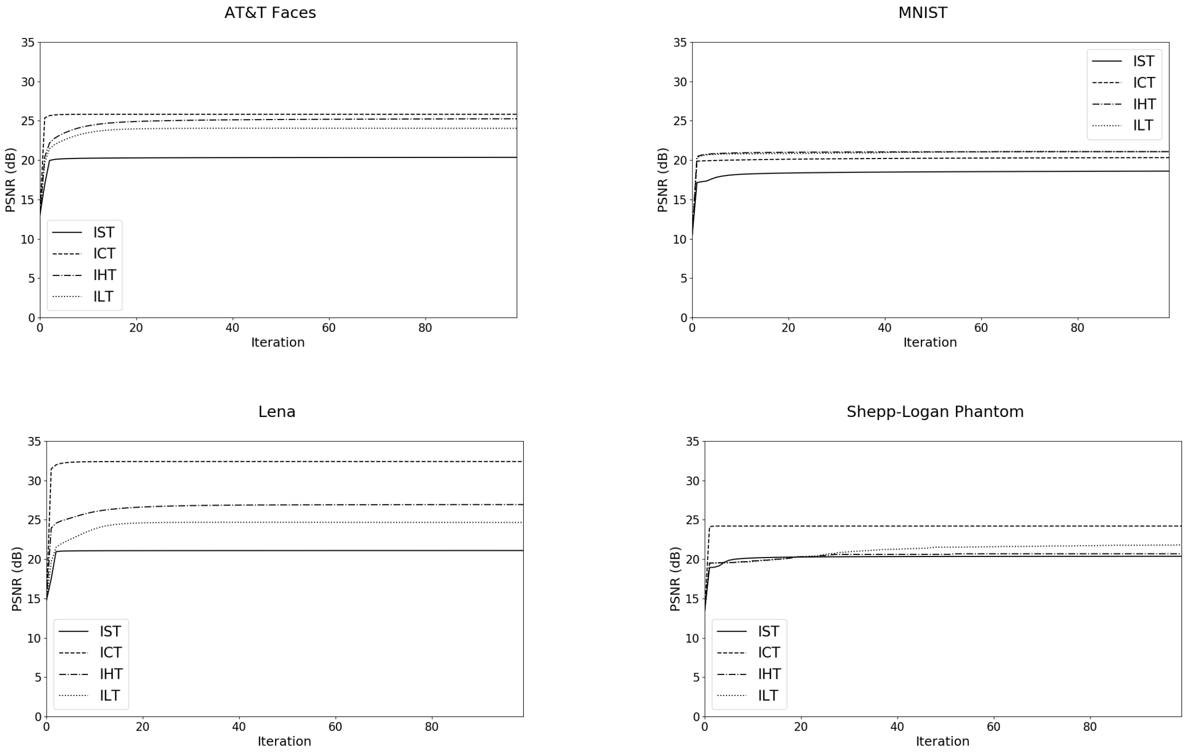


FIGURE 8. Learning iterations vs PSNR (in dB) for CSC using different algorithms for z-step.

The performance of the four representation learning approaches is also assessed through quantitative analysis. The PSNR values for the reconstruction of each sample was computed and their average values as well as their standard deviations are reported in Table 2 and Fig. 7 showing their respective boxplots. Table 3 reports the average proportion of

TABLE 2. PSNR (in dB) results of CSC using different penalty terms.

Dataset	ICT	IHT	IST	ILT
MNIST	20.36 ± 0.61	**21.09 ± 0.22**	18.57 ± 0.82	21.08 ± 1.30
AT&T Faces	**25.82 ± 0.73**	25.31 ± 1.70	20.37 ± 0.26	24.03 ± 0.84
Phantom	**24.20 ± 1.46**	20.67 ± 2.69	20.37 ± 0.22	21.84 ± 0.68
Lena	**32.39 ± 0.89**	25.35 ± 2.59	21.09 ± 0.01	24.65 ± 0.88

TABLE 3. Proportion of non-zero coefficients learned via CSC using different penalty terms.

Dataset	ICT	IHT	IST	ILT
MNIST	99.99 ± 0.00	5.65 ± 0.84	1.16 ± 0.14	**1.03 ± 0.17**
AT&T Faces	99.99 ± 0.00	2.91 ± 1.11	1.65± 0.48	**0.45 ± 0.06**
Phantom	100.00 ± 0.00	2.60 ± 0.66	**1.37 ± 0.11**	88.04 ± 32.54
Lena	99.99 ± 0.00	1.57 ± 3.35	1.60 ± 0.50	**0.30 ± 0.02**

non-zero elements in the learned feature maps. In both Table 2 and Table 3, the best performance for each dataset is shown in bold.

Lastly, in Fig. 8, a plot of the learning performance as function of average PSNR as the iterations progress is provided. From Fig. 8 we can see that ICT and IST reach the plateau corresponding to the highest PSNR early in the learning process, with IHT and ILT reaching their own maximum a few iterations later. It is ICT, however, the one that achieves the highest PSNR and requires the least iterations. The three algorithms, IHT, IST and ILT, require tuning of a number of parameters for optimal performance, whereas for ICT the parameter γ is estimated directly from the data. In fact, the use of the iterative Cauchy algorithm requires the choice of only two values, the learning rate and the scale parameter. As noted in section III-B, γ can be estimated from the original data whilst η_z needs to obey the condition in Eq. 13.

From Table 2 we can see that ICT provides the best PSNR performances in three out of four cases, which is consistent with the visual evaluation. IST, on the other hand, is the one with the worst reconstruction performances although the features learned with it are seemingly sharper in comparison to ICT. IHT and ILT lie in between in terms of reconstruction performance. It is also observed that ILT offers the sparsest feature maps whilst producing a reasonable PSNR for most datasets. Interestingly, for the Shepp-Logan phantom image, ILT did not provide the sparsest solution and produced many more non-zero values in contrast to its behaviour on the other datasets. We believe that this could be remediated by a lengthy fine tuning of the hyperparameters η, δ and λ. IST presents more consistency in regard of the PSNR results obtained as the inter-quartiles range is shorter than the other two algorithms, as seen in Fig. 7.

We observe an increase of the average PSNR as the images increase in size for the four thresholding algorithms considered, with ICT exhibiting the highest of such jumps. Indeed, ICT performed better as the size of the images increased, having very similar performance to IHT and ILT for the small size MNIST (28 × 28) dataset as opposed to the case of the Shepp-Logan phantom (256 × 256) and Lena (512 × 512) images. Furthermore, despite the high PSNR values obtained with IHT, some degradation in the reconstructed images surrounding the edges and the lose of details is apparent. In the case of IST, the images exhibit a smoothing effect regardless of their dimension. Lastly, the reconstructed images produced by ICT also present some artifacts near the edges, which become more apparent in smaller image sizes. Further experiments would be needed to inspect this behaviour in more detail as the image content varies greatly from dataset to dataset. For example, Lena and the AT&T datasets are richer in information than the MNIST and Shepp-Logan phantom.

With respect to Table 3, it is evident that ICT is the approach that offers the least sparse solutions. Having a closer look to the histograms of coefficients (Fig. 9) it can be observed that most coefficients are in a very close vicinity of 0, which might explain the ability to learn most of the features most of the times whilst reducing their noisy appearances. Even though ICT does not promote sparsity in the same way the ℓ_1- and ℓ_0- norms do, it is based on a function with various important statistical characteristics such as ability to model non-Gaussianity and heavy-tails remarkably well compared to the ℓ_1- and ℓ_0- norms. Due to its statistical characteristics and closed-form proximal operator existence, we believe CCSC to be an excellent tool for optimal image data representation.

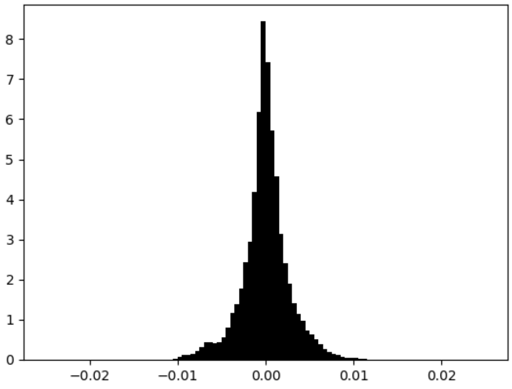


FIGURE 9. Histogram of coefficient values from the 25 feature maps involved in the reconstruction of the image Lena using ICT. Y axis scale factor: 10^5.

Of course, we are aware of the existence of approaches such as Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [41] or the Accelerated Iterative Hard Thresholding [42] for the ℓ_1- and the ℓ_0-norm, respectively, which can lead to better reconstruction performance. However, investigating this is beyond the scope of this paper, since our interest is in demonstrating the benefits of using a better statistical model as the driving force behind such algorithms, while enhancements similar to those in [41] could be incorporated in our approach as well.

VI. CONCLUSION AND FUTURE WORK

In this work a new convolutional sparse coding framework based on a Cauchy model assumption is proposed. This approach enables the learning of filters and their respective feature maps by using said distribution as prior for the coefficients in the latter, which results in a new cost function. The Cauchy proximal operator was derived and used for its optimisation. This requires a preliminary step before the learning process, which involves the estimation of the corresponding scale parameter. In fact, we consider this to be one of the strongest features of our approach, as it requires virtually no tuning of hyper-parameters, as opposed to IHT, ICT, and ILT, for which a bad choice of parameters is likely to have a negative impact in the output of the algorithms. The performance was evaluated on four different datasets and compared against the reconstruction performance achieved using existing methods. Even though CCSC does not achieve the same degree (and type) of sparsity as the other three algorithms, the filters learned are seemingly better for the reconstruction task based on their higher PSNR values achieved. The current implementation of the ICT algorithm is computationally more demanding than the other methods included in this study, as can be seen from Table 1 and Eq. 12.

Our current work focuses on investigating the discriminative power of the proposed representation in classification problems. In addition, further investigations will be conducted on the potential use of the CCSC algorithm in a wider variety of scenarios and for solving different imaging inverse problems.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, ''Representation learning: A review and new perspectives,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[2] A. Coates and A. Y. Ng, ''The importance of encoding versus training with sparse coding and vector quantization,'' in *Proc. ICML*, 2011, pp. 921–928. [Online]. Available: https://icml.cc/2011/papers/485_icmlpaper.pdf

[3] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, ''An augmented linear mixing model to address spectral variability for hyperspectral unmixing,'' *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[4] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, ''Sparse representations in audio and music: From coding to source separation,'' *Proc. IEEE*, vol. 98, no. 6, pp. 995–1005, Jun. 2010.

[5] O. Bryt and M. Elad, ''Compression of facial images using the K-SVD algorithm,'' *J. Vis. Commun. Image Represent.*, vol. 19, no. 4, pp. 270–282, May 2008.

[6] I. Horev, O. Bryt, and R. Rubinstein, ''Adaptive image compression using sparse dictionaries,'' in *Proc. 19th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, 2012, pp. 592–595.

[7] W. Fu, S. Li, L. Fang, and J. A. Benediktsson, ''Adaptive spectral–spatial compression of hyperspectral image with sparse representation,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 671–682, Oct. 2016.

[8] M. Elad and M. Aharon, ''Image denoising via sparse and redundant representations over learned dictionaries,'' *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[9] A. Achim, A. Bezerianos, and P. Tsakalides, ''Novel Bayesian multiscale method for speckle removal in medical ultrasound images,'' *IEEE Trans. Med. Imag.*, vol. 20, no. 8, pp. 772–783, Aug. 2001.

[10] C. Jiang, Q. Zhang, R. Fan, and Z. Hu, ''Super-resolution CT image reconstruction based on dictionary learning and sparse representation,'' *Sci. Rep.*, vol. 8, no. 1, pp. 1–10, Dec. 2018.

[11] Y. Huang, L. Shao, and A. F. Frangi, ''Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6070–6079.

[12] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, ''Low-rank sparse coding for image classification,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 281–288.

[13] B. Chen, J. Li, B. Ma, and G. Wei, ''Convolutional sparse coding classification model for image classification,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1918–1922.

[14] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, ''Shift-invariance sparse coding for audio classification,'' 2012, *arXiv:1206.5241*. [Online]. Available: http://arxiv.org/abs/1206.5241

[15] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, ''Sparse coding with anomaly detection,'' *J. Signal Process. Syst.*, vol. 79, no. 2, pp. 179–188, May 2015.

[16] O. Karakuş, N. Anantrasirichai, A. Aguersif, S. Silva, A. Basarab, and A. Achim, ''Detection of line artifacts in lung ultrasound images of COVID-19 patients via nonconvex regularization,'' *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 67, no. 11, pp. 2218–2229, 2020, doi: 10.1109/TUFFC.2020.3016092.

[17] A. Majumdar and V. Singhal, ''Noisy deep dictionary learning: Application to Alzheimer's disease classification,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2679–2683.

[18] H. Hongxing, J. M. Bioucas-Dias, and V. Katkovnik, ''Interferometric phase image estimation via sparse coding in the complex domain,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2587–2602, May 2015.

[19] J. Kang, D. Hong, J. Liu, G. Baier, N. Yokoya, and B. Demir, ''Learning convolutional sparse coding on complex domain for interferometric phase restoration,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 826–840, Feb. 2021.

[20] O. Karakuş and A. Achim, ''On solving SAR imaging inverse problems using nonconvex regularization with a Cauchy-based penalty,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5828–5840, 2021, doi: 10.1109/TGRS.2020.3011631.

[21] S. Wold, K. Esbensen, and P. Geladi, ''Principal component analysis,'' *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[22] A. J. Bell and T. J. Sejnowski, ''An information-maximization approach to blind separation and blind deconvolution,'' *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.

[23] M. A. Kramer, ''Nonlinear principal component analysis using autoassociative neural networks,'' *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.

[24] I. Tosic and P. Frossard, ''Dictionary learning,'' *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[25] B. A. Olshausen and D. J. Field, ''Sparse coding with an overcomplete basis set: A strategy employed by V1?'' *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[26] I. Daubechies, M. Defrise, and C. De Mol, ''An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,'' *Commun. Pure Appl. Math., A J. Courant Inst. Math. Sci.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[27] T. Blumensath and M. E. Davies, ''Iterative thresholding for sparse approximations,'' *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 629–654, Dec. 2008.

[28] D. Malioutov and A. Aravkin, ''Iterative log thresholding,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 7198–7202.

[29] O. Karakus, P. Mayo, and A. Achim, ''Convergence guarantees for nonconvex optimisation with Cauchy-based penalties,'' *IEEE Trans. Signal Process.*, vol. 68, pp. 6159–6170, 2020.

[30] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, ''Deconvolutional networks,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.

[31] M. Aharon, M. Elad, and A. Bruckstein, ''K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,'' *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[32] P. Pad, F. Salehi, E. Celis, P. Thiran, and M. Unser, ''Dictionary learning based on sparse distribution tomography,'' in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2731–2740.

[33] F. Wen, L. Chu, P. Liu, and R. C. Qiu, ''A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning,'' *IEEE Access*, vol. 6, pp. 69883–69906, 2018.

[34] J.-J. Mei, Y. Dong, T.-Z. Huang, and W. Yin, ''Cauchy noise removal by nonconvex ADMM with convergence guarantees,'' *J. Sci. Comput.*, vol. 74, no. 2, pp. 743–766, Feb. 2018.

[35] F. Sciacchitano, Y. Dong, and T. Zeng, ''Variational approach for restoring blurred images with Cauchy noise,'' *SIAM J. Imag. Sci.*, vol. 8, no. 3, pp. 1894–1922, 2015.

[36] T. Wan, N. Canagarajah, and A. Achim, ''Segmentation of noisy colour images using Cauchy distribution in the complex wavelet domain,'' *IET Image Process.*, vol. 5, no. 2, pp. 159–170, 2011.

[37] A. Achim and E. E. Kuruoglu, ''Image denoising using bivariate α-stable distributions in the complex wavelet domain,'' *IEEE Signal Process. Lett.*, vol. 12, no. 1, pp. 17–20, Jan. 2005.

[38] M. I. H. Bhuiyan, M. O. Ahmad, and M. N. S. Swamy, ''Spatially adaptive wavelet-based method using the Cauchy prior for denoising the SAR images,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 500–507, Apr. 2007.

[39] J. J. Ranjani and S. J. Thiruvengadam, ''Dual-tree complex wavelet transform based SAR despeckling using interscale dependence,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2723–2731, Jun. 2010.

[40] Q. Gao, Y. Lu, D. Sun, Z.-L. Sun, and D. Zhang, ''Directionlet-based denoising of SAR images using a Cauchy model,'' *Signal Process.*, vol. 93, no. 5, pp. 1056–1063, May 2013.

[41] A. Beck and M. Teboulle, ''A fast iterative shrinkage-thresholding algorithm for linear inverse problems,'' *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[42] T. Blumensath, ''Accelerated iterative hard thresholding,'' *Signal Process.*, vol. 92, no. 3, pp. 752–756, Mar. 2012.

PERLA MAYO (Student Member, IEEE) received the B.Sc. degree (Hons.) in computer systems engineering from the Autonomous University of Aguascalientes, Mexico, in 2013, and the M.Sc. degree in biomedical engineering from the University of Bristol, U.K., in 2017, where she is currently pursuing the Ph.D. degree. In between the ending of her bachelor's and master's degrees, she joined Multinational Company as a Software Developer. In 2017, she joined the Visual Information Laboratory, Department of Electrical and Electronic Engineering, University of Bristol. Her research interests include the image processing and classification fields, with particular interest in image encoding under sparsity constraints, this mainly for biomedical applications.

OKTAY KARAKUŞ (Member, IEEE) received the B.Sc. degree (Hons.) in electronics engineering from Istanbul Kültür University, Turkey, in 2009, and the M.Sc. and Ph.D. degrees in electronics and communication engineering from the İzmir Institute of Technology (IZTECH), Turkey, in 2012 and 2018, respectively. From October 2009 to January 2018, he was associated with the Department of Electrical and Electronics Engineering, Yaşar University, Turkey, and the Department of Electronics and Communication Engineering, İzmir Institute of Technology, as a Research Assistant. He was a Visiting Scholar with the Institute of Information Science and Technologies (ISTI-CNR), Pisa, Italy, in 2017. Since March 2018, he has been a Research Associate in image processing with the Visual Information Laboratory, Department of Electrical and Electronic Engineering, University of Bristol. His research interests include statistical/Bayesian signal and image processing, inverse problems with applications on SAR and ultrasound imagery, heavy tailed data modeling, and telecommunications and energy.

ROBIN HOLMES received the B.Sc. degree (Hons.) in physics from the University of Southampton, U.K., in 1991, the M.Sc. degree in medical physics from the University of Exeter, U.K., in 1993, and the Ph.D. degree in the analyses of functional neuroimaging from the University of Bristol, U.K., in 2012. From 2014 to 2018, he was a Postdoctoral Research Fellow with the National Institute of Health Research, U.K. His research interests include image simulation using 3-D printing and the subresolution sandwich method.

ALIN ACHIM (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the POLITEHNICA University of Bucharest, Romania, in 1995 and 1996, respectively, and the Ph.D. degree in biomedical engineering from the University of Patras, Greece, in 2003. In October 2004, he joined the Department of Electrical and Electronic Engineering, University of Bristol, U.K., as a Lecturer, became a Senior Lecturer (Associate Professor), in 2010, and a Reader in biomedical image computing, in 2015. Since August 2018, he has been the Chair in computational imaging with the University of Bristol. From 2019 to 2020, he was a Leverhulme Trust Research Fellow with the Laboratoire I3S, Université Cote d'Azur. He has coauthored over 140 scientific publications, including 45 journal articles. His research interests include statistical signal, image and video processing, with particular emphasis on the use of sparse distributions within sparse domains and with applications in both biomedical imaging and remote sensing. He was/is an Elected Member of the Bio Imaging and Signal Processing Technical Committee, IEEE Signal Processing Society, an Affiliated Member (invited) of the Signal Processing Theory and Methods Technical Committee, IEEE Signal Processing Society, and a member of the IEEE Geoscience and Remote Sensing Society's Image Analysis and Data Fusion Technical Committee. He then obtained an ERCIM (European Research Consortium for Informatics and Mathematics) Postdoctoral Fellowship which he spent with the Institute of Information Science and Technologies (ISTI-CNR), Pisa, Italy, and with the French National Institute for Research in Computer Science and Control (INRIA), Sophia Antipolis, France. He is currently a Senior Area Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, an Associate Editor of the IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING, and an Editorial Board Member of *Remote Sensing* (MDPI).