# The DUDE Framework for Grayscale Image Denoising

Giovanni Motta, Erik Ordentlich, Ignacio Ramirez, Gadiel Seroussi, Marcelo J. Weinberger

HP Laboratories
HPL-2009-252

**Keyword(s):**
Image denosing, impulse noise, discrete universal denoising, dude

## Abstract:

We present an extension of the Discrete Universal DEnoiser (DUDE) specialized for the denoising of grayscale images. The original DUDE is a low-complexity algorithm aimed at recovering discrete sequences corrupted by discrete memoryless noise of known statistical characteristics. It is universal, in the sense of asymptotically achieving, without access to any information on the statistics of the clean sequence, the same performance as the best denoiser that does have access to such information. The denoising performance of the DUDE, however, is poor on grayscale images of practical size. The difficulty lies in the fact that one of the DUDE's key components is the determination of conditional empirical probability distributions of image samples, given the sample values in their neighborhood. When the alphabet is moderately large (as is the case with grayscale images), even for a small-sized neighborhood, the required distributions would be estimated from a large collection of sparse statistics, resulting in poor estimates that would cause the algorithm to fall significantly short of the asymptotically optimal performance. The present work enhances the basic DUDE scheme by incorporating statistical modeling tools that have proven successful in addressing similar issues in lossless image compression. The enhanced framework is tested on additive and non-additive noise, and shown to yield powerful denoisers that significantly surpass the state of the art in the case of non-additive noise, and perform well for Gaussian noise.

External Posting Date: September 6, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: September 6, 2009 [Fulltext]

# The DUDE Framework for Grayscale Image Denoising

Giovanni Motta, Erik Ordentlich, Ignacio Ramírez, Gadiel Seroussi, and Marcelo J. Weinberger

**Abstract**

We present an extension of the Discrete Universal DEnoiser (DUDE) specialized for the denoising of grayscale images. The original DUDE is a low-complexity algorithm aimed at recovering discrete sequences corrupted by discrete memoryless noise of known statistical characteristics. It is universal, in the sense of asymptotically achieving, without access to any information on the statistics of the clean sequence, the same performance as the best denoiser that does have access to such information. The denoising performance of the DUDE, however, is poor on grayscale images of practical size. The difficulty lies in the fact that one of the DUDE's key components is the determination of conditional empirical probability distributions of image samples, given the sample values in their neighborhood. When the alphabet is moderately large (as is the case with grayscale images), even for a small-sized neighborhood, the required distributions would be estimated from a large collection of sparse statistics, resulting in poor estimates that would cause the algorithm to fall significantly short of the asymptotically optimal performance. The present work enhances the basic DUDE scheme by incorporating statistical modeling tools that have proven successful in addressing similar issues in lossless image compression. The enhanced framework is tested on additive and non-additive noise, and shown to yield powerful denoisers that significantly surpass the state of the art in the case of non-additive noise, and perform well for Gaussian noise.

## I. Introduction

The *discrete universal denoiser* (DUDE), introduced in [1], [2], aims at recovering a discrete, finite-alphabet sequence, after it has been corrupted by discrete memoryless noise of known statistical characteristics. It is shown in [2] that the DUDE is universal, in the sense of asymptotically achieving, without access to any information on the statistics of the clean sequence, the same performance as an optimal denoiser *with* access to such information. It can also be implemented with low complexity. In [3], the definition of the DUDE was formally extended to two-dimensionally indexed data, and an implementation of the scheme for binary images was shown to outperform other known schemes for denoising this type of data.

The DUDE algorithm performs two passes over the data. In a first pass, a conditional probability distribution is determined for each (noisy) sample given the sample values in a (spatial) neighborhood, or *context*, by collecting statistics of joint occurrences. This *context model* is then used for determining conditional probability distributions

G. Motta is with Qualcomm Inc., San Diego, California, U.S.A. E-mail: gmotta@qualcomm.com. This work was done while the author was with Hewlett-Packard Company.

E. Ordentlich, G. Seroussi, and M. J. Weinberger are with Hewlett-Packard Laboratories, Palo Alto, California, U.S.A. E-mail: {erik.ordentlich, gadiel.seroussi, marcelo.weinberger}@hp.com.

I. Ramírez is with Instituto de Ingenería Eléctrica, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay. E-mail: nacho@fing.edu.uy. Part of this work was done while the author was visiting Hewlett-Packard Laboratories, Palo Alto, California, U.S.A.

for *clean* samples given each (noisy) context pattern and the sample value observed at the corresponding location. In a second pass, a denoising decision is made for each sample based on this conditional distribution. The decision is essentially the Bayes optimal one with respect to the above distribution and a given loss function. A more detailed description of the DUDE algorithm is given in Section II.

Although the asymptotic results of [2] apply to any finite alphabet, it was observed in [3] that extending the results to grayscale images[1] (or, in general, to data over large alphabets) presented significant challenges. The main challenge stems from the fact that, in a context model over an alphabet of size $M$, parametrized by the symbol conditional probabilities, and with a neighborhood of size $d$, the number of free parameters is $M^d(M{-}1)$ (for example, in an 8-bit image, a rather small $3 \times 3$ neighborhood consisting of the eight samples closest to a given sample, yields $2^{64}{\cdot}255 \approx 5{\cdot}10^{21}$ free parameters). This means that context-conditioned statistics for estimating these parameters are likely to be sparse and provide little, if any, information on the structure of the image. This well known phenomenon is sometimes referred to as the “sparse context” problem. The theoretical results of [2] indeed show that the DUDE’s rate of convergence to optimal performance depends strongly on the size of the context model. This convergence rate is determined largely by the degree to which the law of large numbers has taken hold on random subsequences of noisy samples occurring in a given context pattern and having a given underlying clean sample value. Convergence requires that these subsequences be relatively long, implying numerous occurrences of each noisy pattern and underlying clean sample value.

Due to these facts, the original DUDE scheme, as defined in [2], will not approach optimal performance for images of current or foreseeable practical size over a large (say, 256-symbol) alphabet. This problem has also been noticed in [4, p. 509], where the direct application of the DUDE’s tools to model the necessary conditional probability distributions for grayscale images is deemed to be “almost hopeless.” Although this pessimistic assessment appears justified at first sight, we show that the basic scheme of [2] can be enhanced with image modeling tools, enabling effective implementations of the DUDE framework in grayscale image applications.

A “sparse context” problem very similar to that confronting the DUDE exists, and has been successfully addressed, in lossless image compression (see, e.g., [5], [6], or the survey [7]), where state-of-the-art algorithms are also based on the determination of probability distributions of samples of the input image, conditioned on their contexts.[2] In this and other inference problems the concept is formalized by the notion of *model cost* [8], a penalty proportional to the number of free statistical parameters in the model, which is paid to learn or describe the estimated model parameters. The principle underlying the tools developed for lossless image compression is that one should not impose on the universal algorithm the task of learning properties of the data *which are known a-priori*. For example, one instance of the problem is formally studied in [9], where it is shown how the widely used practice of coding prediction errors (rather than original samples) can be seen as allowing the statistics of the data to be learned, effectively,

[1] These are images with a relatively large dynamic range (e.g., in our examples, $256$ grayscale values), for which the main assumption is that the numerical sample values preserve, up to quantization, continuity of brightness in the physical world. Most of our discussion will refer to monochrome grayscale images, although the algorithms extend by the usual methods to color images. Notice also that we assume a truly *discrete* setting: the noisy symbols are discrete, and they are assumed to belong to the same finite alphabet as the clean symbols (e.g., we consider discrete Gaussian noise). Other works in the literature often assume that the noisy samples are arbitrary real numbers, which provides the denoiser with more information than the corresponding quantized and possibly clipped values assumed in the discrete setting. It can be argued that such continuous information is often not available in practice, e.g. when denoising a raw digital image acquired by a digital camera.

[2] In the image compression case, the contexts are causal, whereas for the DUDE, the contexts are generally non-causal.

with a much smaller model. The use of prediction is based on our prior knowledge of the targeted images being generally smooth, and of the fact that similar variations in brightness are likely to occur in regions of the image with different baseline brightness levels. Explicit or implicit application of these principles has led to some of the best schemes in lossless image compression [5], [6], which are all based on prediction and context modeling.

The foregoing discussion suggests that modeling tools developed and tested in lossless image compression could be leveraged for estimating the distributions required by the DUDE, together with tools that are specific to the assumptions of the denoising application. In this paper, we pursue this strategy, and show that enhancing the basic DUDE algorithm with such tools yields powerful and practical denoisers for images corrupted by a variety of types of noise. Although the goal of this work is not necessarily to achieve the current record in denoising performance for each image and each type of noise studied, the enhanced DUDE scheme significantly surpasses the state-of-the-art for some important types of (non-additive) noise, and performs well for Gaussian noise. We expect that further refinements, extensive experimentation, and synergistic incorporation of ideas from other approaches will enable improvements in denoising performance over the results reported on here. We refer to the enhanced DUDE schemes described in the paper as a *framework*, since what is described is a general architecture for a denoising system following the basic DUDE structure from [2], which is then specialized for each noise type of interest. Thus, although all the schemes described will have the same basic components fulfilling the same functions, the specific details of some of the components will vary depending on the type of noise targeted.

Denoising of signals and in particular digital images has been given considerable attention by the signal processing community for decades, starting from the works by Kalman [10] and Wiener [11]. A comprehensive review included in [4] gives an account of the rapid progress on the problem in recent years. Inspection of the literature reveals that the work is divided into two fairly disjoint classes: additive (usually Gaussian) noise, and non-additive noise (the latter includes multiplicative noise, although our focus in this class will be mostly on the so-called *impulse* noise types). We survey the main ideas behind a few of the recent approaches, which either have some conceptual connections to our own, or will be used as references in the results of Section V. The reader is referred to [4] and references therein for a more comprehensive account, including the *wavelet thresholding* approach of [12] and the *Gaussian scale mixtures* of [13].

For additive noise, the most relevant schemes are presented in [4] and [14] and are discussed next; other approaches include the *fields of experts* of [15] and the *sparse representation* of [16] (which is combined with the *multiscale approach* in [17]). Also, in a different offshoot of [2], image denoising schemes for Gaussian noise [18] have been derived from extensions of the DUDE ideas to continuous-alphabet signals [19]. The non-parametric Bayesian least squares estimator developed in [4] is predicated on the observation that "every small window in a natural image has many similar windows in the same image." The method uses this assumption to estimate each clean image sample $x_i$ as a weighted average of all the samples $z_j$ in the noisy image, where the weight of $z_j$ increases monotonically with a measure of similarity between the contexts of $z_j$ and $z_i$ (the noisy version of $x_i$). Despite being couched in very different mathematical languages, there is much affinity between the approach in [4] and the one in this paper—taken to bare-bones simplicity, the DUDE approach can be seen as also taking advantage of the quoted observation. This concept has been combined in [14] with a three-dimensional (3D) DCT-coefficient denoising technique, resulting in a scheme that achieves unprecedented performance for Gaussian noise. In this scheme, image windows with sample values similar to those in a window surrounding the sample to be denoised are aggregated into a 3D array which is then denoised in the 3D DCT domain using thresholding or Wiener filtering. This procedure is repeated for multiple relative positions of the same noisy sample in the window, and the final estimate for that sample is obtained as a weighted average.

Although the models in the above works could be used with other types of noise, some of them were specifically designed with additive Gaussian noise in mind, and the results published are for that type of noise. Non-additive noise, on the other hand, poses different problems. A typical example for this type of noise is the *Salt and Pepper* (S&P) noise, where a portion of the image samples are saturated to either totally black or totally white. For this type of noise where outliers are very common, median-based estimators are widespread and fairly effective. Works like [20], [21] or [22] also exploit the fact that it is possible to identify with good accuracy candidate noisy samples, so as to avoid changing samples that are not corrupted, and sometimes to exclude noisy samples from some computations. Impulse noise strongly impacts image gradients, and therefore the *variational* approach of [23] (see also [24]) is well-suited. In this approach, used in [25] and [26] to denoise highly corrupted images, the denoised image is the result of a tradeoff between fidelity and total variation. While the fidelity term measures the difference between an image model based on edge-preserving priors and the observed data, the total variation term measures the "roughness" of the image. Another, more difficult type of impulse noise is the *$M$-ary symmetric* noise, where $M$ stands for the alphabet size of the clean (and noisy) signals. In this type of noise, a sample is substituted, with some probability, by a random, uniformly distributed value from the alphabet, and a simple thresholding cannot separate out the clean samples. Image denoising for $M$-ary symmetric noise is also addressed in [25] and [21].

The rest of the paper is organized as follows. In Section II we review the basic concepts, notations, and results from [2] and [3]. Section III describes the tools used to address model cost and other challenges in the enhanced DUDE framework. Section IV describes the implementation of the framework for S&P, $M$-ary symmetric noise, and Gaussian noise. In Section V we describe experiments performed with the resulting denoisers, comparing whenever possible with other denoisers from the literature.

## II. BASIC DUDE

In this section, we review the basic DUDE algorithm from [2], as extended to two-dimensional data in [3].

### A. Notation and problem setting

Throughout, an *image* is a two-dimensional array over a finite alphabet $\mathcal{A}$ of size $|\mathcal{A}|=M$ (without loss of generality, $\mathcal{A} = \{0, 1, \ldots, M-1\}$). We let $\mathbf{x}^{m\times n} \in \mathcal{A}^{m\times n}$ denote an $m \times n$ image, also denoted $\mathbf{x}$ when the superscript $m \times n$ is clear from the context. Let $\mathbb{Z}$ denote the set of integers, and let $V_{m\times n}$ denote the set of two-dimensional indices $V_{m\times n} = \{(i_r, i_c) \in \mathbb{Z}^2 \,|\, 1 \le i_r \le m,\ 1 \le i_c \le n\}$. The $i$-th component of a vector $\mathbf{u}$ will be denoted by $u_i$, or sometimes $\mathbf{u}[i]$ when $\mathbf{u}$ is a vector expression. Similarly, we denote a typical entry of $\mathbf{x}$ by $x_i$ (or $\mathbf{x}[i]$), $i \in V_{m\times n}$. When the range of an image index $i$ is not specified, it is assumed to be $V_{m\times n}$.

We assume that a clean image $\mathbf{x}$ is corrupted by *discrete memoryless noise* characterized by a transition probability matrix $\mathbf{\Pi} = \{\Pi(a,b)\}_{a,b\in\mathcal{A}}$, where $\Pi(a,b)$ is the probability that the noisy symbol is $b$ when the input symbol is $a$. The noise affects each sample in the clean image $\mathbf{x}^{m\times n}$ independently, resulting in a noisy image $\mathbf{z}^{m\times n}$, where $z_i$ is a random variable distributed according to $P(z_i = b) = \Pi(x_i, b),\ b \in \mathcal{A}$. We regard this process as $\mathbf{x}^{m\times n}$ going through a *noisy channel*, refer to $\mathbf{\Pi}$ as the *channel matrix*, and to $\mathbf{z}^{m\times n}$ as the *channel output*. We assume, for simplicity, that the clean and noisy images are defined over the same alphabet—the setting in [2] is more general, allowing for different input and output alphabets. We also assume, following [2], that $\mathbf{\Pi}$ is non-singular. In later sections of this paper, however, we will consider some channel matrices which are non-singular but badly conditioned and we treat them, in practice, as singular.

A $m{\times}n$ *image denoiser* is a mapping $\hat{\boldsymbol{\chi}}^{m\times n} : \mathcal{A}^{m\times n}{\rightarrow}\mathcal{A}^{m\times n}$. Assume a given per-symbol *loss function* $\Lambda : \mathcal{A}^2{\rightarrow}[0,\infty)$, represented by a matrix $\mathbf{\Lambda} = \{\Lambda(a,b)\}_{a,b\in\mathcal{A}}$, where $\Lambda(a,b)$ is the loss incurred by estimating the symbol $a$ with the symbol $b$. For $\mathbf{x}, \mathbf{z} \in \mathcal{A}^{m\times n}$ we let $L_{\hat{\boldsymbol{\chi}}}(\mathbf{x},\mathbf{z})$ denote the normalized denoising loss, as measured by $\Lambda$, of the image denoiser $\hat{\boldsymbol{\chi}}^{m\times n}$ when the observed noisy image is $\mathbf{z}$ and the underlying clean one is $\mathbf{x}$, i.e.,

$$L_{\hat{\boldsymbol{\chi}}}(\mathbf{x},\mathbf{z}) = \frac{1}{mn}\sum_{i\in V_{m\times n}} \Lambda(x_i, \hat{\boldsymbol{\chi}}^{m\times n}(\mathbf{z})[i]),$$

where we recall that $\hat{\boldsymbol{\chi}}^{m\times n}(\mathbf{z})[i]$ is the component of $\hat{\boldsymbol{\chi}}^{m\times n}(\mathbf{z})$ at the $i$-th location. We seek denoisers that minimize this loss in a stochastic sense (under the distribution generated by the channel). Notice that the mapping $\hat{\boldsymbol{\chi}}^{m\times n}$ may depend on the channel matrix $\mathbf{\Pi}$ and the loss function $\Lambda$, but not on the clean image $\mathbf{x}$, i.e., given $\mathbf{\Pi}$ and $\Lambda$, a noisy image $\mathbf{z}$ will always result in the same denoised image $\hat{\boldsymbol{\chi}}^{m\times n}(\mathbf{z})$, independently of which combination of clean image and noise realization produced $\mathbf{z}$.

### *B. Description and properties of the DUDE*

We start with some definitions that formalize the usual notion of context. A *neighborhood* $\mathcal{S}$ is a finite subset of $\mathbb{Z}^2$ that does not contain the origin $(0,0)$ (referred to as the *center* of the neighborhood). As an example, the $3\times 3$ neighborhood referred to in Section I is $\mathcal{S} = (\{-1,0,1\}{\times}\{-1,0,1\}) \setminus \{(0,0)\}$. For $i \in \mathbb{Z}^2$, we denote by $\mathcal{S}{+}i$ the set $\{j{+}i \,|\, j \in \mathcal{S}\}$, and, by extension, we say that $i$ is its center. For an image $\mathbf{z}$ and $\mathcal{S}{+}i \subseteq V_{m\times n}$ we denote by $\mathcal{S}_i^{\mathbf{z}}$ a vector of dimension $|\mathcal{S}|$ over $\mathcal{A}$, indexed by the elements of $\mathcal{S}$, such that $\mathcal{S}_i^{\mathbf{z}}[j] = z_{i+j},\ j \in \mathcal{S}$. We refer to such vectors as $\mathcal{S}$*-contexts*, or simply *contexts* (with a known underlying neighborhood $\mathcal{S}$ implied), and say that $z_i$ *occurs in context* $\mathcal{S}_i^{\mathbf{z}}$ (recall that $i \notin \mathcal{S}{+}i$). For "border" indices $i$ such that $\mathcal{S}{+}i \not\subseteq V_{m\times n}$, the vector $\mathcal{S}_i^{\mathbf{z}}$ is also well defined by assuming, e.g., that the value of any "out of bound" sample is set to an arbitrary constant from $\mathcal{A}$.

For a neighborhood $\mathcal{S}$ and a generic context vector $\mathbf{s}$, we let $\mathbf{m}(\mathbf{z},\mathbf{s})$ denote the $M$-dimensional column vector whose components are

$$\mathbf{m}(\mathbf{z},\mathbf{s})[a] = \left|\{i \in V_{m\times n} : \mathcal{S}_i^{\mathbf{z}} = \mathbf{s}, z_i = a\}\right| , \quad a \in \mathcal{A}. \tag{1}$$

In words, $\mathbf{m}(\mathbf{z},\mathbf{s})[a]$ denotes the number of occurrences of the symbol $a$, in context $\mathbf{s}$, in the image $\mathbf{z}$.

We denote by $\mathbf{u} \odot \mathbf{v}$ the component-wise (Schur) product of the $M$-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$, namely, $(\mathbf{u} \odot \mathbf{v})[a] = \mathbf{u}[a]\mathbf{v}[a],\ 0 \le a \le M{-}1$. The transpose of a matrix (or vector) $A$ is denoted $A^T$, and if $A$ is a

For each index $i \in V_{m\times n}$:

1) Determine $\hat{P}_{\mathbf{z}}\,(\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$, the empirical distribution of *noisy* symbols $z_j$ whose context $\mathcal{S}_j^{\mathbf{z}}$ is identical to $\mathcal{S}_i^{\mathbf{z}}$.
2) From $\hat{P}_{\mathbf{z}}\,(\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$ and the noise transition matrix, estimate a distribution $\hat{P}_{\mathbf{x}}\,(\cdot\,|\,\mathcal{S}_i^{\mathbf{z}}, z_i)$ of *clean* symbols whose corresponding *noisy* symbol is $z_i$ with context $\mathcal{S}_i^{\mathbf{z}}$. An expression for the estimated distribution is given, up to normalization, in (3).
3) Using the loss matrix $\boldsymbol{\Lambda}$, produce a denoised value $\hat{\boldsymbol{\chi}} = \hat{\boldsymbol{\chi}}_{\mathcal{S}}^{m\times n}(\mathbf{z})[i]$ according to (2), so that the expectation of the loss $\Lambda(\hat{\boldsymbol{\chi}}, \hat{x}_i)$ with respect to the distribution $\hat{P}_{\mathbf{x}}\,(\cdot\,|\,\mathcal{S}_i^{\mathbf{z}}, z_i)$ estimated in Step 2 is minimized.

Fig. 1. Outline of the DUDE algorithm.

nonsingular matrix, we write $A^{-T}$ as shorthand for $(A^{-1})^T$. Finally, let $\boldsymbol{\lambda}_a$ and $\boldsymbol{\pi}_a$ denote the $a$-th columns of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Pi}$, respectively, for $a \in \mathcal{A}$.

We are now ready to define the basic DUDE. For a given neighborhood $\mathcal{S}$, the $m\times n$ fixed-neighborhood DUDE $\hat{\boldsymbol{\chi}}_{\mathcal{S}}^{m\times n}$ is defined by

$$\hat{\boldsymbol{\chi}}_{\mathcal{S}}^{m\times n}(\mathbf{z})[i] = \arg\min_{\xi\in\mathcal{A}} \boldsymbol{\lambda}_\xi^T \cdot \Big( \big(\boldsymbol{\Pi}^{-T}\mathbf{m}\,(\mathbf{z},\mathcal{S}_i^{\mathbf{z}})\big) \odot \boldsymbol{\pi}_{z_i} \Big), \quad i \in V_{m\times n}\ . \tag{2}$$

The $m \times n$ basic DUDE $\hat{\boldsymbol{\chi}}_{\mathsf{univ}}^{m\times n}$ is obtained by letting the size of the neighborhood $\mathcal{S}$ grow at a suitable rate with $m$ and $n$ (refer to [2] and [3] for details).

The intuition behind the denoising rule in the fixed-neighborhood DUDE (2) is as follows. After proper normalization, the vector $\mathbf{m}\,(\mathbf{z},\mathcal{S}_i^{\mathbf{z}})$ in (2) can be seen as an empirical estimate, $\hat{P}_{\mathbf{z}}(\cdot|\mathcal{S}_i^{\mathbf{z}})$, of the conditional distribution of a noisy sample given its context, and the vector $\boldsymbol{\Pi}^{-T}\mathbf{m}\,(\mathbf{z},\mathcal{S}_i^{\mathbf{z}})$ as an estimate of the empirical distribution $P_{\mathbf{x}}(\cdot|\mathcal{S}_i^{\mathbf{z}})$ of the underlying *clean* sample given the *noisy* context (we say that the multiplication by the matrix $\boldsymbol{\Pi}^{-T}$ performs the “channel inversion”). The vector

$$\big(\boldsymbol{\Pi}^{-T}\mathbf{m}\,(\mathbf{z},\mathcal{S}_i^{\mathbf{z}})\big) \odot \boldsymbol{\pi}_{z_i}\ , \tag{3}$$

in turn, can be interpreted, after normalization, as an estimate $\hat{P}_{\mathbf{x}}(x_i\,|\,\mathcal{S}_i^{\mathbf{z}})$ of the posterior distribution of the clean sample given the noisy context $\mathcal{S}_i^{\mathbf{z}}$ *and* the noisy sample $z_i$. The expression (2) corresponds to a loss-weighted maximum a posteriori estimate of $x_i$ with respect to $\hat{P}_{\mathbf{x}}(\cdot\,|\,\mathcal{S}_i^{\mathbf{z}}, z_i)$. In a sense, the expression in (3) combines two pieces of “advice” on the value of the clean symbol $x_i$. On one hand, the estimated conditional distribution $\hat{P}_{\mathbf{x}}(\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$ conveys information on what the clean symbol in position $i$ is likely to be, given what is observed in the same context in the rest of the noisy image, while on the other hand, the noisy sample $z_i$ itself conveys information on the likelihood of $x_i$ which is independent of the rest of the image, given the memoryless assumption on the noise. If the noise level is not too high, the advice of $z_i$ is given more weight, while in more noisy conditions, the advice of the context gains more weight. The algorithm is outlined in Figure 1.

The universality of the denoiser $\hat{\boldsymbol{\chi}}_{\mathsf{univ}}^{m\times n}$ has been shown in two settings. In the *stochastic* setting, the image $\mathbf{x}$ is assumed to be a sample of a spatially stationary and ergodic process. The results of [2], as extended to the two-dimensional case in [3], state that in the limit (as $\min(m,n) \to \infty$), almost surely (with respect to both the input and the channel probability laws), the DUDE loss does not exceed that of the *best* $m\times n$ denoiser. In the *semi-stochastic* setting, the input is assumed to be an individual image, not generated by any probabilistic source, while the channel is still assumed probabilistic. It is shown in this case that, almost surely (with respect to the channel probability law), the asymptotic loss of the DUDE is optimal among *sliding window denoisers* (see [2] and [3] for details). Here, the result holds independently for each individual image $\mathbf{x}$ (in particular, the competing denoisers could be designed with full knowledge of the pair of images $\mathbf{x}, \mathbf{z}$). Notice that most image denoisers used in practice are of the sliding-window type.

In addition to its theoretical properties, the DUDE is also practical (see [2] for an analysis showing linear running time and sub-linear working storage complexities). The algorithm, in both its one-dimensional and two-dimensional versions, has been implemented, tested, and shown to be very effective on binary images [2], [3], text [2], and large HTML code files [27]. In [28] and the current work, we enhance the basic scheme to make it effective on grayscale images.

## III. iDUDE: AN ENHANCED FRAMEWORK FOR GRAYSCALE IMAGE DENOISING

In this section, we describe some tools added to the DUDE framework to enable effective denoising of grayscale images. We shall refer to the enhanced framework as iDUDE. The enhancements are described here in generality covering a broad class of channels. Detailed implementations for specific channels are presented in Section IV.

### *A. Addressing the model cost problem*

Estimating empirical conditional distributions $P_{\mathbf{x}}(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$ of image samples given their (noisy) context is a crucial component of the DUDE algorithm. As mentioned in Section I, estimating these distributions by collecting sample counts for "raw" contexts $\mathcal{S}_i^{\mathbf{z}}$ is ineffective for images of practical size. To address this problem, we exploit our prior knowledge of the structure of the input data via a stochastic model $P_X(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{x}})$ in which contexts share and aggregate their information. This will allow us to learn additional information about the distribution of, say, $x_i$ given its context $\mathcal{S}_i^{\mathbf{z}}$, from occurrences of samples $z_j$, depending on how "close" $\mathcal{S}_i^{\mathbf{z}}$ is to $\mathcal{S}_j^{\mathbf{z}}$ in an appropriate sense. We will then use our estimate of $P_X(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{x}})$ as an estimate of $P_{\mathbf{x}}(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$, and apply the denoising rule. Expressed in a different mathematical language, this "shared learning" paradigm can be seen to be taken to the limit in [4], where every context contributes, in an appropriately weighted form, to the denoising of every location of the image.

For the targeted grayscale images, prior knowledge takes the form of assumptions of brightness continuity (or, in short, *smoothness*), statistical invariance under constant shifts in absolute brightness (*DC invariance*), and *symmetry*. Next, we discuss these assumptions, and how they translate to various modeling tools. The assumptions and tools apply to *clean* images (denoted as $\mathbf{x}$), and they will clearly break down in some cases of images affected by noise. We defer the discussion of how, nevertheless, the tools are used in the iDUDE framework to Subsection III-D. Until then, we ignore the effect of noise.

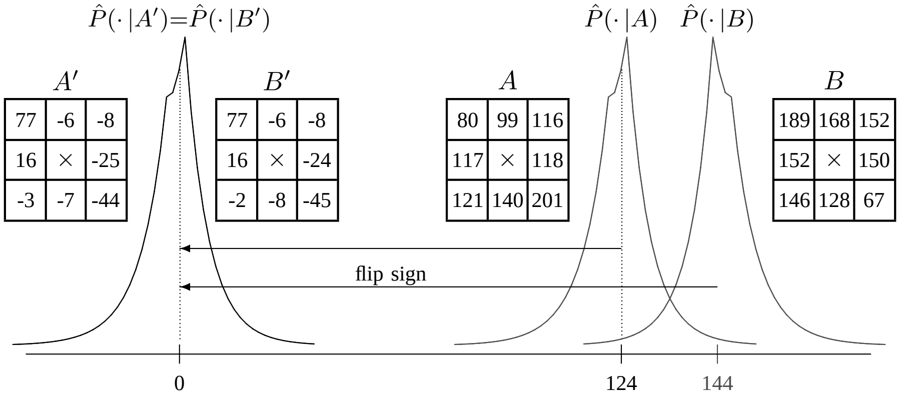
A1) *Smoothness.* By this property, contexts $\mathcal{S}_i^{\mathbf{x}}$ that are close as vectors will tend to produce similar conditional distributions for their center samples. Therefore, they can be clustered into *conditioning classes* of vectors that are "similar" in some sense, e.g., close in Euclidean space, and the conditional statistics of the member contexts can be aggregated into one conditional distribution for the class, possibly after some adjustment in the support of each distribution (see A2 below). There is a trade-off between the size of a conditioning class (or the total number of classes) and the accuracy of the merged distributions as approximations of the individual context-conditioned distributions. If classes are too large, they will include contexts with dissimilar associated conditional distributions, and the merged distribution will not be a good representative of the individual member distributions. If classes are too small, the associated merged statistics will be sparse, and they will not have faithfully captured the structure of the data. This is the well known trade-off in stochastic modeling which is at the core of the minimum description length (MDL) approach to statistical inference [29]. Algorithmic approaches to the optimization of the model size (number of classes) exist, and have been implemented successfully in lossless image compression [30]. However, simpler schemes based on carefully tuned but fixed models, such as those used in [5], [6] achieve similar levels of performance at a lower complexity cost. We will take the latter approach in our design of a context model for iDUDE. Although we have mentioned Euclidean distance between contexts (as vectors) as a natural measure of "closeness," similarities in other features may also be used, such as a measure of the *activity level* in the context (e.g., empirical variance), or a signature of the context's *texture* [6]. The use of these tools in iDUDE will be discussed concretely when we describe implementations in Section IV.

A2) *DC invariance* (i). Since similar contexts are expected to generate conditional statistics which are similar in shape but with slightly misaligned supports, merged conditional statistics generated as in A1 would be "blurred." This misalignment can be compensated for by using a *predictor* for the center sample of each context as a function of the context samples, and accumulating statistics of the prediction errors rather than the original sample values. It has long been known (see, e.g., [31]) that such prediction error distributions are peaked and centered near zero (Laplacian or generalized Gaussian models have proven very useful to model these distributions). When the merged distribution is used for a specific sample, e.g., in Step 2 of the procedure in Figure 1, the prediction error distribution should be re-centered at the predicted value for the sample, which can always be recovered from the sample's original context. The next item shows how the use of prediction allows for a broader notion of similarity between contexts.

A3) *DC invariance* (ii). Since contexts that differ only by a constant brightness level are likely to produce similar conditional distributions up to a shift in their support, statistics may be conditioned on *gradients* (differences between spatially close sample values) rather than the sample values themselves, so that conditional statistics of contexts that differ only by a constant intensity vector are merged. More specifically, if $\mathbf{s}$ is a context, $a{\in}\mathcal{A}$, and $\mathbf{a}$ denotes a constant vector with all entries equal to $a$, then

$$P(x_i = b \,|\, \mathbf{s}) \approx P(x_i = b + a \,|\, \mathbf{s} + \mathbf{a})\,, \quad b, b{+}a \in \mathcal{A}, \quad \mathbf{s}, \mathbf{s}{+}\mathbf{a} \in \mathcal{A}^{|\mathcal{S}|}\,, \tag{4}$$

where the $\approx$ sign denotes that we expect these probabilities to be "similar" in some sense appropriate to the application. Clearly, before they are merged, the conditional distributions must be shifted so that they are centered at a common point. This is accomplished by using prediction as described in A2. Also, when gradients are used to build contexts in lieu of the original sample values, the clustering described in A1 above is applied *after* the switch to gradient space. Notice that although the use of prediction described here and



$A'$

| | | |
|---|---|---|
| 77 | -6 | -8 |
| 16 | × | -25 |
| -3 | -7 | -44 |

$B'$

| | | |
|---|---|---|
| 77 | -6 | -8 |
| 16 | × | -24 |
| -2 | -8 | -45 |

$A$

| | | |
|---|---|---|
| 80 | 99 | 116 |
| 117 | × | 118 |
| 121 | 140 | 201 |

$B$

| | | |
|---|---|---|
| 189 | 168 | 152 |
| 152 | × | 150 |
| 146 | 128 | 67 |

Fig. 2. Merging of context conditional distributions.

the one described in A2 are related and derive from the same assumption, they are not equivalent. The use of prediction as mentioned in A2 is advantageous but optional when original samples are used to form the contexts, but it becomes mandatory if contexts are based on gradients.

A4) *Symmetry.* Patterns often repeat in different orientations, and statistics are not very sensitive to left/right, up/down, or black/white reflections.[3] Thus, contexts that become close as vectors after shape-preserving rotations or reflections of the underlying neighborhood pattern, or gradient sign changes (i.e., change in sign of all the differences mentioned in A3), will tend to produce similar conditional distributions, which can be merged as in A1–A3. To take advantage of these symmetries, contexts should be brought, by means of sign changes, and shape-preserving neighborhood rotations and reflections, to some canonical representation that uniquely represents the context's equivalence class under the allowed mappings (an example of such a canonical representation will be described below in Example 1). When bringing a context to canonical representation involves a gradient sign change, the support of the corresponding conditional distribution should be flipped around zero before merging with the other distributions in the conditioning class.

Clearly, the accuracy and appropriateness of the assumptions underlying A1–A4 will vary across images, or even across parts of the same image. Nevertheless, they have proven very useful in image compression and other image modeling applications. In particular, the use of a prediction function in the iDUDE framework allows for conditional distributions that would otherwise be considered different to "line-up" and be merged in a useful way. Thus, as in data compression, prediction is an important tool in model cost reduction [9] and the quality of the predictor affects the performance of the system. The better the predictor, the more skewed the distribution of prediction error values, which, in turn, will lead to a "sharper" selection of a likely reconstruction symbol as learned from the context (see the discussion following (2)).

*Example 1:* Figure 2 shows an example of the application of the tools described in A1–A4. Assume that $\mathcal{S}$ is the $3{\times}3$ neighborhood of samples closest to the center sample, and that the empirical distribution of this sample conditioned on each of the contexts labeled $A$ and $B$ is as illustrated on the right-hand side of Figure 2. We use the

[3] By black/white reflection invariance we mean that if $\mathbf{s}$ is a context vector, and $\mathbf{w}$ is a constant vector with all entries equal to the largest possible sample value, $M{-}1$, then we expect $P_{\mathbf{x}}(x\,|\,\mathbf{s}) \approx P_{\mathbf{x}}(M{-}1{-}x\,|\,\mathbf{w}{-}\mathbf{s})$.

average of each context, namely, $\mathrm{avg}(A) = 124$, and $\mathrm{avg}(B) = 144$, as a predictor. We then subtract the predicted value from each sample to obtain a *differential representation*, $\mathcal{D}(C)$, of each context $C$, as follows:

$$\mathcal{D}(A) = \begin{array}{|c|c|c|}\hline \text{-44} & \text{-25} & \text{-8} \\ \hline \text{-7} & \times & \text{-6} \\ \hline \text{-3} & 16 & 77 \\ \hline \end{array} \qquad \text{and} \quad \mathcal{D}(B) = \begin{array}{|c|c|c|}\hline 45 & 24 & 8 \\ \hline 8 & \times & 6 \\ \hline 2 & \text{-16} & \text{-77} \\ \hline \end{array}$$

We define the canonical representation of a context as one in which the upper left corner contains the largest entry, in absolute value, of the four corners of the context (this can always be arrived at by $90^\circ$ rotations), and the upper right corner contains the largest entry, again in absolute value, of the two corners on the secondary diagonal of the neighborhood (this can be achieved by a reflection, if needed, around the main diagonal after the initial rotation). Furthermore, we require the entry at the upper-left corner to be nonnegative, and we flip the sign of the context if this is not the case.[4] The array marked $A'$ in the figure shows the result of the above transformations on context $A$. For context $B$, the same transformations would result in the value $-77$ at the upper left corner. Therefore, we change the sign of all the entries in the context, resulting in the array labeled $B'$ in the figure. This sign change also means that prediction errors are accounted for in the merged histogram with their signs changed, or equivalently, that the original empirical distribution conditioned on $B$ is reflected around zero before merging. Finally, we observe that the canonical representations $A'$ and $B'$ are close in Euclidean distance, and we assume that they will be assigned to the same conditioning class. Thus, the distributions conditioned on $A'$ and $B'$ merge, resulting in the common distribution centered at zero represented on the left-hand side of the figure.

### *B. The formal model and its estimation from clean data*

In this subsection, we formalize the prediction-based model $P_X(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{x}})$ outlined in Subsection III-A for samples $x_i$ of an image $\mathbf{x}$ conditioned on their contexts $\mathcal{S}_i^{\mathbf{x}}$ for a *given* neighborhood $\mathcal{S}$ (which, as mentioned, we will not attempt to optimize). We first define the notation and terminology. Let $\tilde{x} : \mathcal{A}^{\mathcal{S}} \to \mathcal{A}$ denote a mapping that predicts a sample as a function of its context, and let $\mathcal{D} : \mathcal{A}^{\mathcal{S}} \to \mathbb{Z}^{\mathcal{S}}$ denote a function mapping a context to a differential representation (e.g., through the use of gradients) which is invariant under constant translations of the context components. Let $\mathcal{C} : \mathbb{Z}^{\mathcal{S}} \to \mathbb{Z}^{\mathcal{S}}$ denote a function that maps differential representations to a unique canonical representation by applying shape-preserving rotations and reflections to $\mathcal{S}$ (e.g., as described in Example 1).[5] Finally, let $\mathcal{Q} : \mathbb{Z}^{\mathcal{S}} \to \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_K\}$ , $K \geq 1$ , denote a classification function mapping canonical representations to a set of $K$ *conditioning classes*, or *clusters* (this function may be image-dependent). Abusing notation, we will also use $\mathcal{Q}$ to denote the composition of $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{Q}$, so that $\mathcal{Q}(\mathcal{S}_0)$ denotes the cluster corresponding to a context $\mathcal{S}_0$.

Our model of the image $\mathbf{x}$ is generated by conditional probability distributions $P_E(e\,|\,\mathcal{Q}_\kappa)$, of prediction error values $e$, $-M{+}1 \leq e \leq M{-}1$, associated with each conditioning class $\mathcal{Q}_\kappa$, $\kappa = 1, 2, \ldots, K$ (the above ranges of $\kappa$ and $e$ are implicitly assumed throughout the discussion). Under this model, a prediction error $e$ has probability $P_E(e\,|\,\mathcal{Q}_\kappa)$ and the corresponding conditional distribution $P_X(x\,|\,\mathcal{Q}_\kappa)$ of a sample $x$, $0 \leq x \leq M{-}1$, that occurs

[4] We omit a discussion of ambiguities and tie-breakers, which are easily handled so that the canonical representation is unique.

[5] To simplify notation, we will assume the canonical representation does not include sign changes; this technique is also easily implemented, cf. Example 1 and [5], [6].

in context $\mathcal{S}_0$, given its conditioning class $\mathcal{Q}_\kappa = \mathcal{Q}(\mathcal{S}_0)$, is implied by the relation

$$x = \max\left(0, \min(e + \tilde{x}(\mathcal{S}_0), M-1)\right).$$

In words, the conditional distribution $P_E(e \mid \mathcal{Q}_\kappa)$ is shifted by $\tilde{x}(\mathcal{S}_0)$ and the mass corresponding to negative values accumulates at $0$, whereas the mass corresponding to values larger than $M{-}1$ accumulates at $M{-}1$ (i.e., the signal "saturates" at the black and white levels).[6] This relation is more conveniently expressed in vector notation, by letting $\mathbf{u}_M^a$ denote an indicator (column) vector of length $M$, with a $1$ in position $a$, $0 \le a \le M{-}1$, and zeros elsewhere, and representing the observation of a sample $x$ as $\mathbf{u}_M^x$. Define the $M \times (2M{-}1)$ matrix

$$\mathbf{C}(a) = \left[\begin{array}{c|c|c} \overset{\xleftrightarrow{M-1-a}}{\begin{matrix}1 & 1 & \cdots & 1\\ 0 & 0 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 0\end{matrix}} & \quad \mathbf{I}_M \quad & \underset{\xleftrightarrow[a]{}}{\begin{matrix}0 & 0 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 0\\ 1 & 1 & \cdots & 1\end{matrix}}\end{array}\right], \quad a \in \mathcal{A}, \tag{5}$$

where $\mathbf{I}_k$ denotes an identity matrix of order $k$. With these definitions, the relation between $x$ and $e$ takes the form

$$\mathbf{u}_M^x = \mathbf{C}(\tilde{x}(\mathcal{S}_0))\mathbf{u}_{2M-1}^{e+M-1}. \tag{6}$$

Similarly, we will regard conditional probability distributions $P_U(\,\cdot \mid c)$ as (column) vectors $\mathbf{P}_U(c)$, indexed by the sample space of $P_U$.

With access to $\mathbf{x}$, the distribution $\mathbf{P}_E(\mathcal{Q}_\kappa)$ can be estimated from samples $x_i$ occurring in the context $\mathcal{S}_i^{\mathbf{x}}$ by selecting a suitable *estimation matrix* $\mathbf{M}(\tilde{x}_i)$, that depends on the predicted value $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{x}})$, and accumulating $\mathbf{M}(\tilde{x}_i)\,\mathbf{u}_M^{x_i}$ into a vector $\mathbf{e}_\kappa$ of dimension $2M{-}1$. The role of $\mathbf{M}(\tilde{x}_i)$ is to map the $M$-dimensional indicator vector $\mathbf{u}_M^{x_i}$ into a vector of the same dimension as the desired estimate. Specifically, letting $\mathcal{Q}_i^{\mathbf{x}} \triangleq \mathcal{Q}(\mathcal{S}_i^{\mathbf{x}})$, the estimate

$$\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa) = \left(\sum_{i:\mathcal{Q}_i^{\mathbf{x}}=\mathcal{Q}_\kappa} \mathbf{M}(\tilde{x}_i)\mathbf{C}(\tilde{x}_i)\right)^{-1} \left(\sum_{i:\mathcal{Q}_i^{\mathbf{x}}=\mathcal{Q}_\kappa} \mathbf{M}(\tilde{x}_i)\,\mathbf{u}_M^{x_i}\right) \triangleq \mathbf{R}\cdot\left(\sum_{i:\mathcal{Q}_i^{\mathbf{x}}=\mathcal{Q}_\kappa} \mathbf{M}(\tilde{x}_i)\,\mathbf{u}_M^{x_i}\right) \tag{7}$$

where the matrix $\mathbf{R}$ acts as a normalization factor, is in fact unbiased for any choice of the estimation matrices $\mathbf{M}(\tilde{x}_i)$ that leads to a well-defined $\mathbf{R}$.[7] This property is readily seen by replacing $x = x_i$ in (6), pre-multiplying each side of (6) by $\mathbf{M}(\tilde{x}_i)$, summing both sides over all the indexes $i$ such that $\mathcal{Q}_i^{\mathbf{x}} = \mathcal{Q}_\kappa$, and noting that the expectation of $\mathbf{u}_{2M-1}^{e+M-1}$ under $P_E(\,\cdot \mid \mathcal{Q}_\kappa)$ is $\mathbf{P}_E(\mathcal{Q}_\kappa)$. A natural choice for $\mathbf{M}(\tilde{x}_i)$ is the matrix $\mathbf{S}_{\tilde{x}_i}$, where

$$\mathbf{S}_a \triangleq \left[\mathbf{0}_{M\times(M-1-a)}\ |\mathbf{I}_M|\ \mathbf{0}_{M\times a}\right]^T, \qquad a \in \mathcal{A}, \tag{8}$$

with $\mathbf{0}_{j\times k}$ representing a $j{\times}k$ zero matrix. This choice corresponds to incrementing the entry $x_i - \tilde{x}_i$ of $\mathbf{e}_\kappa$ by one

[6] Our implementation uses this saturation model for simplicity. Other, more sophisticated models are possible.

[7] If necessary, pseudo-inverse techniques can be used, as discussed in Subsection III-F. However, as will become clear later in this subsection, the invertibility problem will not arise for our choice of estimate.

```
1) Initialization. Initialize to zero a histogram, $\mathbf{e}_\kappa$, of prediction error residual occurrences for each context cluster $\mathcal{Q}_\kappa,\ 1 \le \kappa \le K$.
2) Statistics collection. For each index $i \in V_{m\times n}$:
    a) Set $\tilde{x}_i = \tilde{x}(\mathcal{S}_i^{\mathbf{y}})$, the predicted value for $x_i$.
    b) Set $\bar{\mathcal{S}}_i^{\mathbf{y}} = \mathcal{D}(\mathcal{S}_i^{\mathbf{y}})$, the differential representation of $\mathcal{S}_i^{\mathbf{y}}$.
    c) Set $\mathcal{C}_i^{\mathbf{y}} = \mathcal{C}(\bar{\mathcal{S}}_i^{\mathbf{y}})$, the canonical representation of $\bar{\mathcal{S}}_i^{\mathbf{y}}$.
    d) Set $\mathcal{Q}_i^{\mathbf{y}} = \mathcal{Q}(\mathcal{C}_i^{\mathbf{y}})$, the conditioning class of $\mathcal{S}_i^{\mathbf{y}}$.
    e) Set $\mathbf{e}_\kappa \leftarrow \mathbf{e}_\kappa + \mathbf{M}'(\tilde{x}_i)\,\mathbf{u}_M^{z_i}$ for $\kappa$ such that $\mathcal{Q}_\kappa = \mathcal{Q}_i^{\mathbf{y}}$.
3) Normalization. For each $\kappa$, normalize $\mathbf{e}_\kappa$ to obtain $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$.
4) Conditional distributions for individual contexts. For each index $i \in V_{m\times n}$:
    a) Set $\tilde{x}_i$, $\mathcal{S}_i^{\mathbf{y}}$, and $\mathcal{Q}_i^{\mathbf{y}}$ as in Step 2 above.
    b) Set $\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}}) = \mathbf{C}(\tilde{x}_i)\,\hat{\mathbf{P}}_E(\mathcal{Q}_i^{\mathbf{y}})$ .
```

Fig. 3. Estimation of conditional distributions based on prediction and context classification.

for index $i$: the observation of $x_i$ gives the observer a sample from the "window" $[-\tilde{x}_i, M{-}1{-}\tilde{x}_i]$ (of size $M$) of the support (of size $2M{-}1$) of $P_E(\,\cdot\,|\,\mathcal{Q}_\kappa)$.

The normalization by $\mathbf{R}$ differs from the natural choice of (uniformly) normalizing by the sum $\sum_e \mathbf{e}_\kappa[e]$. This difference accounts for two factors: first, the saturation in the model (6), and second, the fact that the number of times a given entry $e$ of $\mathbf{e}_\kappa$ has an opportunity to be incremented, denoted $n_e$, depends on the number of predicted values $\tilde{x}_i$ such that $e$ falls in the window $-\tilde{x}_i \le e \le M{-}1{-}\tilde{x}_i$. Notice, however, that the variance of the ratio $\mathbf{e}_\kappa[e]/n_e$ is, under reasonable assumptions, inversely proportional to $n_e$. Therefore, small values of $n_e$ will produce estimates of high variance for the corresponding entry of $\hat{P}_E(\,\cdot\,|\,\mathcal{Q}_\kappa)$ and, hence, uniform normalization has the effect of producing estimates with a more uniform variance, which is a desirable property. Consequently, we will replace $\mathbf{R}$ with a diagonal matrix effecting uniform normalization and use the resulting estimate for $\hat{P}_E(\,\cdot\,|\,\mathcal{Q}_\kappa)$.

With the estimated distribution $\hat{P}_E(\,\cdot\,|\,\mathcal{Q}_i^{\mathbf{x}})$ in hand, the corresponding estimate of $P_X(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{x}})$ based on $\mathbf{x}$ is given by the vector

$$\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{x}}) = \mathbf{C}(\tilde{x}_i)\,\hat{\mathbf{P}}_E(\mathcal{Q}_i^{\mathbf{x}})\,,\ \ i \in V_{m\times n}\,. \tag{9}$$

The overall modeling procedure is outlined in Figure 3 where, in preparation for the situation in which the model is estimated from noisy data, we have decoupled three images that so far have been folded into $\mathbf{x}$: the noisy input image $\mathbf{z}$, an *available* image $\mathbf{y}$ (which will be derived from $\mathbf{z}$), and the clean image $\mathbf{x}$. Thus, contexts are denoted $\mathcal{S}_i^{\mathbf{y}}$ and are formed from $\mathbf{y}$, and the update of $\mathbf{e}_\kappa$ uses the observed sample $z_i$ (rather than the unavailable value $x_i$), appropriately replacing $\mathbf{M}(\tilde{x}_i)$ with a matrix $\mathbf{M}'(\tilde{x}_i)$ to be introduced in Subsection III-D. The case of estimating the model from clean data corresponds to $\mathbf{z} = \mathbf{y} = \mathbf{x}$.

In Figure 3, as in the preceding discussion, we have assumed, for simplicity, that the prediction function $\tilde{x}$ is fixed, in the sense that its value depends only on the sample values of the context it is applied to. The actual procedure used in iDUDE is enhanced with the addition of an adaptive component to the prediction function, that depends also on image statistics associated to the context, and a two-stage context clustering strategy. We discuss these enhancements next.

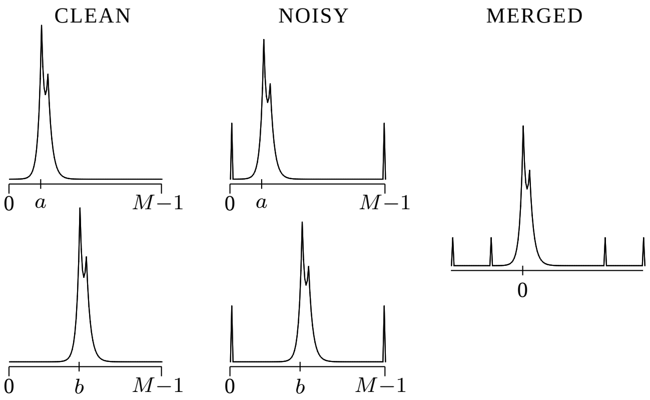


Fig. 4. Effect of S&P noise on merging of similarly-shaped distributions centered at different values.

### C. Two-stage modeling

It has been observed (see, e.g., [5]) that conditional distributions of prediction errors produced by a fixed predictor exhibit context-dependent biases. To improve prediction accuracy, a *bias cancellation* component is used in conjunction with the fixed predictor. To derive this component, contexts are clustered in two stages.

Let $\tilde{x}$ be a fixed predictor, as discussed in Subsection III-B. We assume that a first-stage classification function $\mathcal{R} : \mathbb{Z}^{\mathcal{S}} \rightarrow \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_J\},\ \ J \geq 1$, mapping canonical representations to prediction clusters (or classes), is defined. Let $\mathcal{I}_\ell$ denote the set of sample indices $j$ such that $\mathcal{R}_\ell = \mathcal{R}(\mathcal{S}^{\mathbf{x}}_j)$ (where, again, we abuse the notation for $\mathcal{R}$), and let $n_\ell = |\mathcal{I}_\ell|$. For each cluster $\mathcal{R}_\ell$, $1 \leq \ell \leq J$, we compute a bias correction value that will be applied to samples in $\mathcal{I}_\ell$ as

$$\varepsilon_\ell = \frac{1}{n_\ell} \sum_{j \in \mathcal{I}_\ell} (x_j - \tilde{x}(\mathcal{S}^{\mathbf{x}}_j))\,,\ \ \ell \in \{1, 2, \ldots, J\}\,. \tag{10}$$

The final predicted value for $x_i$, $i \in \mathcal{I}_{\ell_0}$, is then given by $\hat{x}_i = [\tilde{x}_i + \varepsilon_{\ell_0}]$, where $[v]$ denotes the integer closest to $v$. Due to this rounding operation, rounding to an integer is no longer necessary in the fixed prediction function $\tilde{x}$. Therefore, we reinterpret this function as one mapping contexts to real numbers, while the refined predictor can be seen as an integer-valued function $\hat{x}(\mathcal{S}^{\mathbf{x}}_i)$ that depends on the samples in $\mathcal{S}^{\mathbf{x}}_i$, and also on the image $\mathbf{x}$ through the bias value estimated for $\mathcal{R}(\mathcal{S}^{\mathbf{x}}_i)$.

After applying the bias correction, statistics for the corrected prediction errors are collected in a (generally) coarser set of context clusters, i.e., the clusters $\mathcal{R}_\ell$ are *re-clustered* into the set of conditioning classes $\{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_K\}$, where each class $\mathcal{Q}_\kappa$ , $1 \leq \kappa \leq K$ , merges samples from several clusters $\mathcal{R}_\ell$. Hereafter, we interpret the modeling procedure in Figure 3 as using the prediction function $\hat{x}$ and corresponding prediction values $\hat{x}_i$ throughout in lieu of $\tilde{x}$ and $\tilde{x}_i$, respectively.

### D. Model estimation in the presence of noise

The discussion in Subsections III-A–III-C has focused on the modeling of a clean image, ignoring the effect of noise. To illustrate the effect of applying the preceding modeling assumptions to noisy data to estimate $\hat{\mathbf{P}}_{\mathbf{z}}(\mathcal{S}^{\mathbf{z}}_i)$, consider, for example, a S&P channel. In this channel, a fraction $\delta$ of the samples are saturated to black (sample value $0$) or white (sample value $M-1$) with equal probability. Clearly, noisy samples in this case generally will not obey smoothness or DC-invariance assumptions. This affects both the samples whose distributions we are modeling, and the contexts that condition these distributions. On the one hand, contexts that are similar (and could be clustered) in the clean image will generally not remain so in the noisy image. On the other hand, distributions that have similar shapes up to translation in the clean image may not remain so, as they may be differently positioned with respect to the spikes at $0$ and $M-1$ caused by the noise. The latter effect is illustrated in Figure 4, where it is clear that, since the merged statistics are not typical of a S&P channel output, application of the channel inversion procedure and denoising as in Figure 1 will not remove the noise.

Although the effect of noise may be more benign for other channels, given the above discussion, our approach will not be to apply the preceding modeling assumptions to the noisy data. Rather, we will translate the operations required by the tools to a "cleaner" domain, where the working assumptions are more likely to hold. To this end, we will further assume that the formal model of Subsection III-B (for the clean samples) still applies when the (unavailable) clean context is replaced by the corresponding one in a *pre-filtered* image $\mathbf{y}$. The image $\mathbf{y}$ is available for model estimation and can be obtained as the result of a "rough" denoising of the noisy image $\mathbf{z}$ using a (possibly simpler) denoiser appropriate for the type of noise (e.g., a median filter for S&P noise). In the iterative process described in Subsection III-E, $\mathbf{y}$ can also be the output of a previous iteration of iDUDE. Thus, we will postulate that $\mathcal{S}_i^{\mathbf{y}}$ can replace $\mathcal{S}_i^{\mathbf{x}}$ in the modeling of the $i$–th sample, as described in Figure 3. Our rationale for this assumption is based on the fact that $\tilde{x}(\mathcal{S}_i^{\mathbf{y}})$ is still a good predictor of $x_i$, and therefore an effective model with few conditioning classes, via context aggregation, can be built from $\mathbf{y}$. The image $\mathbf{y}$ is also used for bias cancellation, with $y_i$ replacing $x_i$ in the bias calculation (10). For zero-mean, additive noise, we could use the noisy samples $z_i$, since the effect of noise will tend to cancel. However, such a strategy would fail for non-additive noise.

It should be noted that pre-filtering introduces some dependence of contexts $\mathcal{S}_i^{\mathbf{y}}$ on their noisy center samples $z_i$, since the value of $z_i$ might have participated in the rough denoising of some of the components of $\mathcal{S}_i^{\mathbf{y}}$. This "contamination" is undesirable since, by virtue of the independence assumptions on the channel, in the denoising rule (2), the information on $z_i$ is fully incorporated in $\boldsymbol{\pi}_{z_i}$ to produce, via the Schur product, the correct overall clean symbol likelihoods used by the rule. However, it turns out that practical heuristics will allow us to detect when this dependence is strong enough to negatively impact the performance of the denoising algorithm and act accordingly (see Subsection IV-B). Pre-filtering can also be seen as a tool for capturing higher-order dependencies without increasing model cost. Clearly, with conditioning classes based on a pre-filtered image, the conditioning events for the original noisy samples depend on a larger number of original samples than the size of the neighborhood used. Thus, pre-filtering increases the effective size of the contexts used to condition the distributions, without increasing the number of conditioning classes.

Our task is now to estimate the above model for *clean* image samples, conditioned on contexts formed from an available image $\mathbf{y}$ (which is obtained from the noisy image $\mathbf{z}$). To this end we follow the procedure of Figure 3, but take into consideration the fact that we have access to $\mathbf{z}$, rather than to the clean image $\mathbf{x}$. Notice that while our goal coincides with the main step in the baseline DUDE algorithm, namely to produce an estimate of the posterior distribution of the clean symbol $x_i$ given the noisy context $\mathcal{S}_i^{\mathbf{z}}$ and the noisy symbol $z_i$, we will accomplish it directly, without going through the intermediate step of estimating distributions of noisy samples.

To see how the model is estimated from noisy data using the DUDE approach, we revisit the derivation in Subsection III-B. When the image being sampled is noisy, each sample $z_i$ provides information about the $(2M-1)$-vector $\mathbf{P}_E(\mathcal{Q}_\kappa)$, subject to the same arbitrary shifts and saturation as before (see Equation (6)), but also to noise. Now, recall from the discussion following (2) that, in DUDE, the channel inversion is accomplished by pre-multiplication by the matrix $\mathbf{\Pi}^{-T}$. Thus, just as an occurrence of $x_i$ contributes $\mathbf{M}(\hat{x}_i)\mathbf{u}_M^{x_i}$ to the estimate in (7) (when based on clean data), an occurrence of $z_i$ can be seen as contributing $\mathbf{M}(\hat{x}_i)\mathbf{\Pi}^{-T}\mathbf{u}_M^{z_i}$. This motivates the DUDE-like estimate (for the prediction setting)

$$\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa) = \mathbf{R}' \cdot \left( \sum_i \mathbf{M}(\hat{x}_i)\, \mathbf{\Pi}^{-T}\, \mathbf{u}_M^{z_i} \right) \tag{11}$$

where $\mathbf{R}'$ is a normalization matrix. It can be shown that if $\mathbf{R}'$ is set to $\mathbf{R}$ as in (7) (with $\hat{x}_i$ in lieu of $\tilde{x}_i$), then (11) becomes an unbiased estimate of $\mathbf{P}_E(\mathcal{Q}_\kappa)$. Replacing again $\mathbf{R}'$ with a uniform normalization, it follows that the procedure in Figure 3 applies, with

$$\mathbf{M}'(\hat{x}_i) = \mathbf{S}_{\hat{x}_i}\, \mathbf{\Pi}^{-T} \tag{12}$$

and $\mathbf{S}_{\hat{x}_i}$ as defined in (8). At first sight, with this choice, the update of $\mathbf{e}_\kappa$ in Figure 3 involves $M$ operations per image sample. However, as we shall see in Section IV, for the channels of interest, this procedure can be implemented with one scalar increment to a histogram of prediction errors per sample, followed by adjustments whose complexity is independent of the image size.

The above estimate can be interpreted as follows. Define $\mathcal{Q}_{\kappa,p} = \mathcal{Q}_\kappa \cap \{\mathcal{S}_i^{\mathbf{y}} \mid \hat{x}(\mathcal{S}_i^{\mathbf{y}}) = p\}$, referred to as a *sub-cluster*. For a cluster $\mathcal{Q}_\kappa$, the result of Step 2 of the procedure in Figure 3 (with the choice (12) for $\mathbf{M}'(\hat{x}_i)$) can be written as

$$\begin{aligned} \mathbf{e}_\kappa &= \sum_{i:\mathcal{S}_i^{\mathbf{y}} \in \mathcal{Q}_\kappa} \mathbf{S}_{\hat{x}_i}\, \mathbf{\Pi}^{-T}\mathbf{u}_M^{z_i} = \sum_{p\in\mathcal{A}} \sum_{i:\, \mathcal{S}_i^{\mathbf{y}} \in \mathcal{Q}_{\kappa,p}} \mathbf{S}_p \mathbf{\Pi}^{-T}\mathbf{u}_M^{z_i} \\ &= \sum_{p\in\mathcal{A}} \mathbf{S}_p \mathbf{\Pi}^{-T} \sum_{i:\, \mathcal{S}_i^{\mathbf{y}} \in \mathcal{Q}_{\kappa,p}} \mathbf{u}_M^{z_i} = \sum_{p\in\mathcal{A}} \mathbf{S}_p \left( \mathbf{\Pi}^{-T}\mathbf{m}_{\kappa,p} \right) \end{aligned} \tag{13}$$

where $\mathbf{m}_{\kappa,p}$ denotes a vector of occurrence counts of noisy symbols in the sub-cluster $\mathcal{Q}_{\kappa,p}$. The expression $\mathbf{\Pi}^{-T}\mathbf{m}_{\kappa,p}$ in the sum on the right-hand side of (13) represents an estimate of the empirical distribution $\hat{P}_{\mathbf{x}}(\, a \mid \mathcal{Q}_{\kappa,p})$ of samples $a \in \mathcal{A}$ conditioned on the sub-cluster $\mathcal{Q}_{\kappa,p}$, where the multiplication by $\mathbf{\Pi}^{-T}$ performs the "channel inversion." Shifted by $\mathbf{S}_p$, it becomes a conditional distribution of prediction errors. Equation (13) says that our estimate follows along the lines of the basic DUDE, except that it does so for the sub-clusters $\mathcal{Q}_{\kappa,p}$. The distributions of prediction errors for clean symbols obtained for the sub-clusters are merged to yield $\hat{P}_E(\,\cdot \mid \mathcal{Q}_\kappa)$, and the estimated conditional distribution of $x_i$ given $\mathcal{S}_i^{\mathbf{y}}$ is given by Equation (9). Notice, however, that the goodness of this estimate does not rely on the law of large numbers "kicking in" for *each* sub-cluster, but rather for each cluster.

In general, the matrix $\mathbf{M}'(\hat{x}_i)$ in (12) may have negative entries, which may place the estimate $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$ obtained in Step 3 of Figure 3 outside the probability simplex. This situation reflects statistical fluctuations and is more likely to occur if the sample size is not large enough. The estimate is then modified as follows. Let $p_e$ denote the entries

- Set $\mathbf{y} = \mathcal{F}(\mathbf{z})$, a pre-filtered version of the noisy input image $\mathbf{z}$.
- Repeat until $\mathbf{y}$ satisfies the stopping criterion:
  - Construct a set $\mathcal{Q}^{\mathbf{y}}$ of $K$ conditioning classes, and apply the procedure of Figure 3 to estimate the conditional distributions $\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}})$, $i \in V_{m\times n}$.
  - Denoise $\mathbf{z}$ using the rule (15) with the conditional distributions derived in the previous step, and let $\mathbf{y}$ denote the resulting denoised image.

Fig. 5. Iterative denoising with pre-filtering.

of $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$, $-M{+}1 \le e \le M{-}1$, and, for real numbers $a, b$, let $(a-b)^+$ denote $a-b$ if $a > b$, or $0$ otherwise. Consider a real number $\gamma$, $0{\le}\gamma{\le}1$. Since $\sum_e p_e = 1$, there exists a real number $\mu_\gamma$ such that

$$\sum_{e=-M+1}^{M-1} (p_e - \mu_\gamma)^+ = \gamma\,. \tag{14}$$

It is not difficult to verify that if $\gamma = 1$, the vector with entries $p'_e = (p_e - \mu_\gamma)^+$ represents the point on the probability simplex that is closest in $L_2$ distance to $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$. The transformation from $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$ to the vector of entries $p'_e$ can be seen as a "smoothing" of $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$, which clips its negative entries, if any, and possibly also some of the small positive ones. Choosing $\gamma < 1$ and renormalizing effects a more aggressive smoothing of the tails of the distribution, which was found to be useful in practice to obtain more robust denoising performance. We refer to this operation as a *regularization* of the estimated distribution $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$, and include it as part of Step 3 in the procedure of Figure 3.

Finally, the corresponding estimate $\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}})$ obtained in Step 4b of the procedure of Figure 3 is used to compute the estimated posterior

$$\hat{\mathbf{P}}_{\mathbf{x}}(\mathcal{S}_i^{\mathbf{y}}, z_i) = \hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}}) \odot \boldsymbol{\pi}_{z_i}$$

employed by the DUDE rule (see Equation (3)). The rule (2) then takes the form

$$\hat{\boldsymbol{\chi}}_{\mathcal{S}}^{m\times n}(\mathbf{z})[i] = \arg\min_{\xi\in\mathcal{A}} \boldsymbol{\lambda}_\xi^T \cdot \hat{\mathbf{P}}_{\mathbf{x}}(\mathcal{S}_i^{\mathbf{y}}, z_i), \quad i \in V_{m\times n}\,. \tag{15}$$

### *E. Iterative process*

The process of using a pre-filtered image $\mathbf{y}$ for the purpose of context formation can be repeated *iteratively*, using the iDUDE output from one iteration as the input for the next, starting from some "roughly denoised" image. The iterations tend to improve the quality of the context $\mathcal{S}_i^{\mathbf{y}}$ and increase the effective size of the neighborhoods, as discussed. The iterative procedure can be stopped after a fixed number of iterations, provided that a method for detecting undesirable "contamination" of the contexts with the values of their center samples is used (see Subsection IV-C). The iterative procedure is summarized in Figure 5, where we denote by $\mathcal{Q}^{\mathbf{y}}$ the set of conditioning classes derived from an image $\mathbf{y}$. It is important to notice that, in each iteration, while the prediction classes $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_J\}$ and the predictions $\hat{x}$ are computed from pre-filtered samples from $\mathbf{y}$, the statistics used for estimating the cluster-conditioned distributions used in the actual denoising (the vectors $\mathbf{e}_\kappa$) are derived from the original noisy samples in $\mathbf{z}$.

### *F. Channel matrix inversion*

iDUDE, as the original DUDE, relies on computing the inverse of the channel transition matrix $\mathbf{\Pi}$ to estimate the posterior distributions used in the denoising decisions. Although $\mathbf{\Pi}$ is formally non-singular for the channels we consider, it is very badly conditioned in some important cases, and, most notably, in the Gaussian case. Notice, however, that the choice of estimation matrices $\mathbf{M}(\hat{x}_i)$ in (11) is arbitrary, and that a different choice, that would formally cancel $\mathbf{\Pi}^{-T}$, may alleviate the problem. Another approach for these channels is to proceed as in the derivations of (7) and (11), but perform the channel inversion by solving for the conditional distributions $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$ with a numerical procedure to minimize a function of the form $||\mathbf{U} - \mathbf{V} \cdot \hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)||$ (up to numerical tolerances and stability), subject to the constraint that $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$ represent valid probability distributions, where $|| \cdot ||$ denotes some norm on $(2M{-}1)$–vectors,

$$\mathbf{U} = \sum_i \mathbf{M}(\hat{x}_i)\, \mathbf{u}_M^{z_i}\ , \qquad \mathbf{V} = \sum_i \mathbf{M}(\hat{x}_i)\, \mathbf{\Pi}^T\, \mathbf{C}(\hat{x}_i)$$

and the sums are over all occurrences of $\mathcal{Q}_\kappa$. The matrices $\mathbf{M}(\hat{x}_i) = \mathbf{S}_{\hat{x}_i}$ are again a natural but arbitrary choice, and can be replaced with a suitable set of estimation matrices that would result in a better numerical behavior.

Maximum-likelihood estimation of $\mathbf{P}_E(\mathcal{Q}_\kappa)$ is also possible, at a higher computational cost. This approach becomes attractive when both the noise process and the conditional distributions $\mathbf{P}_E(\mathcal{Q}_\kappa)$ admit simple parametric models; we illustrate it by describing its application in the (quantized) Gaussian noise case. As mentioned, context-conditioned distributions of prediction errors for clean natural images are well modeled by a discrete Laplacian [31], [32], which is parametrized by a decay factor $\theta$ and a mean $\mu$ (not necessarily an integer value). Denoting $\mu' = \lceil \mu \rceil$, a prediction error $e$ is assigned, under this model, probability $C(\theta, \mu)\theta^{e-\mu'}$ if $e \geq \mu'$ and $(1-\theta-C(\theta,\mu))\theta^{\mu'-e-1}$ otherwise, where the coefficient $C(\theta, \mu)$ is such that the mean of the distribution equals $\mu$. We assume this model for the difference $(x_i - \hat{x}(\mathcal{S}_i^{\mathbf{x}}))$, conditioned on the cluster of $\mathcal{S}_i^{\mathbf{y}}$, where we recall that $\hat{x}(\mathcal{S}_i^{\mathbf{x}})$ is the (unobserved) predicted value for $x_i$ that would have been obtained by applying the predictor on the clean image $\mathbf{x}$. To estimate the unknown, cluster-dependent parameters $\theta, \mu$ from the data, we first notice that

$$z_i - \hat{x}(\mathcal{S}_i^{\mathbf{y}}) = (z_i - x_i) + (x_i - \hat{x}(\mathcal{S}_i^{\mathbf{x}})) + (\hat{x}(\mathcal{S}_i^{\mathbf{x}}) - \hat{x}(\mathcal{S}_i^{\mathbf{y}}))\,. \tag{16}$$

Assuming, for simplicity, that the prediction function is an average of $k$ samples, $(\hat{x}(\mathcal{S}_i^{\mathbf{x}}) - \hat{x}(\mathcal{S}_i^{\mathbf{z}}))$ is well modeled by a zero-mean normal random variable with variance $\sigma^2/k$. While $\hat{x}(\mathcal{S}_i^{\mathbf{y}})$ is a better approximation to $\hat{x}(\mathcal{S}_i^{\mathbf{x}})$, we adopt this normal model also for $(\hat{x}(\mathcal{S}_i^{\mathbf{x}}) - \hat{x}(\mathcal{S}_i^{\mathbf{y}}))$. Thus, conditioned on $\mathcal{S}_i^{\mathbf{y}}$, the left-hand side of (16) can be modeled as the convolution of a zero-mean normal distribution with variance $\sigma^2(1+k^{-1})$ and a Laplacian. We refer to such a convolution as a *LG distribution*, or LG$(\theta, \mu, \varsigma^2)$, with $\varsigma^2$ denoting the variance of the normal distribution participating in the convolution; in the foregoing example, $\varsigma^2 = \sigma^2(1+k^{-1})$. Explicit formulas for the probability mass function of a discrete LG distribution and its derivatives with respect to the parameters $\theta, \mu$ and $\varsigma$ can be derived in terms of the error function erf$(\cdot)$. Although these expressions are rather unwieldy, they lend themselves to numerical computations, and therefore allow for a numerical maximum-likelihood estimation of the parameters $\theta$ and $\mu$ ($\varsigma$ is assumed given) from the statistics $z_i - \hat{x}(\mathcal{S}_i^{\mathbf{y}})$ collected for the conditioning class cluster

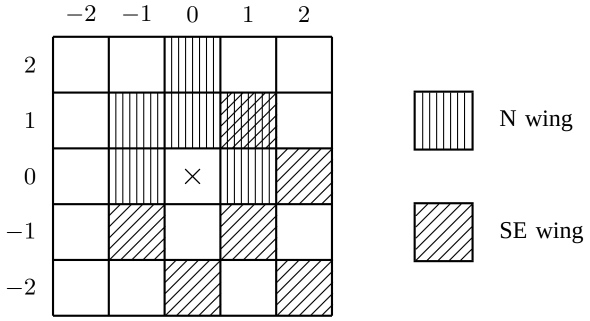


Fig. 6. Neighborhood for the WGT modeling scheme.

of $\mathcal{S}_i^{\mathbf{y}}$. With these estimated parameters on hand, we write

$$x_i = (x_i - \hat{x}(\mathcal{S}_i^{\mathbf{x}})) + (\hat{x}(\mathcal{S}_i^{\mathbf{x}}) - \hat{x}(\mathcal{S}_i^{\mathbf{y}})) + \hat{x}(\mathcal{S}_i^{\mathbf{y}})$$

and estimate $\mathbf{P}_X(\mathcal{S}_i^{\mathbf{y}})$ to be a LG$(\hat{\theta}, \hat{\mu}, \sigma^2/k)$ centered at $\hat{x}(\mathcal{S}_i^{\mathbf{y}})$, where $\hat{\theta}$ and $\hat{\mu}$ are the estimated Laplacian parameters. This derivation extends to cases where other linear or piecewise-linear predictors are used, with appropriate adjustments of the constant $k$ above. For more complex predictors, the parameter $\varsigma$ can be estimated together with the other parameters, under the constraint that $\varsigma \geq \sigma$. The estimate $\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}})$, in turn, would be a LG$(\hat{\theta}, \hat{\mu}, \varsigma^2 - \sigma^2)$. In our implementation we use a simplified version of this procedure, described in Subsection IV-F.

Aside from providing an alternative to the channel matrix inversion, this parametric approach has model cost advantages, since only two parameters, $\theta$ and $\mu$, need to be estimated per conditioning class [32], as opposed to $M-1$ parameters when individual probabilities for each symbol are estimated.

## IV. Implementation for various noise types

In this section, we describe implementations of the iDUDE framework for three types of noise, namely, S&P noise, $M$-ary symmetric noise (which leaves a sample intact with a certain probability $1-\delta$, or replaces it with a uniformly distributed random value from the complement of the alphabet with probability $\delta$), and quantized additive white Gaussian noise. We assume that the Euclidean ($L_2$) norm is used for the loss function $\Lambda$ in all cases (other norms are easily implemented by suitably adapting the optimization (15)). In all cases, we follow the flow of the DUDE algorithm, with model estimation as outlined in Figure 3. We begin by describing components that are common to more than one channel, and then discuss the specifics of the implementation for each channel.

### A. Prediction and context classification

Our context model is based on the $5\times5$ neighborhood $\mathcal{S}$ shown in Figure 6. We describe a predictor and a quantization scheme that map a generic context $\mathcal{S}_i^{\mathbf{y}}$ from an image $\mathbf{y}$ into a fixed prediction value $\tilde{x}(\mathcal{S}_i^{\mathbf{y}})$, a conditioning class $\mathcal{Q}(\mathcal{S}_i^{\mathbf{y}})$, and a prediction class $\mathcal{R}(\mathcal{S}_i^{\mathbf{y}})$. The predictor and quantizer draw from ideas in [6] and [5] to classify contexts by computing a context signature derived from gradients as well as a bitmap reflecting the context's "texture." For ease of reference, we will refer to both the predictor and the context quantizer as WGT (*wing gradients and texture*).

We denote by $y_{(a,b)}$ the value of the sample in coordinate $(a, b)$ of the neighborhood in Figure 6, with $-2 \leq a, b \leq 2$. As the neighborhood slides accross the image, the actual coordinates of the context samples are $i{+}(a, b),\ i{\in}V_{m\times n}$; for clutter reduction, we omit the center coordinate $i$ in this discussion and in the Appendix. A context is brought to canonical form via rotations and reflections as described in Example 1, with "entry at the upper-left corner" interpreted as the sum $y_{(-2,2)} + y_{(-1,2)} + y_{(-1,1)} + y_{(-2,1)}$, and analogously for the other corners. Context signs are not implemented.

Once the context is in canonical form, it is decomposed into eight (overlapping) *wings*: four horizontal/vertical wings labeled $N$, $S$, $E$ and $W$, and four diagonal wings labeled $NE$, $SE$, $NW$ and $SW$. Referring to Figure 6, the $N$ wing consists of the samples with coordinates $(-1, 0)$, $(1, 0)$, $(-1, 1)$, $(0, 1)$, $(1, 1)$, and $(0, 2)$. The $S, E$, and $W$ wings are defined similarly, following appropriate $90^\circ$ rotations. As for the diagonals, the $SE$ wing is formed by the samples with coordinates $(1, 1)$, $(-1, -1)$, $(2, 0)$, $(1, -1)$, $(0, -2)$, and $(2, -2)$, with the $NW$, $SE$, $SW$ being formed by appropriate $90^\circ$ rotations. For each wing, we compute a sample average and a directional gradient. The fixed predictor $\tilde{x}$ is computed as a nonlinear weighted function of the wing averages and gradient magnitudes, with more weight given to wings with lower gradient magnitudes. The goal is to emphasize parts of the context that are "smooth" (i.e., of low gradient), and de-emphasize parts that might be crossed by sharp edges. The precise details of the computation are given in the Appendix.

Gradient magnitudes computed for prediction are also used to derive an integer-valued *activity level*, $A(\mathcal{S}_i^{\mathbf{y}})$, for each context, as also described in detail in the Appendix. Conditioning classes are obtained by quantizing $A(\mathcal{S}_i^{\mathbf{y}})$ into $K$ regions, such that the induced classes are of approximately the same cardinality. To form the prediction classes, the activity level classification is refined by computing a representation of the *texture* of the context. This representation takes the form of a bitmap with one bit per context sample; the bit is set to $0$ if the corresponding sample value is smaller than the value $\tilde{x}(\mathcal{S}_i^{\mathbf{y}})$ predicted by the fixed predictor, or to $1$ otherwise [6].

The classification of the contexts into prediction classes is accomplished by computing a context signature combining the activity level and the first $T$ bits from the texture bitmap, $T \geq 0$, taken in order of increasing distance from the center. Thus, the number of prediction classes is $J = K\,2^T$. Notice that since the activity level of a context is derived from differences (gradients) between sample values, and the texture map from comparisons with a predicted value, the resulting context classification is DC-invariant.

### B. Choosing denoiser parameters without access to the clean image

In practice, the optimal settings of various iDUDE parameters, such as the number of prediction and conditioning classes, or the number of iterations in the procedure of Figure 5, may vary from image to image. The most obvious difficulty in choosing image-dependent settings is that denoising performance cannot be measured directly, since the clean image is not available to the denoiser. Thus, we have no direct way of telling whether one setting is better or worse than another. Nevertheless, various methods for choosing the best parameters for the DUDE have proven effective in practice, and can be used also for iDUDE. Some of these methods are based on using an observable parameter that correlates with denoising performance, and optimizing the settings based on the observable. An example of such a heuristic, described in [2], suggests using the *compressibility* of the denoised sequence. More principled techniques, based on an unbiased estimate of the DUDE loss, are described in [33].

In our implementations, we have grouped images by size (”very small”, ”small”, ”large”), and by noise level for each channel, and have chosen one set of parameters for each size/channel/noise level combination. The choices, which are fairly robust, were guided by performance on an available set of training images, and also by some basic guidelines on context models: larger images can sustain larger models, and so do cleaner images (intuitively, since less can be learned from noisy data than from clean data). The specific parameter values are given in Table II of Section V.

### *C. Monitoring of the statistical model during iteration*

As mentioned in Subsection III-E, the iDUDE iteration of Figure 5 introduces dependencies between contexts $\mathcal{S}_i^{\mathbf{y}}$ and their noisy center samples $z_i$, since the value of $z_i$ might have participated (directly or indirectly) in the rough denoising of some of the components of $\mathcal{S}_i^{\mathbf{y}}$. We have observed empirically that these dependencies can cause significant deviations from the expected behavior of the statistical model, which, in turn, can translate to a deterioration of the denoising performance after a number of iterations. To prevent this effect, we employ a heuristic that is particularly useful for the non-additive channels.

The heuristic monitors the fraction of potentially noisy samples in each conditioning class, and verifies that the fraction is consistent with the channel parameters. To determine whether $z_i{=}c$ is noisy given that $\mathcal{Q}(\mathcal{S}_i^{\mathbf{y}}){=}\mathcal{Q}_\kappa$, we measure the fraction of times $c$ occurs in $\mathcal{Q}_\kappa$ and $\hat{x}(\mathcal{S}_i^{\mathbf{y}}) \in \mathcal{A}_c^{\mathsf{far}}$, where $\mathcal{A}_c^{\mathsf{far}}$ is the subset of $M'$ values in $\mathcal{A}$ that are farthest away from $c$ (the exact value of $M'$ is not critical; $M' = M/2$ has worked well in our experiments).

The rationale of the heuristic is that, due to the smoothness of images, $x_i = c$ is unlikely if $\hat{x}(\mathcal{S}_i^{\mathbf{y}}) \in \mathcal{A}_c^{\mathsf{far}}$, so the measured frequency of occurrence of $c$ is a good estimate of its probability due to noise in cluster $\mathcal{Q}_\kappa$. This estimate can then be compared against the probability of $z_i{=}c$ due to noise on the channel at hand (i.e., $\delta/2$ in the S&P case, where only $c{=}0$ and $c{=}M{-}1$ are potential noisy values, and $\delta/(M{-}1)$ in the $M$-ary symmetric case, where a corrupted sample can assume any value from $\mathcal{A}$). Assuming the conditioning class is sufficiently populated, a significant deviation of the count from its expected value (measured, say, in multiples of its standard deviation) is strong evidence for the violation of the statistical assumptions of the denoiser. When such a situation is detected, the iDUDE will refrain from making corrections for samples in the affected class, and will leave the value from the pre-filtered image untouched, while samples in “healthier” classes will continue to be refined in the iterative procedure. A threshold of ten to fifteen standard deviations has proven effective in our experiments.

Figure 7 illustrates the effectiveness of the heuristic. The figure plots the PSNR of the denoised image as a function of the number of iterations for one of the S&P denoising experiments of Section V. When the heuristic is not used, there is a large drop in PSNR in the fifth iteration. The drop is prevented when the heuristic is used, and the PSNR follows a concave curve that stabilizes after a few iterations, making the choice of stopping point for the iteration far less critical.

In more generality, when all the off-diagonal entries in each column of the channel matrix $\mathbf{\Pi}$ are equal, which

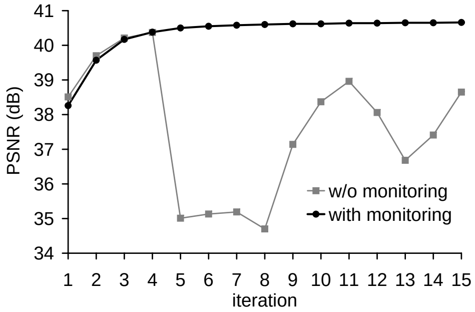


Fig. 7. Effect of statistics monitoring on the iDUDE iteration performance (S&P noise).

is the case for the two non-additive channels studied here, the probability of $z_i = c$ given $x_i \neq c$ (and the cluster $\mathcal{Q}_\kappa$) is clearly the common off-diagonal value in column $c$. For other channels, it may be possible to obtain useful bounds that still allow for meaningful detection of deviations from the expected noise behavior.

Notice that during the first application of the iDUDE, the procedure above can be used to *estimate* the channel parameters, rather than compare against them. Thus, the assumption of known channel parameters is not essential in these cases.

### D. Implementation for Salt and Pepper noise

The channel transition matrix for S&P noise, and its inverse, are given by

$$\mathbf{\Pi}_{\mathsf{sp}}(\delta) = \begin{bmatrix} 1-\frac{\delta}{2} & 0 & \cdots & 0 & \frac{\delta}{2} \\ \frac{\delta}{2} & 1-\delta & \cdots & 0 & \frac{\delta}{2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\delta}{2} & 0 & \cdots & 1-\delta & \frac{\delta}{2} \\ \frac{\delta}{2} & 0 & \cdots & 0 & 1-\frac{\delta}{2} \end{bmatrix}, \quad \mathbf{\Pi}_{\mathsf{sp}}^{-1}(\delta) = \frac{1}{1-\delta} \begin{bmatrix} 1-\frac{\delta}{2} & 0 & \cdots & 0 & -\frac{\delta}{2} \\ -\frac{\delta}{2} & 1 & \cdots & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots & -\frac{\delta}{2} \\ -\frac{\delta}{2} & 0 & \cdots & 1 & -\frac{\delta}{2} \\ -\frac{\delta}{2} & 0 & \cdots & 0 & 1-\frac{\delta}{2} \end{bmatrix}, \quad 0 \leq \delta < 1. \qquad (17)$$

The matrices are well conditioned, except when $\delta$ approaches one.[8]

*1) Pre-filtering:* The iDUDE implementation for the S&P channel uses a pre-filter $\mathcal{F}$ based on a *modified selective median* (MSM) filter for the first step of the procedure of Figure 5. The filter is applied only to samples valued $0$ and $M-1$. It estimates the sample at the center of a $5\times 5$ window by computing the median of a set of $25$ values, namely, the non-center sample values in the window and their average. The pre-filter is improved by running the MSM filter iteratively (still within the first step in Figure 5), using the MSM output of one iteration as the input to the next, and refining the estimate for the samples valued $0$ and $M-1$ in the *original* noisy image. This iteration generally stabilizes, and can be stopped when the $L_2$ distance between the outputs of one iteration and the next falls below a certain threshold (which is not very critical). We refer to this improved pre-filter as an *iterated* MSM, or, in short, IMSM. The improvement of IMSM over MSM is illustrated in Table III and Figure 8 of Section V.

[8] We report on the symmetric case for simplicity. Asymmetric cases where the probability of switching to $0$ or $M-1$ are not necessarily equal are easily handled by adjusting the matrices in (17) accordingly.

The output from the IMSM pre-filter is used as input to the first application of the iDUDE in the second stage of the procedure of Figure 5.

*2) Prediction and context model:* The WGT predictor and context model are used.

*3) Model estimation:* With the matrix $\mathbf{\Pi}_{\mathsf{sp}}^{-1}$ of (17), the update in Step 2e of Figure 3 (with $\mathbf{M}'(\hat{x}_i)$ defined in (12)) consists of adding $(1-\delta)^{-1}$ to $\mathbf{e}_\kappa[z_i-\hat{x}_i]$, and subtracting $\frac{\delta}{2(1-\delta)}$ from $\mathbf{e}_\kappa[-\hat{x}_i]$ and $\mathbf{e}_\kappa[M-1-\hat{x}_i]$. Notice that the latter two subtractions depend on the predicted value $\hat{x}_i$, but not on $z_i$. Thus, the computation of the statistic $\mathbf{e}_\kappa$ can be implemented by just maintaining, for each conditioning class $\mathcal{Q}_\kappa$, a conventional histogram of occurrences of differences $z_i-\hat{x}_i$, together with a histogram of predicted values $\hat{x}_i$, each requiring one scalar increment per sample. After scanning the image in the first pass of the iDUDE, the counts in the two histograms suffice to derive $\mathbf{e}_\kappa$.

*4) Denoising rule:* For the $L_2$ norm, ignoring integer constraints, the minimum in the iDUDE decision rule (15) is attained by the expectation, $\bar{\xi}$, of $x$ under $\hat{P}_{\mathbf{x}}(x\,|\,\mathcal{S}_i^{\mathbf{y}}, z_i)$. For $z_i=0$, writing $\hat{\mathbf{P}}_{\mathbf{x}}(\mathcal{S}_i^{\mathbf{y}}, z_i)$ explicitly as $\gamma\,\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}})\odot\boldsymbol{\pi}_0$, where $\gamma$ is an appropriate normalization coefficient, and substituting the first column of $\mathbf{\Pi}_{\mathsf{sp}}(\delta)$ from (17) for $\boldsymbol{\pi}_0$, we obtain

$$\bar{\xi}_i = \frac{\delta E_{\mathbf{x}}}{2(1-\delta)p_0+\delta}\,,$$

where $E_{\mathbf{x}}$ is the expectation of $x$ under $\hat{P}_X(x\,|\,\mathcal{S}_i^{\mathbf{y}})$ and $p_0=\hat{P}_X(\,0\,|\,\mathcal{S}_i^{\mathbf{y}})$. The reconstructed value for $x_i$ is obtained by rounding $\bar{\xi}_i$ to the nearest integer (which gives the precise integer solution to (15)). An analogous formula can be derived for the case when $z_i=M-1$.

### *E. Implementation for the $M$-ary symmetric channel*

The $M$-ary symmetric channel is defined by a transition probability matrix $\mathbf{\Pi}_{\mathsf{M}}(\delta)$, with entries

$$(\mathbf{\Pi}_{\mathsf{M}})_{a,b} \;=\; \begin{cases} 1-\delta, & b=a\in\mathcal{A}, \\ \frac{\delta}{M-1}, & b\in\mathcal{A}\setminus\{a\}\,. \end{cases} \tag{18}$$

This matrix is generally well-conditioned (except near $\delta=(M-1)/M$), and its inverse $\mathbf{\Pi}_{\mathsf{M}}^{-1}(\delta)$ is given by

$$(\mathbf{\Pi}_{\mathsf{M}})_{a,b}^{-1} = \begin{cases} \frac{M-\delta-1}{(1-\delta)M-1}, & b=a\in\mathcal{A}, \\ -\frac{\delta}{(1-\delta)M-1}, & b\in\mathcal{A}\setminus\{a\}\,. \end{cases} \tag{19}$$

*1) Pre-filtering:* The MSM filter described in Subsection IV-D.1, without iteration, is used for the first step of the procedure in Figure 5.

*2) Prediction and context model:* The WGT predictor and context model are used.

*3) Model estimation:* It follows from (19) that the column with index $a$ in $\mathbf{\Pi}_{\mathsf{M}}^{-T}$ can be written in the form

$$\frac{M-1}{(1-\delta)M-1}\,\mathbf{u}_M^a - \frac{\delta}{(1-\delta)M-1}\,\mathbf{1}_M\,,\quad a\in\mathcal{A}, \tag{20}$$

where $\mathbf{u}_M^a$ is an indicator vector as defined in Section III, and $\mathbf{1}_M$ is an all-one column of dimension $M$. Thus, to implement the update in Step 2e of Figure 3 in the case of the $M$-ary symmetric channel it suffices, again, to maintain a conventional histogram of occurrences of differences $z_i - \hat{x}_i$, together with a histogram of predicted values $\hat{x}_i$, from which the statistic $\mathbf{e}_\kappa$ is obtained at the end of the first pass of the iDUDE over the image.

*4) Denoising rule:* With the entries of $\mathbf{\Pi}_{\mathsf{M}}$ given in (18), the computation of the expectation of $\xi$ under $\hat{P}_{\mathbf{x}}(\,\xi\,|\,\mathcal{S}_i^{\mathbf{z}}, z_i)$ for the $M$-ary symmetric channel yields

$$\bar{\xi}_i = \frac{\delta E_{\mathbf{x}} + ((1-\delta)M - 1)\, p_{z_i} z_i}{\delta + ((1-\delta)M - 1)\, p_{z_i}}\ ,$$

where $E_{\mathbf{x}}$ is defined as before, and $p_{z_i} = \hat{P}_X(\,z_i\,|\,\mathcal{S}_i^{\mathbf{z}})$. The iDUDE estimate for $x_i$ is the integer closest to $\bar{\xi}_i$.

### *F. Implementation for Gaussian noise*

We consider the quantized additive white Gaussian channel, where real-valued noise $\eta_i \sim \mathcal{N}(0, \sigma^2)$ is added (independently) to each clean symbol $x_i$ to produce $\zeta_i = x_i + \eta_i$, the observed output $z_i$ being the value closest to $\zeta_i$ in $\mathcal{A}$. The entries of the channel transition matrix $\mathbf{\Pi}_{\mathsf{G}}$ are readily derived from these definitions in terms of the error function erf$(\,\cdot\,)$.

*1) Pre-filtering:* No pre-filter is used. The iteration of Figure 5 includes at most two applications of the iDUDE, using the identity for $\mathcal{F}$. In principle, pre-filtering and iteration are optional for the Gaussian channel, since an image affected by Gaussian noise can still be seen as satisfying our assumptions A1–A4 on grayscale images, and therefore these assumptions could be used for modeling $\hat{P}_{\mathbf{z}}(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{z}})$. This is reflected in our results in Section V, where we do not use pre-filtering or iteration for the high SNR regime. When we do use one round of iteration in the low SNR regime, the gains are relatively modest. Now, since our results for this channel are preliminary (as will be discussed in Section V), a state-of-the-art denoiser for Gaussian noise such as the one in [14], used as a pre-filter, would have resulted in improved performance. However, the use of such a pre-filter would not reflect the spirit of the (lightweight) rough denoising step.

*2) Prediction and context model:* Two variants of the iDUDE framework were implemented. Both use the WGT predictor of Subsection IV-A. The first variant uses also the WGT context model. This variant is fast, and performs well in the high SNR regime.

In the second variant, contexts $\mathcal{S}_i^{\mathbf{y}}$ are first brought to differential canonical form $\mathcal{C}(\mathcal{S}_i^{\mathbf{y}})$ (see Figure 3). Taking the $\mathcal{C}(\mathcal{S}_i^{\mathbf{y}})$ as 24-dimensional real vectors, the contexts are initially classified into $N$ clusters $V_1, V_2, \ldots, V_N$ by means of the Linde-Buzo-Gray (LBG) vector quantization algorithm [34], with the $L_2$ metric used to measure distance between contexts. The activity level of a context $\mathcal{S}_i^{\mathbf{y}}$ is defined in this case as $\log \hat{\sigma}_i^2$, where $\hat{\sigma}_i^2$ is the empirical variance of samples in the context. Conditioning classes $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_K$ are defined by uniformly quantizing the activity level. The set of prediction classes is then defined as $\{\,\mathcal{Q}_i \cap V_j\,|\,1 \le i \le K,\ 1 \le j \le N\,\}$, namely, a total of $J = K \cdot N$ classes. The LBG variant of the context model is slower, but performs better, and is the preferred mode of operation, at lower SNR.

*3) Model estimation:* We follow the parametric approach outlined in Subsection III-F, but with a simpler estimation procedure for the cluster-dependent parameters $\theta$ and $\mu$ of the (discrete) Laplacian component of the LG model for $P_{\mathbf{x}}(\,\cdot\,|\,\mathcal{S}_i^{\mathbf{y}})$. First, denoting the variance of the Laplacian by $\tau^2$, we observe that by the definition of the LG model, its variance $\nu^2$ is given by $\nu^2 = \tau^2 + \sigma^2(1 + k^{-1})$. Given the parameters of the Laplacian, $\tau^2$ takes the form

$$\tau^2 = \frac{2\theta}{(1-\theta)^2} + r(1-r) \tag{21}$$

where $r$ denotes the fractional part of $\mu$. In the first pass of the iDUDE we compute the empirical mean, $\hat{\mu}_\kappa$, and variance, $\hat{\nu}_\kappa^2$, of the differences $z_i - \hat{x}(\mathcal{S}_i^{\mathbf{y}})$ observed in each class $\mathcal{Q}_\kappa$. Next, we estimate the variance $\tau_\kappa^2$ of the Laplacian component for $\mathcal{Q}_\kappa$ as

$$\hat{\tau}_\kappa^2 = \max\left(\hat{r}_\kappa(1-\hat{r}_\kappa), \hat{\nu}_\kappa^2 - \sigma^2(1+k^{-1})\right) \tag{22}$$

where $\hat{r}_\kappa$ denotes the fractional part of $\hat{\mu}_\kappa$ and we recall that $k$ is a parameter that accounts for the number of samples participating in the weighted average in the WGT predictor (we use $k=5$). The maximum in (22) accounts for the fact that an estimate $\hat{\nu}_\kappa^2 - \sigma^2(1+k^{-1})$ for the variance could be smaller than the minimum possible variance $\hat{r}_\kappa(1-\hat{r}_\kappa)$ of the discrete Laplacian (obtained for $\theta = 0$, see (21)), due to statistical fluctuations or an inaccurate choice of the parameter $k$. Finally, given $\hat{\mu}_\kappa$ and $\hat{\tau}_\kappa^2$, we use (21) to solve for an estimate $\hat{\theta}_\kappa$.

## V. Results

In this section, we present results obtained with the iDUDE on images corrupted by simulated S&P, $M$-ary, and Gaussian noise. For each type of noise, we compare our results with those of a sample of recent denoising algorithms from the literature for which an objective basis for comparison was available. Our iDUDE experiments are based on a research prototype implementation written in C++, and run on a vintage 2007 Intel-based personal computer.[9] For a very rough complexity reference, we measured the running time of one iDUDE iteration in this implementation (using the WGT context model) on the $2048 \times 2560$ image Bike at approximately 7 seconds, for a throughput of approximately 730 Kpixels/sec. Running times for a given context model do not vary significantly with the noise type or level.

The images used in the experiments are listed in Table I. The "very small" heading in the table refers to a set of $24$ images of dimensions $384{\times}256$ (referred to as $\mathrm{Set}_{24}$) available at [35], for which results of denoising with the state-of-the-art scheme of [26] at various levels of S&P noise are available. The "small" $(512 \times 512)$ images in the table are from the set traditionally used in the image processing literature.[10] Since the images in either set are rather small by today's standards, we include also larger images from the benchmark set used in the development of the JPEG-LS standard [5].

We evaluate denoising performance by measuring Peak Signal to Noise Ratio (PSNR) between the denoised image and the original clean image. Table II summarizes the iteration and model size parameters used for the various experiments and noise types. The parameters, and the general iDUDE configuration for each noise type, were defined in Section IV. We use one set of iteration and model size parameters for each combination of image size category, noise type, and noise level, rather than parameters optimized for each individual image. The fixed predictor parameters $g$ and $\alpha$ (cf. Appendix) were set as follows: $g = 8\%$ of maximum gradient magnitude in the

[9] Specifically, Intel(R) Xeon(R) 5160 CPU, 3 GHz clock speed, 3 GB RAM, running Linux.

[10] We use the versions available at the DenoiseLab site [36]. Additionally, to allow comparison with [26] also on a $512{\times}512$ image, we use the (different) version of the Lena image reported on in [26], which we refer to as Lena$*$. We are not aware of other images for which a reliable comparison with [26] is possible.

| very small images | | | small images | | | large images | | |
|---|---|---|---|---|---|---|---|---|
| image | size | source | image | size | source | image | size | source |
| Set of 24 images ($\mathrm{Set}_{24}$) | $384\times256$ | [35] | Lena | $512\times512$ | TR | Tools | $1524\times1200$ | $\mathrm{JLS}^{\mathrm{Y}}$ |
| | | | Lena* | $512\times512$ | [26] | Toolsk | $1524\times1200$ | $\mathrm{JLS}^{\mathrm{K}}$ |
| | | | Boat | $512\times512$ | TR | Womank | $2048\times2560$ | $\mathrm{JLS}^{\mathrm{K}}$ |
| | | | Barbara | $512\times512$ | TR | Bike | $2048\times2560$ | $\mathrm{JLS}^{\mathrm{Y}}$ |

TABLE I

IMAGES USED IN THE EXPERIMENTS. LEGEND: TR: TRADITIONAL IMAGES; JLS: IMAGES FROM THE JPEG-LS BENCHMARK SET; Y: Y CHANNEL OF YCRCB COLOR SPACE; K: K CHANNEL OF CMYK COLOR SPACE.

| S&P | | | | | | | | | | $M$-ary symmetric | | | | | | | Gaussian | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | LBG | | | | WGT | |
| | v. small | | | small | | | large | | | | small | | | large | | | | | small | | large | | all | |
| $\delta$ | $R$ | $K$ | $T$ | $R$ | $K$ | $T$ | $R$ | $K$ | $T$ | $\delta$ | $R$ | $K$ | $T$ | $R$ | $K$ | $T$ | $\sigma$ | $R$ | $K$ | $N$ | $K$ | $N$ | $K$ | $T$ |
| 10% | 10 | 4 | 8 | 10 | 4 | 14 | 15 | 32 | 16 | 10% | 15 | 4 | 14 | 15 | 8 | 16 | 5 | 1 | 32 | 256 | 96 | 256 | 32 | 6 |
| 30% | 10 | 4 | 8 | 10 | 4 | 14 | 15 | 32 | 16 | 20% | 15 | 4 | 14 | | | | 20 | 2 | 32 | 192 | 32 | 192 | | |
| 50% | 10 | 4 | 8 | 10 | 4 | 14 | 20 | 32 | 16 | 30% | 15 | 4 | 10 | 15 | 8 | 16 | | | | | | | | |
| 70% | 20 | 4 | 8 | 20 | 4 | 14 | 20 | 32 | 14 | 40% | 20 | 4 | 9 | | | | | | | | | | | |
| | | | | | | | | | | 50% | 20 | 4 | 8 | 20 | 16 | 8 | | | | | | | | |

TABLE II

PARAMETERS USED IN THE EXPERIMENTS. $R$: NUMBER OF iDUDE ITERATIONS; $K$, $T$: MODEL SIZE PARAMETERS (CF. SEC. IV-A); $N$: NUMBER OF LBG CLUSTERS (SEC. IV-F.2).

context, $\alpha = 0.075$ for the S&P channel, $\alpha = 0.1$ for the $M$-ary symmetric channel; for the Gaussian channel, $g$ and $\alpha$ were optimized to minimize the observable prediction RMSE for each noisy image, with $g$ varying between $5\%$ and $17\%$, and $\alpha$ between $0$ and $0.05$. This is one case where it is "legitimate" to optimize the parameter for each image, since the optimization is based on observable data.

### *A. S&P noise*

The traditional test images (e.g., Boat, Barbara, Lena), contain very few, if any, pure black (value 0) or pure white (value $M-1$) samples. Therefore, for these image, the S&P channel behaves like an erasure channel, and noisy samples are easily identified. We include the images Toolsk and Womank to test the iDUDE in a more challenging situation. These images have significant amounts of pure black and white pixels, both in large solid regions, and in isolated occurrences scattered across the image.

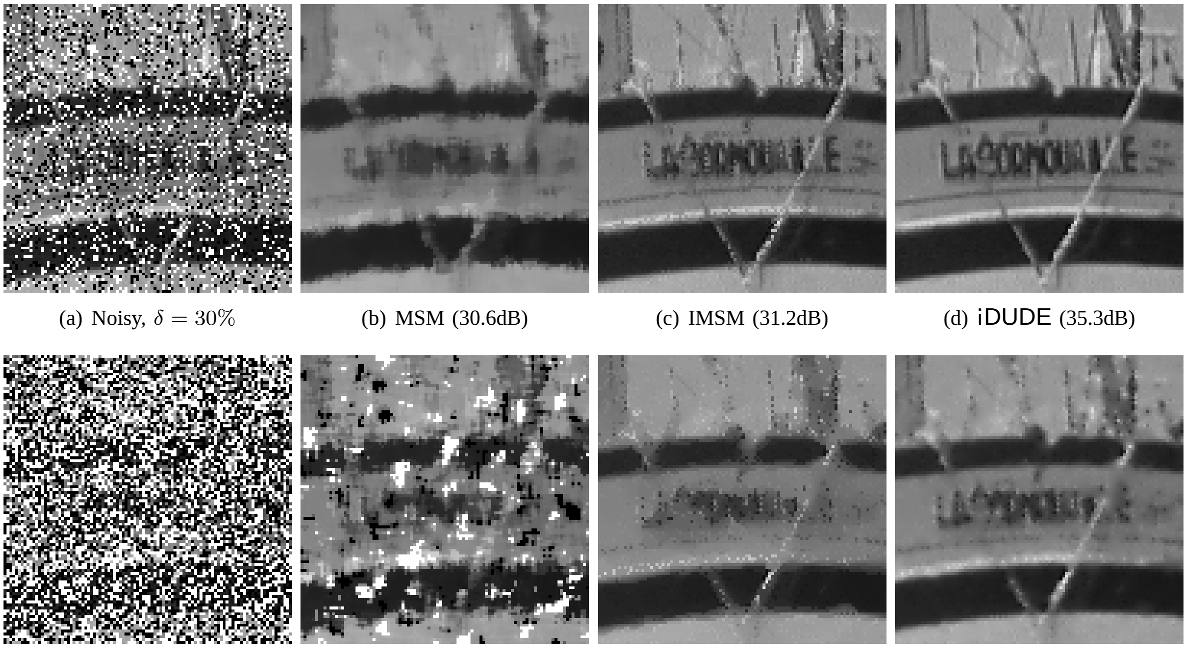
Table III summarizes the results for the S&P channel. Visual examples are given in Figure 8. For this channel, we compare our results to those of [26] on the Lena* variant of the Lena image, and on the mentioned $\mathrm{Set}_{24}$ from [35]. For the latter, for brevity, we list the *average* PSNR over the set images (as done also for the results reported in [26]). In all cases, we compare also with the modified selective median (MSM) filter described in Section IV-D.1, and its iterated version (IMSM). The results show iDUDE outperforming [26] in all cases, and by significant margins in the case of the Lena* image. The advantage of iDUDE diminishes as images become very small and noise levels become high, as expected from a statistical context-model-based scheme.

| | $\delta = 10\%$ | | | $\delta = 30\%$ | | | $\delta = 50\%$ | | | $\delta = 70\%$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| image | MSM | IMSM | iDUDE | MSM | IMSM | iDUDE | MSM | IMSM | iDUDE | MSM | IMSM | iDUDE |
| Lena | 40.1 | 40.4 | 45.2 | 34.1 | 35.2 | 39.7 | 27.4 | 32.0 | 36.3 | 16.7 | 29.1 | 32.8 |
| Boat | 36.3 | 36.5 | 41.0 | 30.6 | 31.2 | 35.3 | 25.5 | 28.3 | 32.0 | 16.4 | 25.7 | 28.9 |
| Barbara | 32.6 | 33.0 | 38.7 | 27.4 | 28.3 | 31.7 | 23.4 | 26.0 | 27.7 | 15.8 | 24.2 | 24.7 |
| Tools | 25.6 | 25.2 | 31.8 | 22.1 | 22.2 | 26.9 | 19.2 | 20.1 | 23.5 | 14.1 | 18.5 | 20.6 |
| Toolsk | 27.1 | 26.8 | 31.0 | 23.6 | 23.8 | 26.4 | 20.0 | 21.7 | 23.6 | 12.9 | 20.2 | 21.2 |
| Womank | 34.0 | 33.9 | 40.7 | 30.0 | 30.3 | 34.9 | 24.6 | 27.9 | 31.2 | 14.3 | 26.1 | 28.1 |
| Bike | 31.2 | 31.3 | 39.4 | 26.5 | 27.4 | 33.1 | 22.4 | 24.7 | 29.0 | 15.0 | 22.1 | 25.1 |

| image/<br>$\delta$ | MSM | IMSM | CHN05 | iDUDE |
|---|---|---|---|---|
| $\mathrm{Set}_{24}$ | | | | |
| 10% | 36.3 | 36.5 | 40.4 | 40.9 |
| 30% | 30.6 | 31.4 | 34.5 | 35.1 |
| 50% | 25.0 | 28.4 | 31.1 | 31.6 |
| 70% | 15.8 | 25.9 | 28.1 | 28.6 |
| Lena* | | | | |
| 10% | 38.9 | 39.2 | 42.3 | 44.8 |
| 30% | 32.9 | 33.9 | 35.6 | 38.8 |
| 50% | 26.4 | 30.8 | 32.3 | 35.4 |
| 70% | 16.1 | 28.0 | 29.3 | 31.7 |

TABLE III

Results for S&P noise. MSM: modified selective median (cf. Section IV-D.1); IMSM: iterated MSM; CHN05: the denoiser of [26]. Comparison with CHN05 displayed separately.



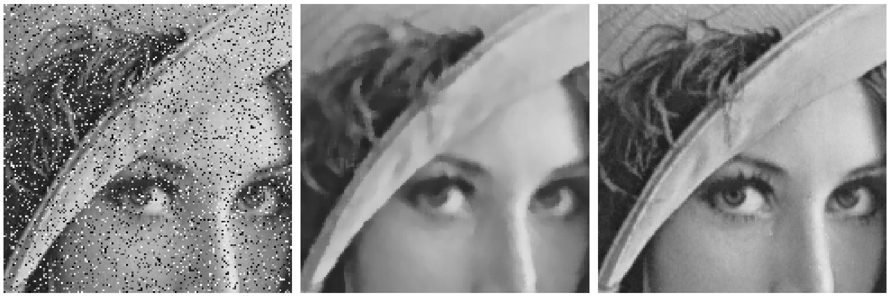
(a) Noisy, $\delta = 30\%$ (b) MSM (30.6dB) (c) IMSM (31.2dB) (d) iDUDE (35.3dB)

(e) Noisy, $\delta = 70\%$ (f) MSM (16.4dB) (g) IMSM (25.7dB) (h) iDUDE (28.9dB)

Fig. 8. Denoising of Boat affected by S&P noise (a $100{\times}100$ image segment is shown).

### B. $M$-ary symmetric noise

Table IV summarizes our results for the $M$-ary symmetric channel. The results are compared with those of the MSM filter, and, for the Lena image, with those published for the state-of-the-art scheme in [21]; a visual comparison is presented in Figure 9. As before, iDUDE significantly outperforms the references.



(a) Noisy, $\delta = 20\%$ (16.2dB) (b) MED (30.1dB) (c) iDUDE (36.9dB)

Fig. 9. Denoising of Lena affected by $M$-ary symmetric noise with $\delta = 20\%$ (a $160\times160$ image segment is shown).

| | $\delta = 10\%$ | | $\delta = 30\%$ | | $\delta = 50\%$ | |
|---|---|---|---|---|---|---|
| image | MED | iDUDE | MED | iDUDE | MED | iDUDE |
| Boat | 26.9 | 33.9 | 25.8 | 29.6 | 23.5 | 26.6 |
| Barbara | 23.1 | 29.9 | 22.7 | 25.4 | 21.2 | 23.5 |
| Tools | 18.9 | 26.9 | 18.4 | 22.3 | 17.1 | 19.2 |
| Bike | 23.4 | 31.1 | 22.4 | 26.0 | 19.9 | 22.2 |

| image: Lena | | | |
|---|---|---|---|
| $\delta$ | MED | ROAD | iDUDE |
| 10% | 30.0 | – | 39.8 |
| 20% | 30.1 | 35.0 | 36.9 |
| 30% | 29.3 | 33.2 | 34.4 |
| 40% | 27.8 | 31.4 | 32.8 |
| 50% | 25.5 | 29.4 | 30.4 |

TABLE IV

RESULTS FOR $M$-ARY SYMMETRIC NOISE. MED: MEDIAN OF A $5\times5$ WINDOW; ROAD: RANK-ORDERED ABSOLUTE DIFFERENCES [21]. COMPARISON WITH ROAD FOR THE LENA IMAGE DISPLAYED SEPARATELY.

### *C. Gaussian noise*

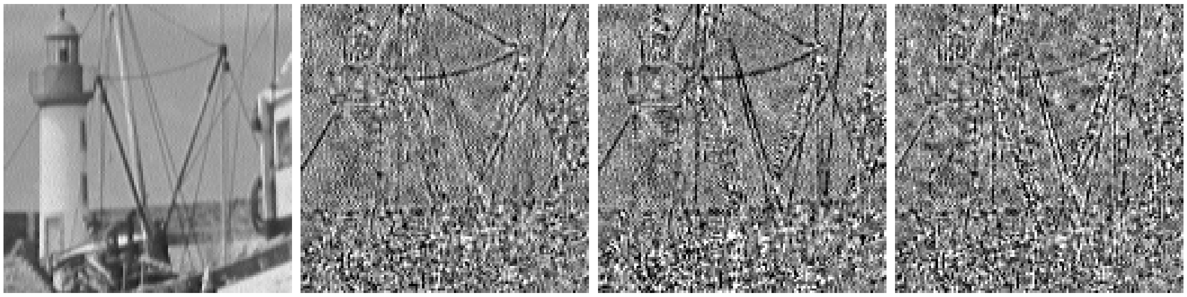
Table V summarizes our results for the Gaussian channel, comparing with the state-of-the-art Block Matching 3D (BM3D) [14], and with the Non Local Means (NML) scheme of [4].[11] We report results for the high SNR regime ($\sigma$=5), and the low SNR regime ($\sigma$=20). For the high SNR regime, we include results for the two variants of iDUDE discussed in Section IV-F.2, namely, one based on LBG clustering, and one based on the WGT model (referred to as iDUDE$^{\mathsf{F}}$). The iDUDE$^{\mathsf{F}}$ variant is competitive at this noise level, and achieves the speeds mentioned above. In the low SNR regime, the LBG-based scheme has a more significant performance advantage, and we report only on this variant. This work has focused on demonstrating the wide applicability of the iDUDE framework for various types of noise and images, rather than optimizing performance specifically for the Gaussian channel, which is work in progress. Although our results for this channel do not reach the performance of [14], they are competitive with those obtained with the denoiser of [4], comparing favorably at $\sigma$=5, and somewhat below at $\sigma$=20. Figure 10 shows denoising error images (i.e., images of differences between denoised and clean samples, re-centered at 128) for a portion of the Boat image at $\sigma$=10. The figure shows that iDUDE and NLM achieve the same PSNR, with iDUDE showing better recovery of edges (which are less marked in the corresponding image) and NLM better performance on smoother areas. BM3D does well on both types of image region, and has better performance overall.

[11] Results for the NLM algorithm were obtained, for $\sigma$=5, using the algorithm described in [4], and for $\sigma$=20, using the slightly different version of the algorithm made available in Matlab by the authors [37]. These versions were found to give the best PSNRs for the respective values of $\sigma$. In all cases, the averaging window was set to 21x21, the similarity window to 7x7, and the parameter $h$ was optimized for each image and $\sigma$. Results for BM3D were obtained with the Matlab code available at [38].

| image | $\sigma = 5$ | | | | $\sigma = 20$ | | |
|---|---|---|---|---|---|---|---|
| | BM3D | NLM | iDUDE | iDUDE$^{\mathsf{F}}$ | BM3D | NLM | iDUDE |
| Lena | 38.7 | 37.7 | 38.0 | 37.8 | 33.0 | 31.3 | 31.3 |
| Boat | 37.2 | 36.1 | 36.6 | 36.3 | 30.9 | 29.6 | 29.4 |
| Barbara | 38.3 | 37.1 | 36.9 | 36.2 | 31.7 | 30.1 | 28.6 |
| Tools | 36.3 | 35.5 | 35.9 | 35.7 | 28.5 | 27.2 | 27.0 |
| Bike | 38.8 | 37.6 | 37.7 | 37.4 | 32.1 | 30.8 | 29.8 |

TABLE V

RESULTS FOR GAUSSIAN NOISE. BM3D: BLOCK MATCHING 3D [14]; NLM: NON LOCAL MEANS [4]; iDUDE: iDUDE USING LBG CONTEXT CLUSTERING; iDUDE$^{\mathsf{F}}$: FAST VARIANT USING WGT CONTEXT CLUSTERING.



(a) Clean (b) BM3D (33.8 dB) (c) NLM (32.9 dB) (d) iDUDE (32.9 dB)

Fig. 10. Denoising of Boat affected by Gaussian noise with $\sigma$=10. A 128 × 128 portion of the denoising error image is shown for each denoiser. The grayscale value in location $i$ of each error image shown is $[8 \cdot (\chi_i - x_i) + 128]$, where the values $\chi_i$ and $x_i$ correspond, respectively, to the denoised and the clean sample in location $i$, and the square brackets denote clamping to the range $[0, 255]$ (multiplication by $8$ enhances visibility of the predominant small-magnitude error values).

## APPENDIX

## DETAILS OF THE WGT PREDICTOR AND CONTEXT CLASSIFIER

We recall that each context is decomposed into eight (overlapping) wings labeled $N$, $S$, $E$ and $W$, $NE$, $SE$, $NW$ and $SW$, defined in Section IV. We recall also that $y_{(a,b)}$ denotes the value of the sample in coordinate $(a, b)$ of the neighborhood in Figure 6. We compute a weighted average, $a_p$, of each wing, as follows:

$$\begin{aligned}
a_N &= \left(2y_{(0,1)} + \sqrt{2}(y_{(-1,1)} + y_{(1,1)}) + y_{(0,2)}\right)/(3 + 2\sqrt{2})\\
a_E &= \left(2y_{(1,0)} + \sqrt{2}(y_{(1,1)} + y_{(1,-1)}) + y_{(2,0)}\right)/(3 + 2\sqrt{2})\\
a_S &= \left(2y_{(0,-1)} + \sqrt{2}(y_{(-1,-1)} + y_{(1,-1)}) + y_{(0,-2)}\right)/(3 + 2\sqrt{2})\\
a_W &= \left(2y_{(-1,0)} + \sqrt{2}(y_{(-1,1)} + y_{(-1,-1)}) + y_{(-2,0)}\right)/(3 + 2\sqrt{2})\\
a_{NE} &= \left(\sqrt{2}(y_{(0,1)} + y_{(1,0)}) + y_{(1,1)}\right)/(1 + 2\sqrt{2})\\
a_{SE} &= \left(\sqrt{2}(y_{(0,-1)} + y_{(1,0)}) + y_{(1,-1)}\right)/(1 + 2\sqrt{2})\\
a_{SW} &= \left(\sqrt{2}(y_{(0,-1)} + y_{(-1,0)}) + y_{(-1,-1)}\right)/(1 + 2\sqrt{2})\\
a_{NW} &= \left(\sqrt{2}(y_{(0,1)} + y_{(-1,0)}) + y_{(-1,1)}\right)/(1 + 2\sqrt{2})
\end{aligned}$$

(in each linear combination, the coefficient of a sample is inversely proportional to its distance to the center of the neighborhood). Additionally, we compute a gradient magnitude, $d_p$, for each wing, as follows:

$$\begin{aligned}
d_N &= \left| \, y_{(0,1)} - y_{(0,2)} + y_{(1,0)} - y_{(1,1)} + y_{(-1,0)} - y_{(-1,1)} \, \right| \\
d_S &= \left| \, y_{(0,-2)} - y_{(0,-1)} + y_{(1,-1)} - y_{(1,0)} + y_{(-1,-1)} - y_{(-1,0)} \, \right| \\
d_E &= \left| \, y_{(2,0)} - y_{(1,0)} + y_{(1,1)} - y_{(0,1)} + y_{(1,-1)} - y_{(0,-1)} \, \right| \\
d_W &= \left| \, y_{(-1,0)} - y_{(-2,0)} + y_{(0,1)} - y_{(-1,1)} + y_{(0,-1)} - y_{(-1,-1)} \, \right| \\
d_{NE} &= \tfrac{1}{\sqrt{2}} \left| \, y_{(2,2)} - y_{(1,1)} + y_{(0,2)} - y_{(-1,1)} + y_{(2,0)} - y_{(1,-1)} \, \right| \\
d_{SE} &= \tfrac{1}{\sqrt{2}} \left| \, y_{(2,-2)} - y_{(1,-1)} + y_{(0,-2)} - y_{(-1,-1)} + y_{(2,0)} - y_{(1,1)} \, \right| \\
d_{NW} &= \tfrac{1}{\sqrt{2}} \left| \, y_{(-1,-1)} - y_{(-2,0)} + y_{(-1,1)} - y_{(-2,2)} + y_{(1,1)} - y_{(0,2)} \, \right| \\
d_{SW} &= \tfrac{1}{\sqrt{2}} \left| \, y_{(-1,1)} - y_{(-2,0)} + y_{(-1,-1)} - y_{(-2,-2)} + y_{(1,-1)} - y_{(0,-2)} \, \right|
\end{aligned}$$

(diagonal gradients are scaled by $\sqrt{2}$).

The fixed prediction value is computed as a linear combination of a subset of the wing averages, with positive weights that decrease with the respective wing gradient, but drop to zero for wings whose gradient magnitude exceeds the minimum gradient in the context by more than a certain *gradient threshold* $g$, which is a parameter of the predictor. Specifically, defining $d_{\min} = \min\{d_N, d_S, d_W, d_E, d_{NW}, d_{NE}, d_{SE}, d_{SW}\}$, wing weights $w_p$ are determined as follows:

$$w_p = \begin{cases} (1+\alpha d_p)^{-1} \,, & d_p - d_{\min} \le g, \quad p \in \{\, \text{N, W, E, S, NW, NE, SW, SE} \,\}\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

Here, $\alpha$ is a parameter of the predictor that controls the effect of the gradient magnitudes on the weights; smaller values of $\alpha$ make the weights vary less with the gradients, with uniform weighting when $\alpha = 0$. We will tend to use smaller values of $\alpha$ when the noise level is high: gradients are less "credible" under those conditions. Finally, the fixed prediction for the context is computed as

$$\tilde{x}(\mathcal{S}_i^{\mathbf{y}}) = \frac{\sum_{p \in \{\text{N,S,W,E,NW,NE,SW,SE}\}} w_p a_p}{\sum_{p \in \{\text{N,S,W,E,NW,NE,SW,SE}\}} w_p} \,. \tag{23}$$

Horizontal/vertical wing gradients are also used to compute the activity level value $A$ of the context, as follows:

$$A(\mathcal{S}_i^{\mathbf{y}}) = d_N + d_S + d_E + d_W \,.$$

## References

[1] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. J. Weinberger, "Universal discrete denoising," in *Proceedings, IEEE Information Theory Workshop*, Bangalore, India, Oct. 2002, pp. 11–14.

[2] ——, "Universal discrete denoising: known channel," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 5–28, 2005.

[3] E. Ordentlich, G. Seroussi, S. Verdú, M. J. Weinberger, and T. Weissman, "A discrete universal denoiser and its application to binary images," in *Proc. IEEE International Conf. Image Processing*, Barcelona, Spain, Sept. 2003, pp. 117–120.

[4] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Sim.*, vol. 4, no. 2, pp. 490–530, 2005.

[5] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standarization into JPEG-LS,” *IEEE Trans. Image Processing*, vol. 9, pp. 1309–1324, Aug. 2000.

[6] X. Wu and N. D. Memon, “Context-based, adaptive, lossless image coding,” *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.

[7] B. Carpentieri, M. J. Weinberger, and G. Seroussi, “Lossless compression of continuous-tone images,” *Proc. IEEE*, vol. 88, no. 11, pp. 1797–1809, Nov. 2000.

[8] J. Rissanen, “Universal coding, information, prediction, and estimation,” *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629–636, July 1984.

[9] M. J. Weinberger and G. Seroussi, “Sequential prediction and ranking in universal context modeling and data compression,” *IEEE Trans. Inform. Theory*, vol. 43, pp. 1697–1706, Sept. 1997.

[10] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Basic Engeneering*, vol. 82, pp. 34–45, 1960.

[11] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, 1949.

[12] D. L. Donoho and I. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, pp. 425–455, 1994.

[13] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of gaussians in the wavelet domain,” *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3D filtering,” in *Proceedings of SPIE in Electronic Imaging: Algorithms and Systems V*, no. 6064A-30, Jan. 2006.

[15] S. Roth and M. J. Black, “Fields of experts: A framework for learning image priors,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, CA, U.S.A., June 2005, pp. 860–867.

[16] M. Elad and M. Aharon, “Image denoising via learned dictionaries and sparse representation,” in *Computer Vision and Pattern Recognition*, 2006, pp. 895–900.

[17] J. Mairal, G. Sapiro, and M. Elad, “Learning multiscale sparse representations for image and video restoration,” *SIAM J. Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, Apr. 2008.

[18] K. Sivaramakrishnan and T. Weissman, “Universal denoising of continuous amplitude signals with applications to images,” in *Proc. IEEE International Conf. Image Processing*, Atlanta, Georgia, USA, Sept. 2006, pp. 2609–2612.

[19] A. Dembo and T. Weissman, “Universal denoising for the finite input general output channel,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 1507–1517, Apr. 2005.

[20] H. Hwang and R. Haddad, “Adaptive median filters: new algorithms and results,” in *Proc. IEEE International Conf. Image Processing*, vol. 4, Washington, DC, U.S.A., Oct. 1995, pp. 499–502.

[21] R. Garnett, T. Huegerich, C. Chui, and W. He, “A universal noise removal algorithm with impulse detector,” *IEEE Trans. Image Processing*, vol. 14, pp. 1747–1754, Nov. 2005.

[22] G. Pok, J.-C. Liu, and A. S. Nair, “Selective removal of impulse noise based on homogeneity level information,” *IEEE Trans. Image Processing*, vol. 12, pp. 85–92, Jan. 2003.

[23] L. I. Rudin and S. Osher, “Total variation based image restoration with free local constraints.” in *Proc. IEEE International Conf. Image Processing*, Austin, TX, USA, Nov. 1994, pp. 31–35.

[24] J.-F. Aujol and G. Gilboa, “Constrained and SNR-based solutions for TV-Hilbert space image denoising,” *Journal of Mathematical Imaging and Vision*, vol. 26, no. 1-2, pp. 217–237, Nov. 2006.

[25] M. Nikolova, “A variational approach to remove outliers and impulse noise,” *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 99–120, 2004.

[26] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and edge-preserving regularization,” *IEEE Trans. Image Processing*, vol. 14, pp. 1479–1485, 2005.

[27] E. Ordentlich, G. Seroussi, S. Verdú, and K. Viswanathan, “Universal algorithms for channel decoding of uncompressed sources,” *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2243–2262, May 2008.

[28] G. Motta, E. Ordentlich, I. Ramírez, G. Seroussi, and M. J. Weinberger, “The DUDE framework for continuous-tone image denoising,” in *Proc. IEEE International Conf. Image Processing*, Genoa, Italy, Sept. 2005, pp. 117–120.

[29] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, ser. Series in Computer Science. World Scientific, 1989.

[30] M. J. Weinberger, J. Rissanen, and R. Arps, “Applications of universal context modeling to lossless compression of gray-scale images,” *IEEE Trans. Image Processing*, vol. 5, no. 4, pp. 575–586, Apr. 1996.

[31] A. Netravali and J. O. Limb, “Picture coding: A review,” *IEEE Proceedings*, vol. 68, pp. 366–406, 1980.

[32] N. Merhav, G. Seroussi, and M. J. Weinberger, “Optimal prefix codes for two-sided geometric distributions,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 121–135, Jan. 2000.

[33] E. Ordentlich, M. Weinberger, and T. Weissman, “Multi-directional context sets with applications to universal denoising and compression,” in *Proc. IEEE International Symp. Inform. Theory*, Adelaide, Australia, Sept. 2005, pp. 1270–1274.

[34] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Comm.*, vol. 28, pp. 84–94, 1980.

[35] http://www.fit.vutbr.cz/˜vasicek/imagedb, Oct. 2008.

[36] http://dmi.uib.es/˜abuades/software.html, Oct. 2008.

[37] http://www.stanford.edu/˜slansel/DenoiseLab/, Oct. 2008.

[38] http://www.cs.tut.fi/˜foi/GCF-BM3D/, Oct. 2008.