The iDUDE Framework for Grayscale Image Denoising

Giovanni Motta, *Member, IEEE*, Erik Ordentlich, *Senior Member, IEEE*, Ignacio Ramírez, Gadiel Seroussi, *Fellow, IEEE*, and Marcelo J. Weinberger, *Fellow, IEEE*

***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 DUDE, however, is not effective 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 relatively 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 not enable effective denoising. 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. Instantiations of the enhanced framework, which is referred to as iDUDE, are described for examples of additive and nonadditive noise. The resulting denoisers significantly surpass the state of the art in the case of salt and pepper (S&P) and M-ary symmetric noise, and perform well for Gaussian noise.**

***Index Terms*—Context-based denoising, discrete universal denoiser (DUDE) algorithm, discrete universal denoising, Gaussian noise, image denoising, impulse noise.**

I. Introduction

THE *discrete universal denoiser* (DUDE), introduced in [1] and [2], aims at recovering a discrete, finite-alphabet sequence, after it has been corrupted by a discrete memoryless noise channel 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. The denoiser 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, through a channel inversion operation, for determining conditional probability distributions 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 upon this conditional distribution. The decision is essentially the Bayes optimal one with respect to the previously mentioned 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 per pixel image, a rather small 3×3 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

G. Motta was with Hewlett-Packard Co., Personal Systems Group, San Diego, CA 92127 USA. He is now with Google, Inc., Mountain View, CA 94043 USA (e-mail: gim@ieee.org).
E. Ordentlich, G. Seroussi and M. J. Weinberger are with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: erik.ordentlich@hp.com; gadiel.seroussi@hp.com; marcelo.weinberger@hp.com).
I. Ramírez was with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA. He is now with the Instituto de Ingenería Eléctrica, Facultad de Ingeniería, Universidad de la República, Montevideo 11300, Uruguay (e-mail: nacho@fing. edu.uy).

[1]These are images with a relatively large dynamic range, e.g., in our examples, 256 grayscale values, which we will refer to as a *large alphabet*. For this image class, 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.

that the DUDE's rate of convergence to optimal performance depends strongly upon 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 yield meaningful denoising 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 DUDE-style schemes 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 upon 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, with a much smaller model. The use of prediction is based upon 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 upon 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. We regard the enhanced DUDE schemes presented in the paper as a *framework* (referred to as the *iDUDE framework*), since what is described is a general architecture for a denoising system incorporating the basic DUDE principles from [2], namely, the estimation of context-conditioned clean sample probability distributions and the application of an optimal Bayes denoising rule based upon the estimated distributions and the given loss function, as well as a set of basic assumptions on grayscale images. The framework can then be specialized for different noise types or image characteristics within the broad class of grayscale images, by changing the specific embodiments of the different algorithmic modules. In particular, knowledge about the noise channel will be incorporated in the form of a *channel transition matrix* describing the probability distributions of noisy samples given clean samples. Some computations will involve the inverse of this matrix (or more generally pseudo-inverses or other methods to estimate distributions of clean samples from distributions of noisy ones). Additionally, as will be described later on, the framework will sometimes rely on a simple prefilter for the noise channel of interest. Thus, by selecting the appropriate matrix, adapting the corresponding inversion methods, and choosing a matching prefilter, the iDUDE framework can be customized to different noise types. On the other hand, prior knowledge about grayscale images, and on the interaction between the targeted channel and the image type will generally be incorporated in the specific design of the context model, including a context aggregation strategy and choices of conventional image predictors used as part of that strategy. We illustrate this flexibility by describing in detail iDUDE instantiations for three popular noise channels, and applying the resulting denoisers to a variety of grayscale image types. These instantiations are intended as specific examples—the framework is not limited to these examples, and we expect that instantiations for other cases can be readily derived using analogous adaptations (we expand on this derivation in the concluding Section VI, after discussing the three specific instantiations presented in the paper). 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, instantiations of iDUDE significantly surpass the state of the art in the case of *salt and pepper* (S&P) and M*-ary symmetric* noise, and perform 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.

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 nonadditive 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.

For additive noise, the most relevant schemes are presented in [4], [12], and [13], and are discussed next; other approaches include *wavelet thresholding* [14], *fields of experts* [15],

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

and *sparse representation* [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 nonparametric 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 [12] with a 3-D 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 3-D array which is then denoised in the 3-D 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. We remark that window/neighborhood-based modeling and processing has also been applied to the denoising of wavelet transform coefficients of additive (Gaussian) noise corrupted images. This approach, pioneered in [13] and refined in [20] and references therein, involves modeling spatially proximate neighborhoods of clean transform coefficients (which may include coefficients from different decomposition levels) as scale mixtures of correlated Gaussian vectors, estimating model parameters from noisy data using empirical Bayes techniques, and denoising a "central" coefficient to its minimum-mean-square-estimate given its noisy neighborhood. A related approach is that of [21] and references therein, in which the marginal distributions of clean coefficients are modeled using parameteric distributions with spatially varying parameter values that are estimated from neighborhoods of noisy transform data. A tradeoff between the estimation neighborhood size and the number of parameters in a specific model based upon Hermite polynomials is studied in [21]. This is somewhat related to the tradeoff we face here between image size and context size/complexity.

Although the models in the previously mentioned 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. Nonadditive noise, on the other hand, poses different problems. A typical example is given by the mentioned 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 [22], [23] or [24] 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 [25] (see also [26]) is well-suited. In this approach, used in [27] and [28] 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 upon 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 [27] and [23].

The rest of the paper is organized as follows. In Section II, we review the basic DUDE concepts, notations, and results from [2] and [3]. Section III describes the tools with which the basic DUDE is enhanced to form the iDUDE framework. We start by defining a set of assumptions capturing properties of smoothness, DC-invariance, and symmetry, which generally hold for natural grayscale images. We then define a statistical model incorporating these assumptions through the use of context aggregation for statistics sharing, including the use of conventional image prediction. Some of the image assumptions, which are postulated initially for clean images, turn out to be less useful or even invalid in the noisy case, in particular under certain types of impulse noise. Consequently, we proceed to adapt the statistical model to noisy conditions. The key idea is to bypass the DUDE's intermediate modeling step of estimating context-conditioned distributions of noisy samples, from which distributions of clean samples are obtained via a linear transformation. In the iDUDE framework, distributions of clean samples are estimated *directly*, without the intermediate stage. To that end, we translate some of the modeling operations to a "cleaner" domain in which our assumptions are again effective, by making use of a *prefiltered* image to build and aggregate contexts, and by effecting the channel inversion step of the DUDE on an amortized, sample-by-sample manner, rather than applying a transformation to a distribution of noisy samples as called for in the basic DUDE. The prefiltered image can be obtained by using a simple denoiser (e.g., a median filter), or from a previous application of an iDUDE denoiser. This leads naturally to an iterative scheme, by which each iteration produces a better reconstruction of the clean image, which is, in turn, used to build the context model for the next iteration. Section IV describes the instantiation of the iDUDE framework for S&P, M-ary symmetric, and Gaussian noise. For each noise type, we describe specific embodiments of the modules of the framework (prefilter, predictor, channel inversion method). We note that the prefilter and iteration mechanisms can lead to violations of the basic DUDE statistical assumptions, and we also describe a statistics monitoring mechanism, instantiated for each noise type, that detects these violations and can be used to stop iteration for the affected parts of the image. In Section V, we describe experiments performed with the denoisers described in Section IV, comparing whenever possible with other denoisers from the literature, including those yielding the best available published results for

the noise type of interest as of the writing of this paper. Finally, in Section VI, we summarize our conclusions and directions of further research.

II. Basic DUDE

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

A. Notation and Problem Setting

Throughout, an *image* is a 2-D 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 2-D 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 ith 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 transition 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 nonsingular. In later sections of this paper, however, we will consider some channel matrices which are nonsingular but badly conditioned and we treat them, in practice, as singular.

A $m \times n$ *image denoiser* is a mapping $\hat{\boldsymbol{\chi}}^{m\times n} : \mathcal{A}^{m\times n} \to \mathcal{A}^{m\times n}$. Assume a given per-symbol *loss function* $\Lambda : \mathcal{A}^2 \to [0, \infty)$, 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 Λ, 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.,

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

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 ith 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 upon the channel transition matrix $\mathbf{\Pi}$ and the loss function Λ, but not on the clean image $\mathbf{x}$, i.e., given $\mathbf{\Pi}$ and Λ, 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 3×3 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

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

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 nonsingular matrix, we write A^{-T} as shorthand for $(A^{-1})^T$. Finally, let $\boldsymbol{\lambda}_a$ and $\boldsymbol{\pi}_a$ denote the ath columns of $\mathbf{\Lambda}$ and $\mathbf{\Pi}$, 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, for $i \in V_{m\times n}$, by

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

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 the empirical conditional distribution, $P_{\mathbf{z}}(\cdot|\mathcal{S}_i^{\mathbf{z}})$, of a noisy sample given its context, and the vector $\mathbf{\Pi}^{-T}\mathbf{m}(\mathbf{z}, \mathcal{S}_i^{\mathbf{z}})$ 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 $\mathbf{\Pi}^{-T}$ performs the "channel inversion"). The vector

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

in turn, can be interpreted, after normalization, as an estimate $\hat{P}_{\mathbf{x}}(\cdot|\mathcal{S}_i^{\mathbf{z}}, z_i)$ of the posterior distribution of the clean sample x_i 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

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

1) Determine $P_{\mathbf{z}}(\cdot \mid \mathcal{S}_i^{\mathbf{z}})$, 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 $P_{\mathbf{z}}(\cdot \mid \mathcal{S}_i^{\mathbf{z}})$ and the channel transition matrix, estimate a distribution $\hat{P}_{\mathbf{x}}(\cdot \mid \mathcal{S}_i^{\mathbf{z}}, z_i)$ 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 $\mathbf{\Lambda}$, produce a denoised value $\hat{\chi} = \hat{\chi}_S^{m\times n}(\mathbf{z})[i]$ according to (2), so that the expectation of the loss $\Lambda(x, \hat{\chi})$ with respect to the distribution $\hat{P}_{\mathbf{x}}(x \mid \mathcal{S}_i^{\mathbf{z}}, z_i)$ estimated in Step 2 is minimized.

Fig. 1. Outline of the DUDE algorithm.

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 Fig. 1.

The universality of the denoiser $\hat{\mathbf{X}}_{\mathsf{univ}}^{m\times n}$ has been shown in two settings. In the *stochastic* setting, the image $\mathbf{x}$ is assumed to be a sample of a spatially stationary process. The results of [2], as extended to the 2-D 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 *semistochastic* 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 sublinear working storage complexities). The algorithm, in both its 1-D and 2-D versions, has been implemented, tested, and shown to be very effective on binary images [2], [3], text [2], and large HTML code files [29]. In [30] and the current work, we enhance the basic scheme to make it effective on grayscale images.

III. iDUDE: A DUDE-Based Framework for Grayscale Image Denoising

In this section, we describe the iDUDE framework in terms of the tools incorporated into the DUDE scheme to enable effective denoising of grayscale images. The framework is described here in generality covering a broad class of channels. Instantiations for specific channels are presented in detail in Sections IV, V.

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 upon 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, our 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 Section 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, contexts 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). There is a tradeoff 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 tradeoff in stochastic modeling which is at the core of the *minimum description length* (MDL) approach to statistical inference [31]. Algorithmic approaches to the optimization of the model size (number of classes) exist, and have been implemented successfully in lossless image compression [32]. However, simpler schemes based upon carefully tuned but fixed models, such as those used in [5] and [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., [33]) 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 Fig. 1, the prediction error distribution should be recentered 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

$$P(x_i = b|\mathbf{s}) \approx P(x_i = b + a|\mathbf{s} + \mathbf{a}) \tag{4}$$

where we assume that $b, b + a \in \mathcal{A}$, $\mathbf{s}, \mathbf{s} + \mathbf{a} \in \mathcal{A}^{\mathcal{S}}$, and 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 is applied *after* the switch to gradient space. Notice that although the use of prediction described here and 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 upon 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 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.

[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 $P_{\mathbf{x}}(x|\mathbf{s}) \approx P_{\mathbf{x}}(M - 1 - x|\mathbf{w} - \mathbf{s})$.

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)). In some applications, additional knowledge on specific image characteristics going beyond the basic set A1–A4 may be available. In those cases, the additional prior knowledge could be incorporated into the design of the predictor, additional symmetries, or other aspects of the context aggregation strategy.

We also note that the use of a fixed context template matches, implicitly, the assumption of stationarity in the stochastic approach to the DUDE in [2] (and is in fact identical to the "sliding window" of the semistochastic approach). The intuitive assumption is that "the same context will produce the same conditional distribution," independently of where the context occurs in the image, which is implicitly adopted in all image processing algorithms based upon sliding windows (including the best lossless image compression schemes).

Example 1: Fig. 2 shows an example of the application of the tools described in A1–A4. Assume that $\mathcal{S}$ is the 3×3 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 Fig. 2. We use the 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:

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

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° 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 previously mentioned 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. 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 Section 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 previously mentioned ranges of κ 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 in context $\mathcal{S}_0$, given its conditioning class $\mathcal{Q}_\kappa = \mathcal{Q}(\mathcal{S}_0)$, is implied by the relation

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

In words, the conditional distribution $P_E(e|\mathcal{Q}_\kappa)$ 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, $0 \leq a \leq M-1$, and zeros elsewhere, and representing the observation of a sample x as $\mathbf{u}_M^x$. For $a \in \mathcal{A}$, define the $M \times (2M-1)$ matrix

$$\mathbf{C}(a) = \overbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}^{M-1-a} \quad \mathbf{I}_M \quad \overbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix}}^{a} \tag{5}$$

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

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

Similarly, we will regard conditional probability distributions $P_U(\cdot|c)$ 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 upon 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

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

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), premultiplying 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 $P_E(\cdot|\mathcal{Q}_\kappa)$ 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

[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].

[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 Section III-F. However, as will become clear later in this subsection, the invertibility problem will not arise for our choice of estimate.

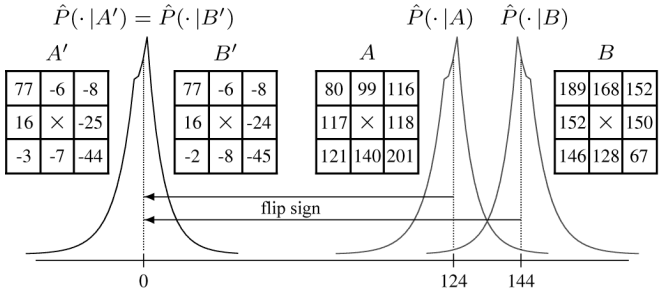


Fig. 2. Merging of context conditional distributions.

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 κ such that $\mathcal{Q}_\kappa = \mathcal{Q}_i^{\mathbf{y}}$.
3) *Normalization.* For each κ, 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 upon prediction and context classification.

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

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 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 upon 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 $P_X(\mathcal{S}_i^{\mathbf{x}})$ based upon $\mathbf{x}$ is given by the vector

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

The overall modeling procedure is outlined in Fig. 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 Section III-D. The case of estimating the model from clean data corresponds to $\mathbf{z} = \mathbf{y} = \mathbf{x}$.

In Fig. 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 upon 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 upon image statistics associated to the context, and a two-stage context clustering strategy. We discuss these enhancements next.

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

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

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 $\hat{x}(\mathcal{S}_i^{\mathbf{x}})$ that depends upon the samples in $\mathcal{S}_i^{\mathbf{x}}$, and also on the image $\mathbf{x}$ through the bias value estimated for $\mathcal{R}(\mathcal{S}_i^{\mathbf{x}})$.

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 *reclustered* into the set of conditioning classes $\{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_K\}$, where each class $\mathcal{Q}_\kappa, 1 \le \kappa \le K$, merges samples from several clusters $\mathcal{R}_\ell$. Hereafter, we interpret the modeling procedure in Fig. 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.

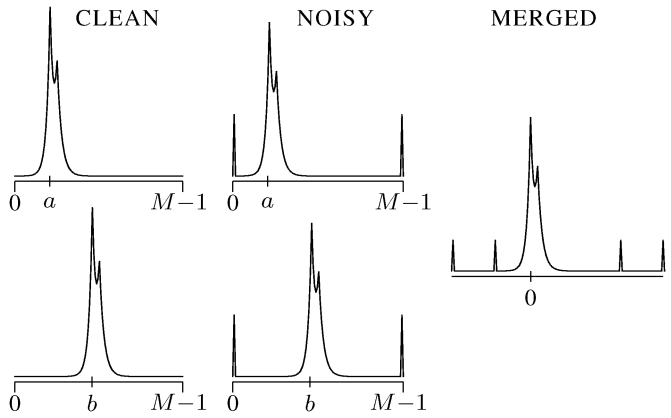


Fig. 4. Effect of S&P noise on merging of similarly-shaped distributions centered at different values.

D. Model Estimation in the Presence of Noise

The discussion in Sections III-A–III-C has focused on the modeling of a clean image, ignoring the effect of noise. A first intuitive approach to incorporating the model into a DUDE-like scheme would apply the context transformation and clustering operations of Fig. 3 to the noisy image to obtain cluster-conditioned distributions as proxies for the distributions $P_{\mathbf{z}}(\cdot|\mathcal{S}_i^{\mathbf{z}})$ called for in Step 1 of Fig. 1, and then, following the procedure of the same figure, use a channel inversion procedure to obtain the distribution estimates $\hat{P}_{\mathbf{x}}(\cdot|\mathcal{S}_i^{\mathbf{z}}, z_i)$ used in the denoising rule. To illustrate why this approach would not be effective in general, consider, for example, a S&P channel. In this channel, a fraction δ 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 Fig. 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 Fig. 1 will not remove the noise.

Although the effect of noise may be more benign for other channels, it is clear from the previous discussion that, in general, additional tools are required to use our image model effectively when the available data is noisy. We discuss the issues of context aggregation and distribution estimation, separately, next.

1) Context Modeling Through Prefiltering: As mentioned, by destroying context similarities, noise can severely limit our ability to aggregate contexts and let them share their statistics. Instead, we will translate the operations on contexts, leading to assignment of samples to conditioning classes, to a "cleaner" domain, where context similarities existing in the unavailable image $\mathbf{x}$ are more likely to be preserved. To this end, we adopt the following additional modeling assumption.

A5) *Robust clustering*. We have access to an image $\mathbf{y}$ such that the formal model of Section III-B (for the clean samples) still applies when the unavailable clean context, $\mathcal{S}_i^{\mathbf{x}}$, is replaced by the corresponding available context, $\mathcal{S}_i^{\mathbf{y}}$, in $\mathbf{y}$.

The image $\mathbf{y}$ can be obtained from available data through "rough" denoising or *prefiltering* of the noisy image $\mathbf{z}$ using a (possibly simpler) denoiser appropriate for the type of noise of interest (e.g., a median filter for S&P noise). Intuitively, we postulate that applying the clustering operations of the procedure in Fig. 3 to the prefiltered image $\mathbf{y}$ will result in an assignment of samples to conditioning classes similar to the one we would obtain if we had access to $\mathbf{x}$. In cases where the effect of noise on context aggregation is less severe (e.g., Gaussian noise at high SNR), a trivial prefilter with $\mathbf{y} = \mathbf{z}$ might suffice.[8] On the other hand, for the more severe cases, Section III-E discusses an iterative process where $\mathbf{y}$ can be the output of a previous iteration of iDUDE.

Our rationale for Assumption A5 is based upon 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 generally fail for nonadditive noise.

It should be noted that prefiltering 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 Section IV-B). Prefiltering can also be seen as a tool for capturing higher order dependencies without increasing model cost. Clearly, with conditioning classes based upon a prefiltered image, the conditioning events for the original noisy samples depend upon a larger number of original samples than the size of the neighborhood used. Thus, prefiltering increases the effective size of the contexts used to condition the distributions, without increasing the number of conditioning classes.

2) Estimation of Clean Sample Distributions From Noisy Data: Our task is now to estimate the previous 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 Fig. 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

[8] In fact, as will be discussed in the sequel, when the alphabet is large and the noise is *additive*, zero-mean, and rapidly decaying, the choice $\mathbf{y} = \mathbf{z}$, together with properties of the channel transition matrix and some mild assumptions, will make the procedure derived from our enhanced approach effectively coincide with the intuitive procedure outlined at the beginning of the section.

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 Section 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 (6)], but also to noise. Now, recall from the discussion following (2) that, in the DUDE algorithm, the channel inversion is accomplished by premultiplication 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 upon 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 Fig. 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).

When the alphabet is large, and the noise is additive, zero-mean, and rapidly decaying, we write $z_i = \hat{x}_i + e_i + \eta_i$, where e_i is the prediction error value, and η_i is the noise addition drawn by the channel. It can be shown that, due to the commutativity of addition, the "shift" effect of the matrix $\mathbf{S}_{\hat{x}_i}$ commutes with the effect of the channel in (12), and the resulting procedure is essentially equivalent (up to negligible border effects) to the intuitive procedure described at the beginning of the subsection, which is indeed effective for this type of noise.

In the more general case, at first sight, with the choice of estimation matrix $\mathbf{M}'$ in (12), the update of $\mathbf{e}_\kappa$ in Fig. 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 estimate of $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$ in (11) can be interpreted as follows. Define $\mathcal{Q}_{\kappa,p} = \mathcal{Q}_\kappa \cap \{\mathcal{S}_i^{\mathbf{y}} | \hat{x}(\mathcal{S}_i^{\mathbf{y}}) = p\}$, referred to as a *subcluster*. For a cluster $\mathcal{Q}_\kappa$, the result of Step 2 of the procedure in Fig. 3 (with the choice (12) for $\mathbf{M}'(\hat{x}_i)$) can be written as

$$\mathbf{e}_\kappa = \sum_{i:\mathcal{S}_i^{\mathbf{y}} \in \mathcal{Q}_\kappa} \mathbf{S}_{\hat{x}_i}\mathbf{\Pi}^{-T}\mathbf{u}_M^{z_i} = \sum_{p \in \mathcal{A}} \sum_{i:\mathcal{S}_i^{\mathbf{y}} \in \mathcal{Q}_{\kappa,p}} \mathbf{S}_p\mathbf{\Pi}^{-T}\mathbf{u}_M^{z_i}$$

$$= \sum_{p \in \mathcal{A}} \mathbf{S}_p\mathbf{\Pi}^{-T} \sum_{i:\mathcal{S}_i^{\mathbf{y}} \in \mathcal{Q}_{\kappa,p}} \mathbf{u}_M^{z_i} = \sum_{p \in \mathcal{A}} \mathbf{S}_p(\mathbf{\Pi}^{-T}\mathbf{m}_{\kappa,p}) \tag{13}$$

where $\mathbf{m}_{\kappa,p}$ denotes a vector of occurrence counts of noisy symbols in the subcluster $\mathcal{Q}_{\kappa,p}$. 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 $\mathbf{P}_{\mathbf{x}}(a|\mathcal{Q}_{\kappa,p})$ of samples $a \in \mathcal{A}$ conditioned on the subcluster $\mathcal{Q}_{\kappa,p}$, 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 subclusters $\mathcal{Q}_{\kappa,p}$. The distributions of prediction errors for clean symbols obtained for the subclusters are merged to yield $\hat{P}_E(\cdot|\mathcal{Q}_\kappa)$, and the estimated conditional distribution of x_i given $\mathcal{S}_i^{\mathbf{y}}$ is given by (9). Notice, however, that the goodness of this estimate does not rely on the law of large numbers "kicking in" for *each* subcluster, 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 Fig. 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 of $\hat{\mathbf{P}}_E(\mathcal{Q}_\kappa)$, $-M+1 \le e \le M-1$, and, for real numbers α, β, let $(\alpha-\beta)^+$ denote $\alpha-\beta$ if $\alpha > \beta$, or 0 otherwise. Consider a real number γ, $0 \le \gamma \le 1$. Since $\sum_e p_e = 1$, there exists a real number μ_γ 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 Fig. 3.

Finally, the corresponding estimate $\hat{\mathbf{P}}_X(\mathcal{S}_i^{\mathbf{y}})$ obtained in Step 4b of the procedure of Fig. 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 (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 prefiltered 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

- 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 prefiltering.

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 Section IV-C). The iterative procedure is summarized in Fig. 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 prefiltered 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 nonsingular 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 [33], [34], which is parametrized by a decay factor θ and a mean μ (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 μ. 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 θ, μ from the data, we first notice that

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

where the left-hand side of (16) is an observed statistic. 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 σ^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 $\mathrm{LG}(\theta,\mu,\varsigma^2)$, with ς^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 θ, μ and ς can be derived in terms of the error function $\mathrm{erf}(\cdot)$ (see Appendix A). Although these expressions are rather unwieldy, they lend themselves to numerical computations, and therefore allow for a numerical maximum-likelihood estimation of the parameters θ and μ (ς is assumed given) from the statistics $z_i - \hat{x}(\mathcal{S}_i^{\mathbf{y}})$ collected for the conditioning class cluster of $\mathcal{S}_i^{\mathbf{y}}$. In our implementation we use a simpler parameter estimation procedure, described in Section IV-F. With these estimated parameters on hand, we write

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

and estimate $\mathbf{P}_X(\mathcal{S}_i^{\mathbf{y}})$ to be a $\mathrm{LG}(\hat{\theta},\hat{\mu},\sigma^2/k)$ 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 ς 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 $\mathrm{LG}(\hat{\theta},\hat{\mu},\varsigma^2-\sigma^2)$. Notice that these computations are carried out only once per conditioning cluster, at the end of the first pass of the DUDE denoising procedure, and independently of the size of the image.

Aside from providing an alternative to the channel matrix inversion, this parametric approach has model cost advantages, since only two parameters, θ and μ, need to be estimated per conditioning class [34], 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 instantiations of the iDUDE framework for three types of noise, namely, S&P noise, M-ary

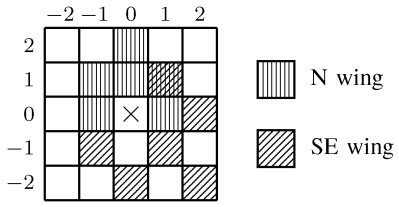


Fig. 6. Neighborhood for the WGT modeling scheme.

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 δ), and quantized additive white Gaussian noise. We assume that the Euclidean (L_2) norm is used for the loss function Λ 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 Fig. 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 Conditioning Model

Our context model is based upon the 5×5 neighborhood $\mathcal{S}$ shown in Fig. 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 [5] and [6] 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 *wing gradients and texture* (WGT).

We denote by $y_{r,s}$ the value of the sample in coordinate (r,s) of the neighborhood in Fig. 6, with $-2 \le r, s \le 2$. As the neighborhood slides accross the image, the actual coordinates of the context samples are $i+(r,s)$, $i \in V_{m\times n}$; for clutter reduction, we omit the center coordinate i in this discussion and in Appendix B. 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 Fig. 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° 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° 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 deemphasize parts that might be crossed by sharp edges. The precise details of the computation are given in Appendix B.

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 Appendix B. 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, $T \ge 0$, 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 Fig. 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 upon using an observable parameter that correlates with denoising performance, and optimizing the settings based upon 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 [35].

In our implementations, we have grouped images by size ("very small," "small," and "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 basic guidelines on context models: larger images can sustain larger models, and so do cleaner images (intuitively, since less is learned from noisy data than from clean data). Specific parameter values are given in Table II of Section V.

C. Monitoring of the Statistical Model During Iteration

As mentioned in Section III-E, the iDUDE iteration of Fig. 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

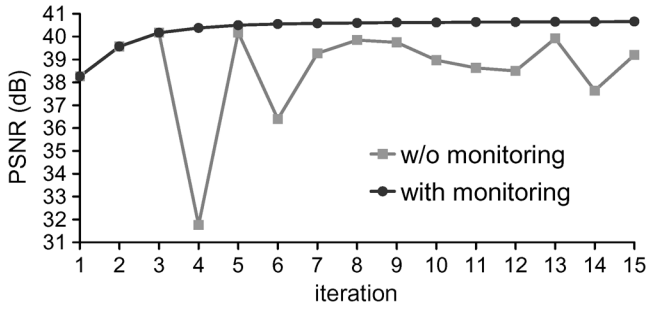


Fig. 7. Effect of statistics monitoring on the iDUDE iteration performance (S&P noise).

(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 nonadditive 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 prefiltered 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.

Fig. 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 offdiagonal entries in each column of the channel matrix $\mathbf{\Pi}$ are equal, which is the case for the two nonadditive channels studied here, the probability of $z_i = c$ given $x_i \neq c$ (and the cluster $\mathcal{Q}_\kappa$) is clearly the common offdiagonal 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 previously mentioned procedure 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 S&P Noise

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

$$\mathbf{\Pi}_{\mathsf{sp}}(\delta) = \begin{bmatrix} 1-\frac{\delta}{2} & 0 & \cdots & 0 & \frac{\delta}{2} \\ \frac{\delta}{2} & 1-\delta & \cdots & 0 & \frac{\delta}{2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\delta}{2} & 0 & \cdots & 1-\delta & \frac{\delta}{2} \\ \frac{\delta}{2} & 0 & \cdots & 0 & 1-\frac{\delta}{2} \end{bmatrix}$$

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

The matrices are well conditioned, except when δ approaches one.[9]

1) Prefiltering: The iDUDE implementation for the S&P channel uses a prefilter $\mathcal{F}$ based upon a *modified selective median* (MSM) filter for the first step of the procedure of Fig. 5. The filter is applied only to samples valued 0 and $M-1$. It estimates the sample at the center of a 5×5 window by computing the median of a set of 25 values, namely, the noncenter sample values in the window and their average. The prefilter is improved by running the MSM filter iteratively (still within the first step in Fig. 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 prefilter as an *iterated* MSM, or, in short, IMSM. The improvement of IMSM over MSM is illustrated in Table III and Fig. 9 of Section V.

The output from the IMSM prefilter is used as input to the first application of the iDUDE in the second stage of the procedure of Fig. 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 Fig. 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 $\delta/(2(1-\delta))$

[9]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.

from $\mathbf{e}_\kappa[-\hat{x}_i]$ and $\mathbf{e}_\kappa[M-1-\hat{x}_i]$. Notice that the latter two subtractions depend upon 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, $\overline{\xi}$, 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 γ is an appropriate normalization coefficient, and substituting the first column of $\boldsymbol{\Pi}_{\mathsf{sp}}(\delta)$ from (17) for $\boldsymbol{\pi}_0$, we obtain

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

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 $\overline{\xi}_i$ 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 $\boldsymbol{\Pi}_{\mathsf{M}}(\delta)$, with entries

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

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

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

1) Prefiltering: A simple median filter on a 5×5 window (modified, as in Section IV-D-1, to include the average of the 24 noncenter samples in lieu of the center) is used for the first step of the procedure in Fig. 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 $\boldsymbol{\Pi}_{\mathsf{M}}^{-T}$ can be written in the form

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

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 Fig. 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 $\boldsymbol{\Pi}_{\mathsf{M}}$ given in (18), the computation of the expectation of ξ under $\hat{P}_{\mathbf{x}}(\xi|\mathcal{S}_i^{\mathbf{z}}, z_i)$ for the M-ary symmetric channel yields

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

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 $\overline{\xi}_i$.

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 ζ_i in $\mathcal{A}$. Thus, letting $\mathrm{erf}(\cdot)$ denote the error function [37], the entries of the channel transition matrix $\boldsymbol{\Pi}_{\mathsf{G}} = \{\Pi_{\mathsf{G}}(a,b)\}_{0 \le a,b \le M-1}$ are given by

$$\Pi_{\mathsf{G}}(a,b) = \begin{cases} \frac{1}{2}\left(\mathrm{erf}\left(\frac{b-a+\frac{1}{2}}{\sqrt{2}\sigma}\right) - \mathrm{erf}\left(\frac{b-a-\frac{1}{2}}{\sqrt{2}\sigma}\right)\right), & 0 \le a < b < M-1 \\ \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{M-\frac{3}{2}-a}{\sqrt{2}\sigma}\right)\right), & 0 \le a < b = M-1 \\ \mathrm{erf}\left(\frac{1}{2\sqrt{2}\sigma}\right), & 0 < a = b < M-1 \\ \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{1}{2\sqrt{2}\sigma}\right)\right), & a = b = M-1 \\ \Pi_{\mathsf{G}}(M-1-a, M-1-b), & \text{otherwise.} \end{cases} \tag{21}$$

As discussed in Section III-F, the matrix $\boldsymbol{\Pi}_{\mathsf{G}}$ is generally ill-conditioned, and we will not attempt to utilize its inverse (see *Model Estimation* in the following).

1) Prefiltering: The iteration of Fig. 5 includes at most two applications of the iDUDE, using a trivial prefilter (i.e., $\mathcal{F}$ is the identity). In principle, prefiltering and iteration are optional for the Gaussian channel, since our assumptions A1–A4 are still effective for images affected by Gaussian noise (especially in the high SNR regime), and therefore these assumptions can be used for modeling $P_{\mathbf{z}}(\cdot|\mathcal{S}_i^{\mathbf{z}})$ and doing channel inversion to obtain $\hat{P}_{\mathbf{x}}(\cdot|\mathcal{S}_i^{\mathbf{z}})$ as in the basic DUDE. This is reflected in our results in Section V, where we use trivial prefiltering and no iteration for the high SNR regime. One round of iteration does help in the low SNR regime, but 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 [12], used as a prefilter, would have resulted in improved performance. However, the use of such a prefilter 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 Section 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 Fig. 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 [38], with the L_2 metric used to measure distance between contexts.

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

very small images			small images			large images		
image	size	source	image	size	source	image	size	source
Set of 24 images (Set_{24})	384×256	[36]	Lena	512×512	TR	Tools	1524×1200	$\mathrm{JLS}^{\mathrm{Y}}$
			Lena*	512×512	[28]	Toolsk	1524×1200	$\mathrm{JLS}^{\mathrm{K}}$
			Boat	512×512	TR	Womank	2048×2560	$\mathrm{JLS}^{\mathrm{K}}$
			Barbara	512×512	TR	Bike	2048×2560	$\mathrm{JLS}^{\mathrm{Y}}$

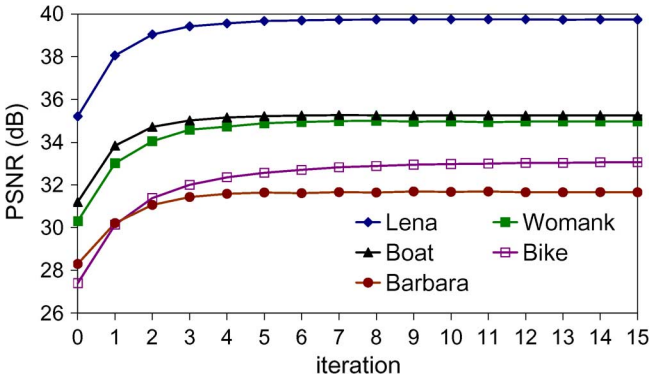


Fig. 8. Evolution of PSNR versus number of iterations for a subset of the image test set under S&P noise with $\delta = 30\%$. Iteration number 0 corresponds to the initial prefiltering stage (IMSM filter).

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 Section III-F, but with a simpler estimation procedure for the cluster-dependent parameters θ and μ 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 τ^2, we observe that by the definition of the LG model, its variance ν^2 is given by $\nu^2 = \tau^2 + \sigma^2(1 + k^{-1})$. Given the parameters of the Laplacian, τ^2 takes the form

$$\tau^2 = \frac{2\theta}{(1-\theta)^2} + r(1-r) \tag{22}$$

where r denotes the fractional part of μ. 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 τ_κ^2 of the Laplacian component for $\mathcal{Q}_\kappa$ as

$$\hat{\tau}_\kappa^2 = \max\left(\hat{r}_\kappa(1-\hat{r}_\kappa), \hat{\nu}_\kappa^2 - \sigma^2(1+k^{-1})\right) \tag{23}$$

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 (23) 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 (22)), due to statistical fluctuations or an inaccurate choice of the parameter k. Finally, given $\hat{\mu}_\kappa$ and $\hat{\tau}_\kappa^2$, we use (22) 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, and including in all cases the schemes with the best available published results as of the writing of this paper. Our iDUDE experiments are based upon a research prototype implementation written in C++, and run on a vintage 2007 Intel-based personal computer.[10] 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 × 2560 image Bike at approximately 7 s, for a throughput of approximately 730 Kpixels/s. Running times for a given context model do not vary significantly with the noise type or level. As will be shown in Fig. 8 and its discussion, the number of iterations necessary to approach the best performance is generally small.

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 × 256 (referred to as Set_{24}) available at [36], for which results of denoising with the state of the art scheme of [28] at various levels of S&P noise are available. The "small" (512 × 512) images in the table are from the set traditionally used in the image processing literature.[11] 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 α (cf. Appendix B) were

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

[11]We use the versions available at the DenoiseLab site [39]. Additionally, to allow comparison with [28] also on a 512 × 512 image, we use the (different) version of the Lena image reported on in [28], which we refer to as Lena^*. We are not aware of other images for which a reliable comparison with [28] is possible.

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

S&P										M-ary symmetric							Gaussian							
	v. small			small			large				small			large					LBG				WGT	
																			small		large		all	
δ	R	K	T	R	K	T	R	K	T	δ	R	K	T	R	K	T	σ	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 III
RESULTS FOR S&P NOISE. MSM: MODIFIED SELECTIVE MEDIAN (CF. SECTION IV-D-1); IMSM: ITERATED MSM; CHN05: THE DENOISER OF [28]. COMPARISON WITH CHN05 DISPLAYED SEPARATELY

	$\delta = 10\%$			$\delta = 30\%$			$\delta = 50\%$			$\delta = 70\%$		
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/ δ	MSM	IMSM	CHN05	iDUDE
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

set as follows: $g = 8\%$ of maximum gradient magnitude in the context, $\alpha = 0.075$ for the S&P channel, $\alpha = 0.1$ for the M-ary symmetric channel; for the Gaussian channel, g and α were optimized to minimize the observable prediction RMSE for each noisy image, with g varying between 5% and 17%, and α 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 upon 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 images, 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 Fig. 9. For this channel, we compare our results to those of [28] on the Lena^* variant of the Lena image, and on the mentioned Set_{24} from [36]. For the latter, for brevity, we list the *average* PSNR over the set of images (as done also for the results reported in [28]). The scheme of [28] (referred to in the table as CHN05) was selected for comparison as it presents, to the best of our knowledge, the best published results for S&P noise available in the literature. 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 [28] 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.

Fig. 8 shows the evolution of PSNR with the number of iterations for a subset of the test images, under S&P noise with $\delta = 30\%$. The figure shows that, typically, most of the gains in performance are obtained in the first few iterations (this fact is also verified in Fig. 7, which corresponds to the Womank image with 10% S&P noise). Thus, the number of iterations provides for a graceful tradeoff between running time and denoising performance.

B. M-ary Symmetric Noise

Table IV summarizes our results for the M-ary symmetric channel. The results are compared with those of the median prefilter, and, for the Lena image, with those published for the state of the art scheme in [23] (referred to in the table as ROAD); a visual comparison is presented in Fig. 10. As before, iDUDE significantly outperforms the references.

C. Gaussian Noise

Table V summarizes our results for the Gaussian channel, comparing with the state of the art Block Matching 3-D

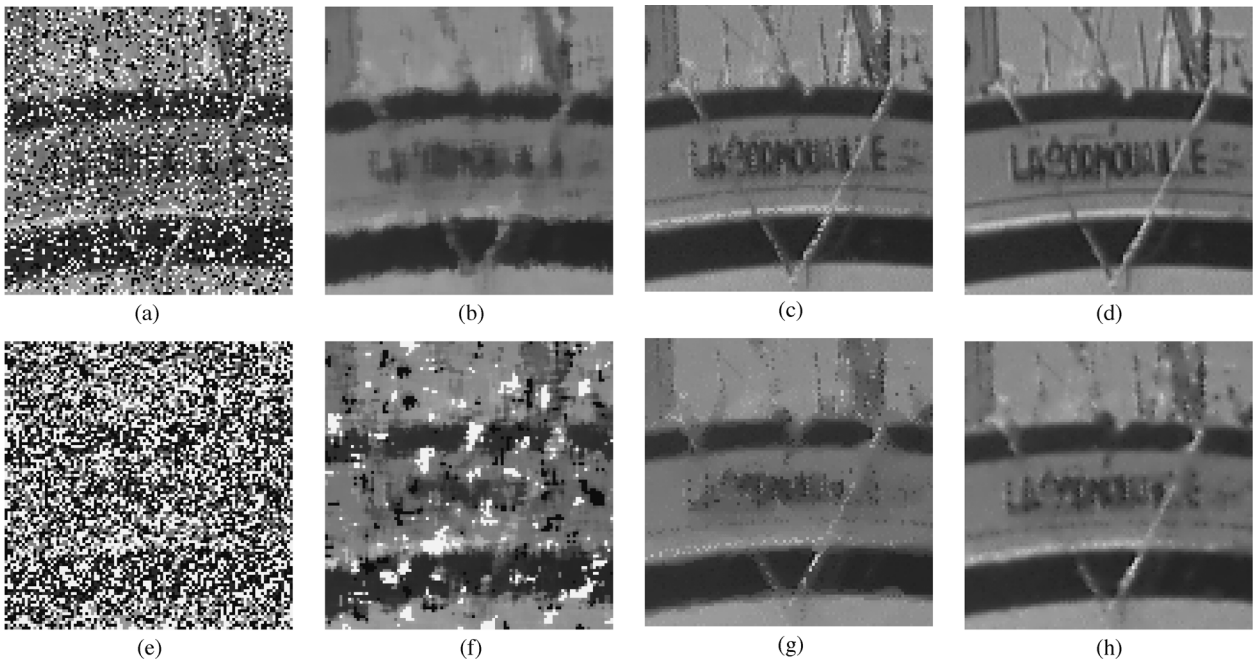


Fig. 9. Denoising of Boat affected by S&P noise (a 100 × 100 image segment is shown). (a) Noisy, $\delta = 30\%$; (b) MSM (30.6 dB); (c) IMSM (31.2 dB); (d) iDUDE (35.3 dB); (e) Noisy, $\delta = 70\%$; (f) MSM (16.4 dB); (g) IMSM (25.7 dB); (h) iDUDE (28.9 dB).

TABLE IV
RESULTS FOR M-ARY SYMMETRIC NOISE. MED: MEDIAN OF A 5 × 5 WINDOW; ROAD: RANK-ORDERED ABSOLUTE DIFFERENCES [23]. COMPARISON WITH ROAD FOR THE LENA IMAGE DISPLAYED SEPARATELY

	$\delta = 10\%$		$\delta = 30\%$		$\delta = 50\%$	
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			
δ	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

(BM3-D) [12], and with the Nonlocal Means (NLM) scheme of [4].[12]

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 DUDE discussed in Section IV-F-2, namely, one based upon LBG clustering, and one based upon the WGT model (referred to as iDUDE^F). The iDUDE^F variant is competitive at this noise level, and achieves the speeds mentioned previously. 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 [12], they are competitive with those obtained with the denoiser of [4], comparing favorably at $\sigma = 5$, and somewhat below at $\sigma = 20$. Fig. 11 shows denoising error images (i.e., images of differences between denoised and clean samples, recentered at brightness level 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. BM3-D does well on both types of image regions, and has better performance overall.

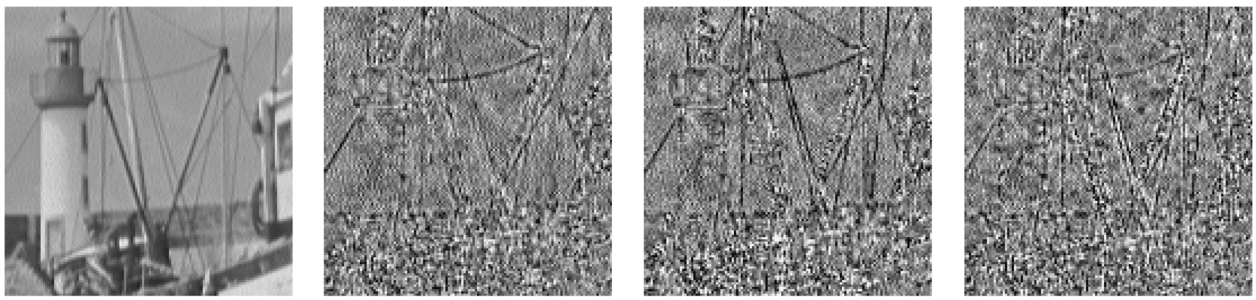
[12]Results for the NLM algorithm were obtained, for $\sigma = 5$, using the algorithm described in [4], and for $\sigma = 20$, using the slightly different version of the algorithm made available in Matlab by the authors [40]. These versions were found to give the best PSNRs for the respective values of σ. In all cases, the averaging window was set to 21 × 21, the similarity window to 7 × 7, and the parameter h was optimized for each image and σ. Results for BM3-D were obtained with the Matlab code available at [41].

VI. CONCLUSION

We have presented a framework for grayscale image denoising based upon the discrete universal denoiser DUDE of [2]. The framework overcomes the practical limitations, stemming from the model cost issues associated with large alphabets and limited sizes of image data, by exploiting prior knowledge on the structure of images, as previously done in lossless image compression, and confirms an important principle in the practical use of universal schemes: Algorithms should be as universal as necessary for the application at hand but not more—they should not be expected to learn what is already known in advance. In that sense, the full universality of the basic DUDE in the class of stationary sources is excessive

(a) (b) (c)

Fig. 10. Denoising of Lena affected by M-ary symmetric noise with $\delta = 20\%$ (a 160×160 image segment is shown). (a) Noisy, $\delta = 20\%$ (16.2 dB); (b) MED: median of a 5×5 window (30.1 dB); (c) iDUDE (36.9 dB).



(a) (b) (c) (d)

Fig. 11. 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 χ_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 eight enhances visibility of the predominant small-magnitude error values). (a) Clean; (b) BM3-D (33.8 dB); (c) NLM (32.9 dB); (d) iDUDE (32.9 dB).

TABLE V
RESULTS FOR GAUSSIAN NOISE. BM3-D: BLOCK MATCHING 3-D [12]; NLM: NON LOCAL MEANS [4]; iDUDE: iDUDE USING LBG CONTEXT CLUSTERING; iDUDE^F: FAST VARIANT USING WGT CONTEXT CLUSTERING

image	$\sigma = 5$				$\sigma = 20$		
	BM3D	NLM	iDUDE	iDUDE^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

for grayscale images. Instantiations of the enhanced iDUDE framework were shown to be effective on a variety of image and noise types, achieving state of the art denoising performance for impulse channels (S&P and M-ary symmetric), and performance competitive with modern denoising schemes for the Gaussian channel. Further improvements in performance for the latter is a subject of ongoing research.

The examples presented in the paper suggest that the following general steps are required for instantiating iDUDE for a different noise channel $\mathbb{C}$, characterized by a channel transition matrix $\mathbf{\Pi}_{\mathbb{C}}$.

1) Determine whether the image assumptions A1–A4 of Section III are effective for images affected by $\mathbb{C}$. If not, choose an appropriate prefilter for $\mathbb{C}$, for use in a prefiltering and iteration loop as described in Fig. 5. If the assumptions are still effective for $\mathbb{C}$, prefiltering and iteration might not be essential (as in the case of Gaussian noise).
2) Determine if a numerically stable inverse $\mathbf{\Pi}_{\mathbb{C}}^{-1}$ can be obtained. If so, the inverse can be used to define an estimation matrix $\mathbf{M}'$ as in (12). Otherwise, an alternative channel inversion method, such as those discussed in Section III-F might be appropriate.
3) Design a context model and aggregation strategy appropriate for images affected by $\mathbb{C}$, and possibly prefiltered. For example, if gradient information remains reliable under $\mathbb{C}$ and prefiltering, a context model similar to WGT (cf. Section IV-A) might be appropriate. Otherwise, as in the low-SNR Gaussian channel case, an LBG-based context model might work better. In addition, some parameters of the context model can be made to depend upon the strength, and not just the type, of the noise (e.g., the parameters α, g in WGT).

Of course, given a specific channel $\mathbb{C}$ and matrix $\mathbf{\Pi}_{\mathbb{C}}$, further optimizations are likely to emerge as a result of experimentation and analysis.

APPENDIX A
LG DISTRIBUTION

To obtain simpler expressions, we approximate the discrete LG distribution as follows. Consider a continuous, infinitely supported Laplacian distribution with probability density function (PDF) parametrized as $P_L(x) = -(1/2)(\ln\theta)\theta^{|x+d|}$, for $0 < \theta < 1$ and, without loss of generality (up to integer translation of the support), $-(1/2) \le d < (1/2)$. Let $X \sim P_L$, and let Y be a normal random variable of zero mean and variance ς^2.

Then, letting $\lambda = \ln\theta$, the cumulative density function (CDF) for $Z = X + Y$ is given by

$$\begin{aligned}F_{\theta,d,\varsigma}(z) = &\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\sqrt{2}}{2}\frac{d+z}{\varsigma}\right)\\ &-\frac{1}{4}e^{\lambda\left(\frac{1}{2}\varsigma^2\lambda+d+z\right)}\left(\mathrm{erf}\left(\frac{\sqrt{2}}{2}\frac{d+z+\varsigma^2\lambda}{\varsigma}\right)+1\right)\\ &-\frac{1}{4}e^{\lambda\left(\frac{1}{2}\varsigma^2\lambda-d-z\right)}\left(\mathrm{erf}\left(\frac{\sqrt{2}}{2}\frac{d+z-\varsigma^2\lambda}{\varsigma}\right)-1\right).\end{aligned}$$

To obtain the PDF for a discrete LG random variable, we write $P_{\theta,d,\varsigma}(Z = z) = F_{\theta,d,\varsigma}(z + (1/2)) - F_{\theta,d,\varsigma}(z - (1/2))$ for values of z away from the borders of the alphabet range, and appropriate accumulation and adjustment at the borders.

Appendix B
Details of the WGT Predictor and Context Classifier

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

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

(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:

$$\begin{aligned}d_N &= |y_{0,1} - y_{0,2} + y_{1,0} - y_{1,1} + y_{-1,0} - y_{-1,1}|\\ d_S &= |y_{0,-2} - y_{0,-1} + y_{1,-1} - y_{1,0} + y_{-1,-1} - y_{-1,0}|\\ d_E &= |y_{2,0} - y_{1,0} + y_{1,1} - y_{0,1} + y_{1,-1} - y_{0,-1}|\\ d_W &= |y_{-1,0} - y_{-2,0} + y_{0,1} - y_{-1,1} + y_{0,-1} - y_{-1,-1}|\\ d_{NE} &= \frac{1}{\sqrt{2}}|y_{2,2} - y_{1,1} + y_{0,2} - y_{-1,1} + y_{2,0} - y_{1,-1}|\\ d_{SE} &= \frac{1}{\sqrt{2}}|y_{2,-2} - y_{1,-1} + y_{0,-2} - y_{-1,-1} + y_{2,0} - y_{1,1}|\\ d_{NW} &= \frac{1}{\sqrt{2}}|y_{-1,-1} - y_{-2,0} + y_{-1,1} - y_{-2,2} + y_{1,1} - y_{0,2}|\\ d_{SW} &= \frac{1}{\sqrt{2}}|y_{-1,1} - y_{-2,0} + y_{-1,-1} - y_{-2,-2}\\ &\qquad + y_{1,-1} - y_{0,-2}|\end{aligned}$$

(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 $D = \{N, W, E, S, NW, NE, SW, SE\}$, and $d_{\min} = \min\{d_p | p \in D\}$, wing weights w_p are determined as follows:

$$w_p = \begin{cases}(1 + \alpha d_p)^{-1}, & d_p - d_{\min} \le g, \quad p \in D\\ 0, & \text{otherwise.}\end{cases}$$

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

$$\tilde{x}\left(\mathcal{S}_i^{\mathbf{y}}\right) = \frac{\sum_{p\in D} w_p a_p}{\sum_{p\in D} w_p}. \tag{24}$$

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

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

References

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

[2] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. J. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 5–28, Jan. 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 Int. Conf. Image Process.*, Barcelona, Spain, Sep. 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 Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[6] X. Wu and N. D. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, 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. Inf. Theory*, vol. IT-30, no. 4, pp. 629–636, Jul. 1984.

[9] M. J. Weinberger and G. Seroussi, "Sequential prediction and ranking in universal context modeling and data compression," *IEEE Trans. Inf. Theory*, vol. 43, no. 5, pp. 1697–1706, Sep. 1997.

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

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

[12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Proc. SPIE Electron. Imag. Algorithms Syst. V*, Jan. 2006, no. 6064A-30.

[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 Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

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

[15] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005, vol. 2, pp. 860–867.

[16] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," *Comput. Vis. Pattern Recognit.*, pp. 895–900, 2006.

[17] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *SIAM J. Multiscale Model. Sim.*, 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 Int. Conf. Image Process.*, Atlanta, GA, Sep. 2006, pp. 2609–2612.

[19] A. Dembo and T. Weissman, "Universal denoising for the finite input general output channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1507–1517, Apr. 2005.

[20] M. Miller and N. Kinsgbury, "Image denoising using derotated complex wavelet coefficients," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1500–1511, Sep. 2008.

[21] S. M. M. Rahman, M. O. Ahmad, and M. N. S. Swamy, "Bayesian wavelet-based image denoising using the gauss-hermite expansion," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1755–1771, Oct. 2008.

[22] H. Hwang and R. Haddad, "Adaptive median filters: New algorithms and results," in *Proc. IEEE Int. Conf. Image Process.*, Washington, DC, Oct. 1995, vol. 4, pp. 499–502.

[23] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with impulse detector," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1747–1754, Nov. 2005.

[24] G. Pok, J.-C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 85–92, Jan. 2003.

[25] L. I. Rudin and S. Osher, "Total variation based image restoration with free local constraints," in *Proc. IEEE Int. Conf. Image Process.*, Austin, TX, Nov. 1994, pp. 31–35.

[26] J.-F. Aujol and G. Gilboa, "Constrained and SNR-based solutions for TV-Hilbert space image denoising," *J. Math. Imag. Vis.*, vol. 26, no. 1–2, pp. 217–237, Nov. 2006.

[27] M. Nikolova, "A variational approach to remove outliers and impulse noise," *J. Math. Imag. Vis.*, vol. 20, no. 1–2, pp. 99–120, 2004.

[28] 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 Process.*, vol. 14, no. 10, pp. 1479–1485, Oct. 2005.

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

[30] G. Motta, E. Ordentlich, I. Ramírez, G. Seroussi, and M. J. Weinberger, "The DUDE framework for continuous-tone image denoising," in *Proc. IEEE Int. Conf. Image Process.*, Genoa, Italy, Sep. 2005, pp. 117–120.

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

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

[33] A. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, no. 3, pp. 366–406, Mar. 1980.

[34] N. Merhav, G. Seroussi, and M. J. Weinberger, "Optimal prefix codes for two-sided geometric distributions," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 121–135, Jan. 2000.

[35] E. Ordentlich, M. Weinberger, and T. Weissman, "Multi-directional context sets with applications to universal denoising and compression," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005, pp. 1270–1274.

[36] http://www.fit.vutbr.cz/~vasicek/imagedb, Oct. 2008.

[37] M. Abramowitz and I. A. Stegun*, Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1964.

[38] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM–28, no. 1, pp. 84–94, Jan. 1980.

[39] http://dmi.uib.es/~abuades/software.html, Oct. 2008.

[40] http://www.stanford.edu/~slansel/DenoiseLab/, Oct. 2008.

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