Scale recognition, regularization parameter selection, and Meyer's G norm in total variation regularization *

David M. Strong † Jean-François Aujol ‡ Tony F. Chan §

December 28, 2004

Abstract

We investigate how TV regularization naturally recognizes scale of individual image features, and we show how perception of scale depends on the amount of regularization applied to the image. We give an automatic method for finding the minimum value of the regularization parameter needed to remove all features below a user-chosen threshold. We explain the relation of Meyer's G norm to the perception of scale, which provides a more intuitive understanding of this norm. We consider other applications of this ability to recognize scale, including the multiscale effects of TV regularization and the rate of loss of image features of various scales as a function of increasing amounts of regularization. Several numerical results are given.

1 Introduction

Consider the problem of restoring a noise-contaminated or otherwise degraded image in $\mathbf{R}^n$: given a measured image $u_0(\vec{x})$, find an approximation $u(\vec{x})$ to the true image $u_{true}(\vec{x})$, where $u_0 = Ku_{true} + \eta$ and where $\eta(\vec{x})$ is the noise or other degradation in the image. The work in this paper results from the case in which the blurring operator K is the identity, in which case the problem could be considered one of filtering or denoising: $u_0 = u_{true} + \eta$. Typically our goal is to recover the true image u_{true} as exactly as possible and/or to find a new image u in which the information of interest is more obvious and/or more easily extracted.

1.1 Total variation regularization in image processing

Just over a decade ago, Rudin, Osher and Fatemi [22] proposed to modify the given image by decreasing the total variation

$$TV(u) \equiv \int |\nabla u(\vec{x})| \, d\vec{x} \tag{1}$$

in the image while preserving some fit to the original data u_0. Equation (1) is typically referred to as the *total variation* or *bounded variation* seminorm of u. There are two common formulations of this problem: the unconstrained or Tikhonov formulation [27],

$$\min_u \ \tfrac{1}{2}||u - u_0||^2 + \alpha \, TV(u) \ , \tag{2}$$

and the noise-constrained problem,

$$\min_u TV(u) \quad \text{subject to} \quad ||u - u_0||^2 = \sigma^2, \tag{3}$$

*This work was supported by grants from the NSF under contracts DMS-9973341, ACI-0072112, INT-0072863, the ONR under contract N00014-03-1-0888, the NIH under contract P20 MH65166, and the NIH Roadmap Initiative for Bioinformatics and Computational Biology U54 RR021813 funded by the NCRR, NCBC, and NIGMS.

†Department of Mathematics, Pepperdine University, `David.Strong@pepperdine.edu`

‡Department of Mathematics, UCLA, `aujol@math.ucla.edu`

§Department of Mathematics, UCLA, `chan@math.ucla.edu`

where the error or noise variance σ^2 is assumed to be known. As shown in [10], solving (2) is equivalent to solving (3). In this paper we consider primarily the unconstrained formulation (2). We note that throughout this paper $\|\cdot\| \equiv \|\cdot\|_{L^2}$.

In (2), $\alpha > 0$ is the regularization parameter that determines the balance between goodness of fit to the original image and the amount of regularization done to the original image u_0 in order to produce the approximation u. TV regularization produces a new image u which has less total variation than u_0, but with no particular bias toward a sharp (discontinuous) or smooth solution. When solving (2), the original image u_0 and the regularization parameter α determine the sharpness or smoothness of the new, regularized function u. Larger values of α result in more regularization and less goodness to fit of u of the original data u_0. This is illustrated in Figure 1. Although TV regularization was originally introduced for deblurring and denoising grayscale images, it has subsequently been employed in a variety of other image processing tasks such as denoising color or other vector-valued images [8], blind deconvolution [13], segmentation [28], inpainting [12], image decomposition [20], and upsampling [2].

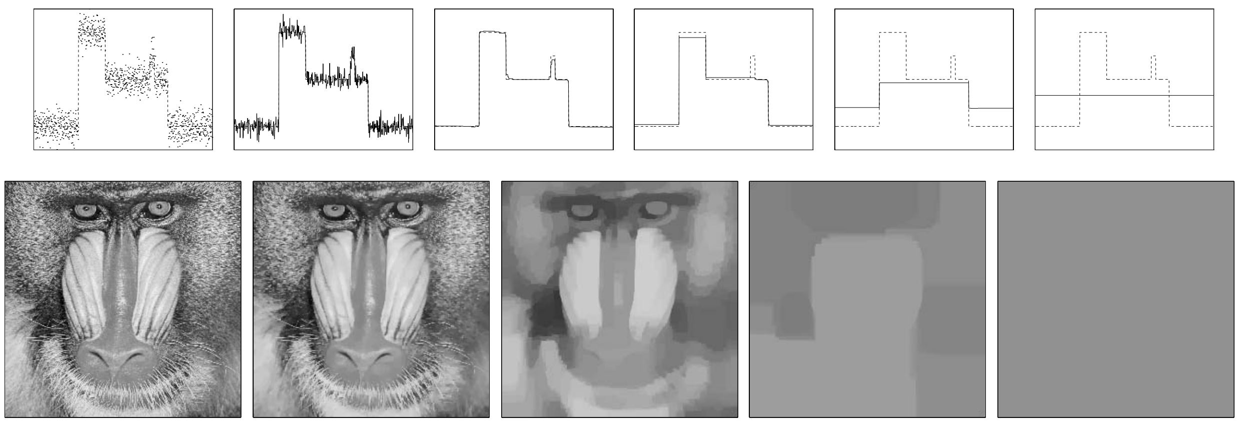


Figure 1: Results of TV regularization of a simple noisy $\mathbf{R}^1$ function (top row) and the standard Mandrill image (bottom row) when using different values of α in solving (2). For the $\mathbf{R}^1$ function, the first plot shows the true and noisy images. For the Mandrill image, the first image is the original (noise-free) image. For both, the subsequent figures are the results of solving (2) using $\alpha = 0.0001, 0.001, 0.01$ and 0.1 (and $\alpha = 1.0$ for the $\mathbf{R}^1$ function), respectively.

1.2 Scale recognition and choice of regularization parameter

Scale is inherently important both in understanding and in manipulating an image. At present the effects of TV regularization—in particular, how these effects relate to the scale of the various image features—are only partially understood. Additionally, how to choose the regularization parameter α when solving (2) is often done haphazardly or experimentally. In contrast, many researchers from a variety of perspectives have more thoroughly investigated other aspects of TV regularization, such as existence and uniqueness of solutions, development and convergence analysis of numerical schemes, and the basic effects of TV regularization on an image. A representative sampling of the literature includes [1], [3], [7], [10], [11], [14], [15], [16], [18], [22], [25] and [29].

If there is some regularity to the noise and if the noise level is known, then we can use (3) to solve the TV regularization problem (and the choice of α in (2) would be inherent). If noise is irregular and/or noise level is unknown, we must in some intelligent way choose a value for α. In the past, this has been done more by trial and error rather than by any well understood theory. While this might result in (indeed, the choice of α is still often driven by) an image which "looks nice," it is generally unclear precisely how the image itself has been affected, which leaves us wondering just how accurately the image produced by regularization represents the true image. Even in the case of known noise level and type, we may want to choose α based on criteria other than trying to match a noise constraint. Additionally, we may want to apply regularization to a noise-free image in order to more easily extract the desired information from the image.

It is clear that it would be helpful to have a more automatic, reliable and theory-based approach for choosing α. Also, applying TV regularization would be an even more mathematically sound and predictable approach to image processing if we better understood how the original image u_0 has been changed, particularly with respect to scale, in order to produce the regularized image u.

1.3 How α depends on the size of the image domain

In this paper, the image domains in $\mathbf{R}^1$ and $\mathbf{R}^2$ will be $[0,1]$ and $[0,1]^2 = [0,1]$ x $[0,1]$, respectively. We choose these unit domains because we prefer to have the scale of image features be consistent, regardless of the discretization (resolution) of the image. Because some readers may be more familiar with values of α in solving (2) for a discrete n x n image when the domain is taken to be $[0,n]^2$ rather than $[0,1]^2$, we give the following lemma as part of this introductory section.

Lemma 1 *If α is the regularization parameter used in solving (2) when the domain is $[0,1]^k, k \in \mathbf{Z}^+$, then $n\,\alpha$ is the value needed in solving (2) when the domain is $[0,n]^k$ in order to produce the same regularized image u.*

Proof Let $[0,1]^k$ be the unit hypercube in R^k, and similarly for $[0,n]^k$. Let $\vec{t} = n\vec{x}$; that is, $(t_1, t_2, ..., t_k) = n(x_1, x_2, ..., x_k)$. Then $d\vec{t} = n^k\, d\vec{x}$. Let u_0 and U_0 be the original image, if defined on $[0,1]^k$ and $[0,n]^k$, respectively. Similarly, let u and U be the regularized image (the solution to (2)) if defined on $[0,1]^k$ and $[0,n]^k$, respectively. That is, for $\vec{t} \in [0,n]^k$, $U_0(\vec{t}) = u_0(\frac{\vec{t}}{n}) = u_0(\vec{x})$ and $U(\vec{t}) = u(\frac{\vec{t}}{n}) = u(\vec{x})$. Since $\frac{\partial u(\vec{x})}{\partial x_i} = \frac{\partial U(\vec{t})}{\partial x_i} = \frac{\partial U(\vec{t})}{\partial t_i}\frac{\partial t_i}{\partial x_i} = n\frac{\partial U(\vec{t})}{\partial t_i}$, then

$$\nabla u(\vec{x}) = (\frac{\partial u(\vec{x})}{\partial x_1}, \frac{\partial u(\vec{x})}{\partial x_2}, ..., \frac{\partial u(\vec{x})}{\partial x_k}) = (n\frac{\partial U(\vec{t})}{\partial t_1}, n\frac{\partial U(\vec{t})}{\partial t_2}, ..., n\frac{\partial U(\vec{t})}{\partial t_k}) = n\,\nabla U(\vec{t}).$$

Then (2) on $[0,1]^k$ can be related to (2) on $[0,n]^k$ as follows:

$$\begin{aligned}
\tfrac{1}{2}||u-u_0||^2 + \alpha\, TV(u) &= \tfrac{1}{2}\int_{\vec{x}\in[0,1]^k} [u(\vec{x}) - u_0(\vec{x})]^2\, d\vec{x} + \alpha \int_{\vec{x}\in[0,1]^k} |\nabla u(\vec{x})|\, d\vec{x} \\
&= \tfrac{1}{2}\int_{\vec{t}\in[0,n]^k} [U(\vec{t}) - U_0(\vec{t})]^2\, \tfrac{1}{n^k}\, d\vec{t} + \alpha \int_{\vec{t}\in[0,n]^k} n\,|\nabla U(\vec{t})|\, \tfrac{1}{n^k}\, d\vec{t} \\
&= \tfrac{1}{n^k}\left\{\tfrac{1}{2}||U-U_0||^2 + n\,\alpha\, TV(U)\right\}
\end{aligned}$$

And, of course,

$$arg\ \min_U\ \tfrac{1}{n^k}\left\{\tfrac{1}{2}||U-U_0||^2 + n\,\alpha\, TV(U)\right\} = arg\ \min_U\ \tfrac{1}{2}||U-U_0||^2 + n\,\alpha\, TV(U).$$

■

Remark Changing the domain from $[0,1]^k$ to $[0,n]^k$ requires us to change α to $n\,\alpha$ in order to produce the same results—this is true for any k. For example, for a 256 x 256 image, if our domain is $[0,1]^2$ and $\alpha = 0.001$, then we would need $\alpha = 0.256$ if our domain were instead $[0,n]^2$ in order to produce the same regularized image.

Finally, in addition to choosing the unit hypercube as our domain, we note that all images in $\mathbf{R}^2$ are grayscale and, again for consistency, have been normalized so that the minimum and maximum image intensity values are 0 and 1, respectively.

1.4 Outline

In Section 2 we discuss how TV regularization naturally perceives scale in an image, including how this perception changes with increasing amounts of regularization (larger values of α) applied to the image. The main contributions of this paper are given in Sections 3 - 5. In Section 3, we motivate and give an algorithm for determining the minimum value of α in (2) that will result in the removal of all features of scale at or below (smaller than) any given threshold. Section 4 is devoted to relating Meyer's G norm to scale, to some degree a consequence of the algorithm given in the previous section, which gives us new insight and a more intuitive understanding of this norm. In Section 5 we give several numerical results of this algorithm. Finally, in Section 6, we begin to explore additional ways to employ TV regularization's ability to recognize scale, including to better understand both the multiscale effects of TV regularization and the rate at which features of any given scale disappear from an image as a function of α. Section 6 contains additional numerical results. Conclusions and other final remarks are given in Section 7.

2 Scale, as perceived by TV regularization

In this section we further develop the notion of scale introduced in [25]. We show how TV regularization naturally recognizes scale, how the notion of scale in TV regularization can be quantified on a pixel-by-pixel basis, and how perception of scale varies with α.

2.1 Scale and intensity change

As shown by Strong and Chan in [25], there are two fundamental properties of TV regularization:

1. Edge locations of image features tend to be preserved, and under certain conditions, are preserved exactly.

2. The intensity change δ experienced by an individual image feature Ω is inversely proportional to the *scale* of that feature,

$$\delta(\vec{x}) = \frac{\alpha}{scale(\vec{x})}, \tag{4}$$

   where we define

$$scale = \frac{|\Omega|}{|\partial\Omega|}. \tag{5}$$

Remark This notion of scale arises naturally in TV regularization, as described in [25], rather than simply being arbitrarily defined. At present there is discussion about other, more general—and also more mathematically abstract—ways of defining scale as it is perceived by TV regularization. For instance, one may define the scale of an object as being the radius of the largest ball which can be contained in the object. See also [23] for notions related to scale, as well as further discussions in [25]. The definition of scale in (5) is essentially a special case of these more general definitions. This definition (5) is intuitively simpler and is practically (as opposed to theoretically) more useful, and ultimately this particular definition of scale makes possible the results that we give in this paper.

Remark Property 1 is quite significant and is a primary reason TV regularization is used in a variety of image processing applications, such as those listed at the end of Section 1.1, not to mention its potential use in applications other than image processing. Property 2 explains in a very basic way how TV regularization works: smaller-scaled features (including noise) experience large reduction in intensity, thus removing or greatly reducing them by flattening them, while larger-scaled features experience relatively little intensity reduction and are consequently left more intact. This was seen in Figure 1. As Figure 1 also illustrates, a less than precise understanding of Property 2 can lead to undesirable results when using TV regularization.

Remark As described in [24], TV regularization can be viewed as a model or unbiased case of anisotropic diffusion, and consequently Property 2 is also one way of explaining how anisotropic diffusion works. We also note that Bellettini, Caselles and Novaga did a related analysis of TV regularization by considering the eigenvalue problem of $-\nabla \cdot (\frac{\nabla v}{|\nabla v|}) = v$. Details can be found in [7].

Equation (4) describes how change in intensity is a function of scale. When rewritten as

$$scale(\vec{x}) = \frac{\alpha}{\delta(\vec{x})}, \tag{6}$$

we see that scale can be viewed as a function of change in image intensity. Although simple—indeed, in part because it is so simple—this relationship is potentially very useful. Essentially what it means is that we can determine what the scales of the various image features are throughout the image by looking at how much intensity changes as a result of applying TV regularization to the image. Understanding how to measure scale as perceived by TV regularization potentially has many uses, including four that we will investigate in this paper:

1. We can find the smallest α needed to remove all features whose scale is less than any scale threshold.
2. We can give an intuitive explanation of Meyer's G norm by relating it to the above notion of scale.
3. We can better develop our understanding of how TV regularization can be used to produce multiscale representations of images.
4. We can begin to understand how quickly the various scales present in the image disappear for increasing values of α.

In this paper we will investigate the first application in detail. We will also consider the three other applications, but we expect that our results will be the beginning of more analysis of these ideas. In other words, we expect that more work can and will be done both by ourselves and others to further develop our understanding of these other aspects of TV regularization. A fifth promising application is that once TV regularization has been applied, we can determine the scales of the remaining features and using (4) and (6) we can determine how much intensity was lost due to TV regularization, and add back this lost intensity to the regularized image to get a more accurate approximation u of the true image u_{true}. This fifth application turns out to be a bit more complicated than it might first seem, and consequently it is being investigated in a separate paper.

2.2 Scale of piecewise constant features

The notion of scale defined in (5) may at first be unclear or even confusing to the reader to whom it is new. To make it easier to understand, we explain it for two simple examples. A circle of radius r would have $scale = \pi r^2 \,/\, 2\pi r = r/2$ that is linearly proportional to radius r. The scale of a sphere would also be increasing linearly in r. Second, a rectangle of k_1 x k_2 pixels on an n x n discretized grid of the unit square [0,1] x [0,1] would have $scale = k_1 k_2 \,/\, 2n(k_1 + k_2)$. Consequently, a k x k square has the same scale as a rectangle of width $k/2$ and infinite length. In general, large, blocky features have relatively large scale, while thin features—even those that are very long—have relatively small scale. This fact was one of the main results of [15], in which Dobson and Santosa use a Fourier analysis to show that TV regularization is particularly suited to denoising images comprised of large, blocky features.

2.3 Determining scale using the scale recognition probe

Using (6), we can compute the scale, as perceived by TV regularization, in an image. We accomplish this by performing what we refer to as a *scale recognition probe* for determining $scale_u(\vec{x})$ in image u:

Scale Recognition Probe Algorithm

1. Choose α_{probe}
2. Find $u_{probe} = arg \min_{\check{u}} \ \frac{1}{2}||\check{u} - u||^2 + \alpha_{probe} TV(\check{u})$
3. Compute $\delta(\vec{x}) = |u_{probe}(\vec{x}) - u(\vec{x})|$
4. Compute $scale_u(\vec{x}) = \dfrac{\alpha_{probe}}{\delta(\vec{x})}$

To illustrate the scale recognition probe, we apply the above algorithm to the simple image shown in Figure 2. In (a) is the noise-free image in which to find scale. In (b) is the image showing scales computed using the scale recognition probe, and in (c) is the image showing scales as predicted by (5). In (b) and (c) we assign the background a value of 0, to more easily see the scales of the shapes.

Remark As seen in the image in (c), the corners of the squares and the corners and ends of the rectangles experience a slightly greater change in intensity, and thus are interpreted as having smaller scale. The definition of scale and the change in intensity (4) is exact for radially symmetric features. From this point of view, the corners of each square, for example, are like smaller-scaled features attached to a larger one: like four small circles connected at each "corner" of a larger circle.

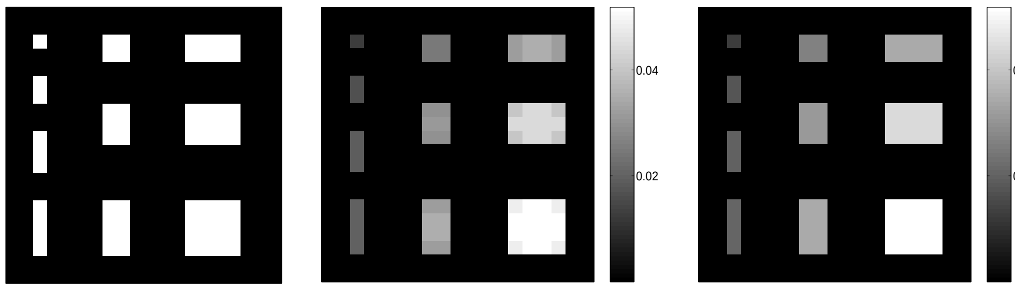


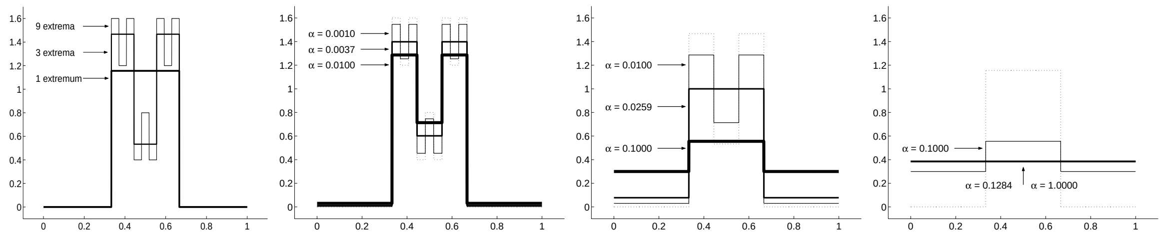
Figure 2: The scales of individual features as perceived by TV regularization, found by applying the scale recognition probe. The first image is the noise-free image in which to recognize scale. The second image is the image showing computed scales found using the scale recognition probe, and the third image is the image showing theoretically predicted scales found using (5).

2.4 Perception of scale dependent on amount of regularization done

When using TV regularization, there are two natural ways to recognize the various levels of scale in an image: first is by simple inspection; second, and more interestingly and usefully, is how TV regularization will perceive scale. We consider how this perception is affected by the choice of α. The value of α chosen in solving (2) determines how much regularization of u_0 occurs in producing u, and thus what scales remain in the regularized function. More precisely, it is well known that for increasing values of α there is increasing loss of smaller scale in the image. We look at this in more detail in the following section.

2.4.1 A simple example of evolution of scale perception

Consider the function labeled as "9 extrema" in Figure 3(a). There are three levels of scale at which this function could be viewed: at the finest level is the actual function, at the next level is the "3 extrema" function, and at the coursest level is the "1 extremum" function. At successively courser levels, the value of the function in each region is simply the mean of the values over the subregions found in the finer levels. Let $\alpha_{9\to3}$ be the value of α at which the 9-extrema function transitions into the 3-extrema function, as seen in (b). It turns out that for $\alpha \geq \alpha_{9\to3}$, TV regularization perceives the function as the 3-extrema function. This is illustrated by the fact that the function produced using $\alpha = 0.0100$ in (b), in which u_0 was the (true) 9-extrema function, is identical to the function produced using $\alpha = 0.0100$ in (c), when using the 3-extrema version of u_0 to solve (2). Similarly, the function produced using $\alpha = 0.100$ in (c) in which u_0 was the 3-extrema function (or equivalently, if the u_0 used were the original 9-extrema function itself), is identical to the function produced using $\alpha = 0.100$ in (d), when using the 1-extremum version of u_0 to solve (2). This illustrates that for $\alpha \geq \alpha_{3\to1}$, TV regularization perceives this function as the 1-extremum function seen in (c). For $\alpha \geq \alpha_{1\to0}$, the resulting regularized image will simply be the constant image shown in (d). It is not difficult to analytically predict what these transitional values of α should be, as well as the behavior of this simple function for other values of α. Indeed, we found analytically, rather than empirically, the values for $\alpha_{9\to3}$, $\alpha_{3\to1}$ and $\alpha_{1\to0}$ given in Figure 3. For brevity we omit the details.



(a) The original function, when perceived as having 9, 3 or 1 extrema.

(b) The evolution from a 9-extrema function to a 3-extrema function. The transition occurs at $\alpha \approx 0.0037$.

(c) The evolution from a 3-extrema function to a 1-extremum function. The transition occurs at $\alpha \approx 0.0259$.

(d) The evolution from a 1-extremum function to a constant function. The transition occurs at $\alpha \approx 0.1284$.

Figure 3: In $\mathbf{R}^1$, the scale present throughout the function, as perceived by TV regularization, increases as α increases. In (a) is the original function, when perceived as having 9, 3 or 1 extrema. For $0 \le \alpha < 0.0037$, the function is perceived as the original 9-extrema function, as seen in (b). For $0.0037 \le \alpha < 0.0259$, the function is perceived as the 3-extrema function, as seen in (b) and (c). For $0.0259 \le \alpha < 0.1284$, the function is perceived as the 1-extrema function, as seen in (c) and (d). For $0.1284 \le \alpha$, the function is perceived as the constant-valued function in (d).

2.4.2 Generalization

The above example helps explain the well known fact that, in general, any image in $\mathbf{R}^n$ will gradually evolve into an image with larger scales—that is, with less detail, as smaller-scaled features are lost—as the value of the regularization parameter α increases. Of course, in general the transition from one scale to another does not occur at a few distinct values of α. Indeed, these transitions are more continuous: for most images, at various locations and for various values of α this transition is continually occuring as α increases. Also, images are not comprised only of piecewise constant features, although in the discrete case an n x n image has n^2 pixels, each with a particular value, so in this sense the image could be thought of as being piecewise constant, albeit on a very fine scale. Consequently, the notion of scale is more complicated than thus far discussed. The analysis in this paper helps (but does not exhaustively) develop a precise understanding of how TV regularization perceives scale in an image and how TV regularization resolves an image into its various scales. It turns out that this relatively simple notion of scale is sufficient (and, indeed, necessary) to obtain the results we give later.

Our goal in the next section will be to determine the value of α needed to remove all features at or below a certain scale. In order to best preserve the other wanted features, we want to find the minimum such value of α, and we would like to find it quickly and accurately.

3 Selection of regularization parameter

We now consider applications of our understanding of TV regularization's recognization of scale in an image. The first application is the specific task of removing from an image all features whose scales are at or below (smaller than) a specific threshold, while leaving all other features as intact as possible. That is, we would like to find the value of α in solving (2) that is just large enough to result in removing all features below $scale_{thresh}$, but no larger. We denote this particular value as $\tilde{\alpha}$. Where $scale_u(\vec{x})$ is the scale of $u = arg \min_u \frac{1}{2}||u - u_0||^2 + \alpha TV(u)$, we define

$$\tilde{\alpha} = \min\{\alpha : \min_{\vec{x}} scale_u(\vec{x}) > scale_{thresh}\}. \tag{7}$$

3.1 An example of what we want to accomplish

As a simple example of what we want to accomplish, we apply TV regularization to the image shown in Figure 4. This image contains checkboard "texture" of two smaller scales and shapes of four larger scales. The first image is the original image, while the next six are the images in which we have removed all features at or below six different scale thresholds, corresponding to the six different scales present in the image. The values of $\tilde{\alpha}$ corresponding to each of the six scale thresholds are given in the caption of Figure 4. These values were found using the algorithm that we subsequently give in Section 3.3.

Remark To be clear, the values of $\tilde{\alpha}$ and the corresponding images given in Figure 4 were not found experimentally, i.e. by choosing a sequence of α values and looking at the resulting images in order to see where the different features of varying scales are completely removed. This process, given in Section 3.3, was automatic and was accomplished as a result of our ability to recognize scale.

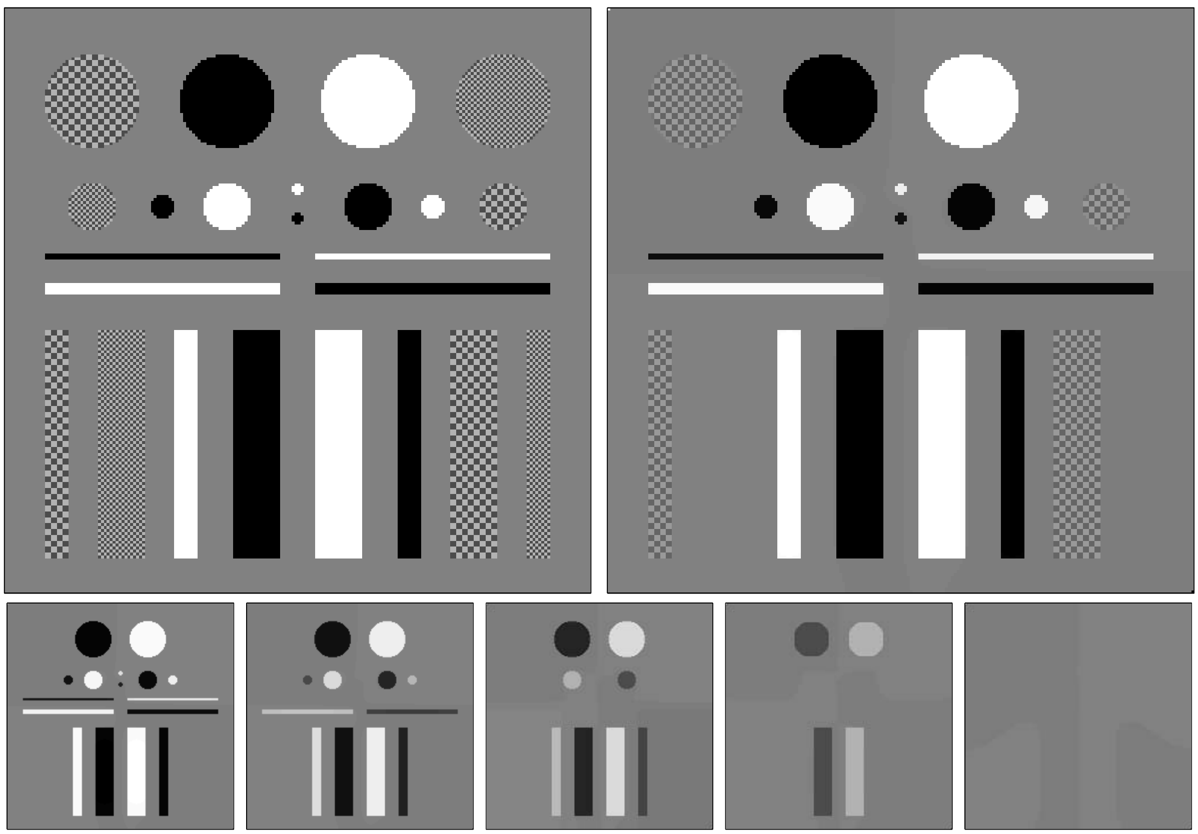


Figure 4: Results of solving (2) using values of α that are just large enough to remove all features at or below specific scale thresholds. The first image is the original image, and the subsequent images are regularized images found by solving (2) using α-values of 0.00025, 0.00050, 0.00239, 0.00495, 0.00990 and 0.01550. These are the value of $\tilde{\alpha}$ resulting from using six scale thresholds corresponding to the six distinct scales present in the image, respectively: 0.00125, 0.00250, 0.00500, 0.01000, 0.02000 and 0.04000. These $\tilde{\alpha}$ values were found automatically, using the $\tilde{\alpha}$ Algorithm given subsequently in Section 3.3. The first two images are larger in order to better see the effects of regularization on the artificial texture. The intensities of the objects are the actual intensities; no rescaling has been done to enhance contrast.

3.2 Basic strategy for finding $\tilde{\alpha}$

3.2.1 Bisection method

The strategy we use to find $\tilde{\alpha}$ is essentially the bisection method, where the desired "root" is $\tilde{\alpha}$, as defined in (7). We will describe why this approach to finding $\tilde{\alpha}$ is a natural one. In using the bisection method, there are two questions. First is the question of how to choose the initial lower and upper bounds on our estimate for $\tilde{\alpha}$, which we denote α_{min} and α_{max}. The simplest choice for α_{min} is 0, since by definition $\alpha \geq 0$. In choosing α_{max}, it is of course necessary that $\tilde{\alpha} \leq \alpha_{max}$. The choice of α_{max} will depend on $scale_{thresh}$ and on the image itself. We revisit how to choose α_{max} when we consider our first numerical example in Section 5.1.

For each iteration i, we find $u_i = arg \min_u \frac{1}{2}||u - u_0||^2 + \alpha_i TV(u)$ where $\alpha_i = (\alpha_{min} + \alpha_{max})/2$. The second question then is what criteria to use in deciding which of the two subintervals $[\alpha_{min}, \alpha_i]$ or $[\alpha_i, \alpha_{max}]$ to move to after iteration i. That is, we need to determine whether $\tilde{\alpha} \leq \alpha_i$ or $\tilde{\alpha} > \alpha_i$. Our task is to determine if there are any features or portions of features in u_i with scale at or below $scale_{thresh}$. To do this we perform a scale recognition probe, as described in Section 2.3, to find the scale in u_i. Once we find $scale_{u_i}(\vec{x})$, we want to determine whether $scale_{u_i}(\vec{x}) \leq scale_{thresh}$ anywhere $\vec{x}$ in the image. If so, then our choice α_i is too small and we should move to the upper half of the interval $[\alpha_i, \alpha_{max}]$. If not, our choice was sufficiently large, and since $\tilde{\alpha}$ is the smallest of all such values of α, we know that $\tilde{\alpha} \leq \alpha_i$, in which case we move to the lower half of the interval $[\alpha_{min}, \alpha_i]$.

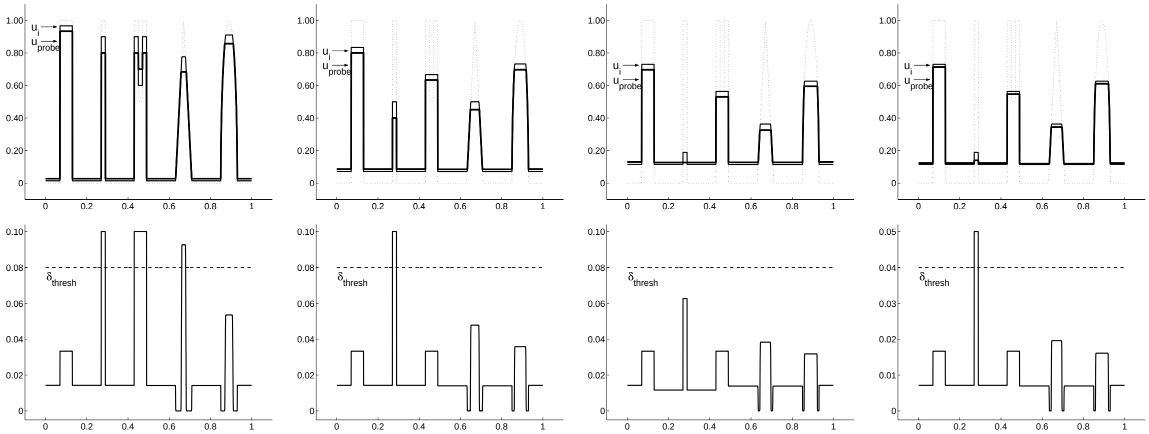
Conceptually, we want to compare $scale_{u_i}(\vec{x})$ to $scale_{thresh}$. Unfortunately, if $\delta(\vec{x}) = 0$ anywhere in the image, we end up dividing by 0. We avoid this by instead simply comparing $\delta(\vec{x})$ to δ_{thresh} where $\delta_{thresh} = \alpha_{probe}/scale_{thresh}$. Since $scale_{u_i}(\vec{x}) \leq scale_{thresh} \Longleftrightarrow \delta(\vec{x}) \geq \delta_{thresh}$, if $\delta(\vec{x}) \geq \delta_{thresh}$ anywhere $\vec{x}$ in the image, then there are still features at or below $scale_{thresh}$, in which case we need to increase the value of α by moving to $[\alpha_i, \alpha_{max}]$; otherwise we move to $[\alpha_{min}, \alpha_i]$.

3.2.2 A simple illustration of this approach

In this subsection we illustrate our approach on a simple example, as seen in Figure 5. Each of (a) - (d) includes a plot of the original function u_0, the regularized function u_i for a given value of α_i, and the second regularized function u_{probe}, found by applying regularization to u_i using α_{probe} when performing the scale recognition probe. To be clear, these images are not the results of the first three steps of the bisection procedure just described above—they are simply the results of three possible choices of α_i. The second plot in each subfigure is a plot of $\delta(\vec{x}) = |u_{probe}(\vec{x}) - u_i(\vec{x})|$, the change in intensity due to the scale recognition probe. We remind the reader that larger δ means smaller *scale*, and inversely. For more clarity, the example involves a noise-free, mostly piecewise constant function in $\mathbf{R}^1$.

Observation For smaller values of α, the smaller features are still present and their scales are recognized. Notice in the top plot in (a) the 3-extrema feature at $x = 0.45$. For small α_i, the feature is recognized (by comparing $\delta(\vec{x})$ to δ_{thresh}) as having three small extrema, while for larger α this feature is perceived as being one larger extrema, as we see in (b) - (d). Next consider the non-piecewise constant features: the triangular and the semicircular features located at approximately $x = 0.65$ and 0.90. Notice that for smaller values of α, as in (a), these are perceived as having smaller scale (larger δ), corresponding to the top of each feature, while for larger α each feature is more flattened and thus each feature is perceived as having larger scale corresponding to the lower part of the feature, as is seen in (b) - (d).

Observation Consider the extremum located at 0.30. In (a) and (b) this feature is recognized as having $scale < scale_{thresh}$ ($\delta > \delta_{thresh}$). Notice that δ for this feature is the same in both cases. In (c), clearly there is some of this feature still remaining, but it turns out there is not enough of it (that is, its intensity relative to the intensity on either side of it is not large enough) to result in $\delta > \delta_{thresh}$, thus it is not recognized as having a scale smaller than $scale_{thresh}$. In (d) we have exactly the same u_i, but this time we use a value of α_{probe} that is half of the α_{probe} used in (c)—notice that the limits on the vertical axis in (d) are exactly half of those in (c). In (d) we see that this time the scale of this feature is accurately computed, which results in the expected $\delta > \delta_{thresh}$ for this feature, as seen in (a) and (b), while for the other features the change



(a) For smaller α, all details and small-scaled features, including those at positions 0.45, 0.65 and 0.90, are present.

(b) For larger α, some features evolved from smaller to larger scale, including at 0.45, 0.65 and 0.90.

(c) The true scale of the feature at 0.3 is not recognized, when using the current value of α_{probe}.

(d) The true scale of the feauture at 0.3 is now recognized, as a result of choosing a smaller α_{probe}.

Figure 5: For increasing values of α, smaller scaled features are lost, as seen in (a) - (c). The ability to accurately measure scale depends on the value of α_{probe} used in the scale recognition probe: smaller α_{probe} makes it more likely that scale will be accurately measured, as seen in comparing (c) and (d). The top plots include the true and regularized functions, and the function resulting from the scale recognition probe. The bottom plots show the change in intensity $\delta = |u_i - u_{scale}|$ due to the scale recognition probe.

in intensity relative to δ_{thresh} is still the same as in (c). This helps explain why we need to choose α_{probe} small, as we will also later confirm analytically with Lemma 3 in Section 4.

3.2.3 A theoretical definition of scale

It turns out that given any u in which to determine scale, in performing the scale recognition probewe could find a value of α sufficiently small to accurately determine scale throughout the image. However, the smaller α_{probe} is, the more precise we would need to be in computing u_{probe}, which is computationally more expensive. Moreover, given any value of α_{probe}, we could contrive an image in which the scale would not be accurately measured using that particular value of α_{probe}. In the end, we could redefine (6) as

$$scale_u(\vec{x}) = \lim_{\alpha_{probe} \to 0^+} \frac{\alpha_{probe}}{\delta(\vec{x})}, \tag{8}$$

where $\delta(\vec{x})$ is the change in intensity due to the scale recognition probe.

3.2.4 Estimate interval for $\tilde{\alpha}$

In trying to determine $\tilde{\alpha}$, it turns out that where $[\alpha_{min}, \alpha_{max}]$ is the current estimate interval for $\tilde{\alpha}$, then in reality the upper bound on $\tilde{\alpha}$ is $\tilde{\alpha} \leq \alpha_{max} + \alpha_{probe}$. This is clearly illustrated in (c) and (d) of Figure 5. Similarly, it turns out that the lower bound on $\tilde{\alpha}$ is $\alpha_{min} + \alpha_{probe}$ rather than α_{min}. Thus rather than $[\alpha_{min}, \alpha_{max}]$, the interval $[\alpha_{min} + \alpha_{probe}, \alpha_{max} + \alpha_{probe}]$ is the true estimate interval for $\tilde{\alpha}$. Note, of course, that the maximum possible absolute error for our estimate of $\tilde{\alpha}$ is still $\alpha_{max} - \alpha_{min}$ in either case, and the maximum possible relative error $(\alpha_{max} - \alpha_{min})/(\alpha_{min} + \alpha_{probe})$ will still be approximately the same as $(\alpha_{max} - \alpha_{min})/\alpha_{min}$ if $\alpha_{probe} \ll \alpha_{max}$. Lemma 3 in Section 4.3 further explains why we need $\alpha_{probe} \ll \alpha_{max}$. In our subsequent numerical examples, we have arbitarily chosen $\alpha_{probe} = \alpha_{max}/100$.

Observation If our current estimate interval for $\tilde{\alpha}$ were $[\alpha_{min}, \alpha_{max}]$, then in (a), (b) and (d) of Figure 5 we would increase α (that is, move to $[\alpha_i, \alpha_{max}]$), while in (c) we would decrease α (move to $[\alpha_{min}, \alpha_i]$).

3.3 The $\tilde{\alpha}$ Algorithm

We now give the complete algorithm for finding $\tilde{\alpha}$.

$\tilde{\alpha}$ Algorithm

1. Choose α_{min}, α_{max}, $scale_{thresh}$
2. Initialize $i = 1$, $\alpha_1 = (\alpha_{min} + \alpha_{max})/2$,
   $\alpha_{probe} = \alpha_{max}/100$, $\delta_{thresh} = \alpha_{probe}/scale_{thresh}$
3. Repeat steps a - d until $error \leq$ desired tolerance
   - 3a. $u_i = arg \min\limits_u \ \frac{1}{2}||u - u_0||^2 + \alpha_i \, TV(u)$
     $u_{probe} = arg \min\limits_u \ \frac{1}{2}||u - u_i||^2 + \alpha_{probe} \, TV(u)$
   - 3b. $\delta_{max} = \max\limits_{\vec{x}} |u_{probe}(\vec{x}) - u_i(\vec{x})|$
   - 3c. If $\delta_{max} \geq \delta_{thresh}$
     $\alpha_{min} = \alpha_i$
     else
     $\alpha_{max} = \alpha_i$
   - 3d. Update: $i = i + 1$, $\alpha_i = (\alpha_{min} + \alpha_{max})/2$,
     $\alpha_{probe} = \alpha_{max}/100$, $\delta_{thresh} = \alpha_{probe}/scale_{thresh}$
4. $\tilde{\alpha} = \alpha_{max} + \alpha_{probe}$

Remark We allow α_{probe} to vary with α_{max}, to ensure that $\alpha_{probe} \ll \alpha_{max}$. In the above algorithm we update α_{probe} at each step to be $\alpha_{probe} = \alpha_{max}/100$. Also, to account for numerical imprecisions, to be more conversative one might choose δ_{thresh} to be slightly smaller than the theoretical δ_{thresh}. In the results given in the subsequent figures we compare δ_{max} with $0.95\,\delta_{thresh}$. Finally, the error in Step 3 could be either the absolute or relative error.

Prior to giving numerical results of this algorithm, in the following section, we carry out a mathematical study of the $\tilde{\alpha}$ Algorithm.

4 Mathematical analysis of the $\tilde{\alpha}$ Algorithm

In this section we analyze the $\tilde{\alpha}$ Algorithm from the perspective of the G norm introduced by Meyer in [19]. Essentially, we show how our notion of scale helps give an intuitive interpretation of the G norm and conversely how this norm gives some enlightening insight into the $\tilde{\alpha}$ Algorithm.

4.1 Meyer's G norm

Recently, Meyer did an interesting mathematical analysis of the Rudin-Osher-Fatemi model in [19]. He introduced a new space, the G space, to model oscillating patterns:

Definition 1 *G is the Banach space composed of the distributions f which can be written*

$$f = \partial_1 g_1 + \partial_2 g_2 = \mathrm{div}\,(g) \tag{9}$$

with g_1 and g_2 in L^∞. On G, the following norm is defined:

$$\|v\|_G = \inf \left\{ \|g\|_{L^\infty} \; : \; v = \mathrm{div}\,(g), \; g = (g_1, g_2), \; g_1 \in L^\infty, \, g_2 \in L^\infty, \; |g(x)| = \sqrt{|g_1|^2 + |g_2|^2}(x) \right\} \tag{10}$$

See [5] for an analysis of G in a discrete setting and [4] for a generalization of Meyer's definition to bounded domains in a continuous setting. We will use the following ball in G ($\alpha > 0$):

$$G_\alpha = \{v \in G : \|v\|_G \leq \alpha\}. \tag{11}$$

We consider the discrete setting. It is shown in [5] that G is then the set of functions with zero mean. The following Lemma will prove to be useful:

Lemma 2 *If f belongs to G (i.e. if f is of zero mean), then*

$$\frac{1}{4n}\|f\|_{L^\infty} \leq \|f\|_G \leq 4n\|f\|_{L^\infty} \tag{12}$$

Proof In [5], it is shown that there exists g such that $f = \mathrm{div}\, g$ and $\|f\|_G = \|g\|_{L^\infty}$. It is easy to check that $\|\mathrm{div}\,\|_{L^\infty} \leq 4n$, which gives the right-hand side inequality in (12). Since the identity $I = \mathrm{div}^{-1}\,\mathrm{div}$, we have $1 \leq \|\mathrm{div}^{-1}\|_{L^\infty}\|\mathrm{div}\,\|_{L^\infty}$, from which we get the left-hand side of (12).

■

Remark It is a standard result that in a finite dimensional normed space, all of the norms are equivalent. Lemma 2 gives the equivalence constants explicitly. $1/n$ is the discretization step if our image is n x n on the unit square: it is clear that as $n \to +\infty$, then the G norm and the L^∞ norm are no long equivalent norms.

4.2 Relating the G norm and ROF model

Let us consider the ROF problem (2). The following proposition is shown in [9] (the proof is based on convex analysis):

Proposition 1 *The solution to (2) is given by*

$$u = u_0 - P_{G_\alpha}(u_0) \tag{13}$$

where $P_{G_\alpha}(u_0)$ denotes the orthogonal projection (with respect to the L^2 scalar product) of u_0 on G_α, defined by (11).

In [19], Meyer introduced the G norm to analyze the mathematical properties of the ROF model. One of the main results of [19] happens to be a straightforward corollary of Proposition 1. We first define

$$\hat{\alpha} = \|u_0 - \bar{u}_0\|_G, \tag{14}$$

and then give the corollary.

Corollary 1 *Where u is the solution of (2) and $\bar{u}_0$ is the mean of u_0, then we have:*

- *If $\alpha \leq \hat{\alpha}$, then $\|u - u_0\|_G = \alpha$.*
- *If $\alpha \geq \hat{\alpha}$, then $u = \bar{u}_0$.*

Remark It is well known that for any image there is a finite α-value above which the solution to (2) is simply the mean of the original image u_0. Thus we see intuitively that $\hat{\alpha} = ||u_0 - \bar{u}_0||_G$ is precisely this value of α. As we can see, the behavior of the ROF model is closely related to the G norm of the initial data u_0. Before now, there has been no easy or intuitive interpretation of the G norm.

Let us again consider (6), which links scale to α. When rewritten as

$$\alpha = \delta\, scale \tag{15}$$

we see that the G norm in Corollary 1 is proportional to scale. We give a rough explanation of Corollary 1:

- If u_0 has features with scale larger than $\delta\, scale$, then $u - u_0$ contains all the features with scale smaller than $\delta\, scale$.
- If all the features in u_0 are of scale smaller than $\delta\, scale$, then $u = \bar{u}_0$.

This confirms the analysis of the ROF model in [25] based on scale.

Remark There is another way to see that the G norm is closely related to the notion of scale. We can see it through the algorithms used to compute them. In this paper, we have presented the $\tilde{\alpha}$ Algorithm to compute the parameter α in (2) to remove all the features with scale equal to or smaller than a given threshold. Thanks to (15), we see that this essentially amounts to constraining the residual $u - u_0$ to be such that $||u - u_0||_G = \delta\, scale$. However, as far as we know, the only algorithm that has been proposed to compute the G norm of an image is the one introduced in [6]. This algorithm is also based on the bisection method: one compares u with $P_{G_\alpha}(u)$, that is one checks if all the features in u are smaller than $\delta\, scale$.

4.3 Mathematical study of the $\tilde{\alpha}$ Algorithm

In this subsection, we give more theoretical insight into the $\tilde{\alpha}$ Algorithm. We take the same notations as in the description of the $\tilde{\alpha}$ Algorithm.

Proposition 2 *If $scale_{thresh} < 1/4n$, then the $\tilde{\alpha}$ Algorithm will return $\tilde{\alpha} = \alpha_{probe}$.*

Remark The aim of this proposition is simply to confirm that the $\tilde{\alpha}$ Algorithm does what we would expect it to do in one extreme case. Indeed, $1/4n$ is the smallest available scale in the image: it is the scale of a single pixel when the image is n x n and the domain is the unit square. If we choose $scale_{thresh} < 1/4n$, than we expect to keep all the features of the original image. Since α_{probe} decreases to 0, this is precisely what the $\tilde{\alpha}$ Algorithm does.

Proof of Proposition 2 From Step 2b of the $\tilde{\alpha}$ Algorithm and Proposition 1, we have

$$u_{probe} = u_i - P_{G_{\alpha_{probe}}}(u_i). \tag{16}$$

We recall that $\delta_{max} = ||u_{probe} - u_i||_{L^\infty}$. From (16) and Lemma 2, we deduce that

$$\frac{1}{4n}\delta_{max} \leq ||P_{G_{\alpha_{probe}}}(u_i)||_G \leq 4n\, \delta_{max}. \tag{17}$$

But by defintion, we know that $||P_{G_{\alpha_{probe}}}(u_i)||_G \leq \alpha_{probe}$. We thus get

$$\delta_{max} \leq 4n||P_{G_{\alpha_{probe}}}(u_i)||_G \leq 4n\, \alpha_{probe}. \tag{18}$$

Using the fact that $\alpha_{probe} = \delta_{thresh}\, scale_{thresh}$, we deduce that

$$\delta_{max} \leq scale_{thresh}\, \delta_{thresh}\, 4n. \tag{19}$$

Since we assume that $scale_{thresh} < \frac{1}{4n}$, we then get $\delta_{max} < \delta_{thresh}$.

∎

The following result helps to further explain why α_{probe} needs to be small.

Lemma 3 *For $i \geq 1$, let us denote by $\hat{\alpha}_i = ||u_i - \bar{u}_0||_G$ (with the notations of the $\tilde{\alpha}$ Algorithm). If $\alpha_{probe} \geq \hat{\alpha}_i$, then we have $\delta_{probe} = \delta_{\hat{\alpha}_i}$.*

Proof From Step 2a of the $\tilde{\alpha}$ Algorithm and Proposition 1, we have:

$$u_i = u_0 - P_{G_{\alpha_i}}(u_0). \tag{20}$$

From Corollary 1, we know that exactly one of the following statements (i) or (ii) holds:

(i) If $\alpha_{probe} \leq \hat{\alpha}_i$, then $\|u_{probe} - u_i\|_G = \alpha_{probe}$.

(ii) If $\alpha_{probe} \geq \hat{\alpha}_i$, then $u_{probe} = \bar{u}_0$.

If $\alpha_{probe} \geq \hat{\alpha}_i$, then we will have $\delta_{probe} = \delta_{\hat{\alpha}_i}$.

■

Remark As a direct consequence of Lemma 3, we see that if $\alpha_{probe} \geq \hat{\alpha}_i$, then (6) cannot be used to compute the scale anymore. Therefore, if we want to check if features with a given scale are still present in u_i, then we need to have $\alpha_{probe} \leq \hat{\alpha}_i$.

Remark The result of Lemma 3 is illustrated by (c) and (d) in Figure 5, which illustrates why α_{probe} needs to be small. On the other hand, the smaller α_{probe} is, the more accurate we need to be when computing u_{probe}. As previously mentioned, we arbitrarily choose $\alpha_{probe} = \alpha_{max}/100$ in our implementation of the $\tilde{\alpha}$ Algorithm.

This ends our mathematical analysis of the $\tilde{\alpha}$ Algorithm. In the following two sections we turn our attention to numerical results of the $\tilde{\alpha}$ Algorithm and to other ways in which to exploit our understanding of how TV regularization recognizes scale in an image.

5 Numerical results of the $\tilde{\alpha}$ Algorithm

We now give some examples of applying the $\tilde{\alpha}$ Algorithm to both noise-free and noisy images.

5.1 A detailed look at the $\tilde{\alpha}$ Algorithm

We first apply the $\tilde{\alpha}$ Algorithm to the Mandrill image shown in Figure 6. In this example we wish to find the value of $\tilde{\alpha}$ that will result in the removal of all features of scale less than or equal to the scale of a single pixel. Of course, larger features will also be affected by the regularization, and some may even be removed, depending on their initial intensity levels and contrast with surrounding features. This image is 256 x 256 with the domain being the unit square, thus $scale_{thresh} = |\Omega|/|\partial\Omega| = (1/n^2)\,/\,(4/n) = 1\,/\,4n = 1/1024$. We choose $\alpha_{min} = 0$. The intensity of the image is normalized to be between 0 and 1, thus we choose α_{max} to be large enough to completely change the intensity of a single pixel by 1. Using (4), $\alpha_{max} = \delta_{max}\, scale_{thresh} = 1 \cdot 1/1024 \approx 0.000977$.

As this is the first time we have seen this algorithm in action, it is enlightening to see what each iteration of the algorithm produces: both the value of α_i (the estimate for $\tilde{\alpha}$ for each iteration) and the corresponding regularized image. The first figure in Figure 6 is the plot of the α_{min}, α_i and α_{max} values for each iteration. Next is the orginal (noise-free) image and the images corresponding to the first few α_i values found by the algorithm. Subsequent images appear virtually identical and are omitted. As seen in the plot of α values and as observed in the images themselves, most of the change occurs within the first few iterations, particularly if good initial values of α_{min} and α_{max} are chosen.

We can find as precise an estimate for $\tilde{\alpha}$ as wanted. Where α_i is our estimate at iteration i for $\tilde{\alpha}$, and where α_{min} and α_{max} are the initial lower and upper bounds for the $\tilde{\alpha}$ Algorithm, then after each iteration we have a bound on the absolute error of $|\alpha_i - \tilde{\alpha}| \leq (\frac{1}{2})^i(\alpha_{max} - \alpha_{min})$, and similarly for the relative error. This additional precision comes at a numerical price and is normally unnecessary, given the resulting insignificant amount of change in the image. The final estimate for $\tilde{\alpha}$ is 0.00052.

Figure 6: Results of applying the $\tilde{\alpha}$ Algorithm to the Mandrill image, where $scale_{thresh}$ is the scale of a single pixel. First is a plot of the values of α_{min}, α_i and α_{max} produced by the $\tilde{\alpha}$ Algorithm. Next is the original image, followed by the images corresponding to the first three values of α_i found by the $\tilde{\alpha}$ Algorithm. The final estimate for $\tilde{\alpha}$ is 0.00052.

5.2 Results of the $\tilde{\alpha}$ Algorithm for noise-free images

We next give results, both the values found for $\tilde{\alpha}$ and the corresponding images, of applying the $\tilde{\alpha}$ Algorithm to three standard (noise-free) images using scale thresholds of 2^{k-1} x 2^{k-1} pixels for $k = 1$ to 7 (e.g. for $k = 1$, the scale threshold is a single pixel). For an n x n image on the unit square, these correspond to scales of $2^{k-1}/4n$ for $k = 1$ to 7. Of course, these scales given for square features correspond to a variety of non-square features. For example, the scale of an 8 x 8 pixels square is also the scale of both a circle of radius 4 pixels and a rectangle of 4 pixels width and infinite length. Results are given in Figure 7 and Table 1. The first images in each set are a bit larger in order to better see the loss of fine detail. The Mandrill and Toys images are both 256 x 256, while the Canaletto image is 512 x 512. Consequently, the scale thresholds for the Mandrill and Toys images range from 1/1024 to 1/16 while the scales thresholds for the Canaletto image range from 1/2048 to 1/32, as seen in Table 1.

Remark In all three cases, it seems that a significant amount of regularization was necessary even for this smallest possible scale threshold, the scale of a single pixel. This is seen in comparing the first and second images (the original, and the result of using $scale_{thresh}$ of one pixel) in each set of images. Later, in Section 6.2 in which we briefly consider the multiscale effects of TV regularization, we look at the results of TV regularization applied to the Mandrill image when using $\alpha = 0.1\tilde{\alpha}, 0.2\tilde{\alpha}, ..., \tilde{\alpha}$ where $\tilde{\alpha} = 0.00052$ is the value of α found earlier when applying the $\tilde{\alpha}$ Algorithm where the scale threshold corresponded to a single pixel.

Remark The results seen in Figure 7 are not as dramatic as those seen in Figure 4. This is expected, as for these images we have not attempted to choose scale thresholds corresponding to specific scales present in the images, as we had done in obtaining the results of Figure 4. Still, for each of the three images in Figure 7, there are a number of specific features which are obviously present in a few of the images in the sequence, but then disappear once a certain scale threshold is reached. The conclusion is that each feature (or portion of a feature) was larger than the scale threshold used to obtain the images in which it was still present, but smaller than the scale threshold used in obtaining the image in which it first was absent, as well as subsequent images in for which increasingly larger scale thresholds were used.

5.3 Results of the $\tilde{\alpha}$ Algorithm for noisy images

We next apply the $\tilde{\alpha}$ Algorithm to three noisy images, using four different noise levels. as shown in Figure 8. We consider the 256 x 256 Peppers image, the 256 x 256 Elaine image, and the 140 x 140 Blood Vessels image. Before adding noise, as usual the images are normalized to minimum and maximum intensities of 0 and 1, and the domain is the unit square. In each case exactly the same Gaussian noise (of four different magnitudes) is added to each image. The four levels of noise are created by scaling the noise to have maximum magnitude (both positive and negative) of 0.25, 0.50, 0.75 and 1.00. Because each image has a different signal level, although the same noise is added to each image, the resulting noisy images have different signal-to-noise ratios, as is seen in the second table in Table 2. The $\tilde{\alpha}$ Algorithm is applied to each of the twelve noisy images where in all cases the scale threshold is $1/4n$ (where the image is n x n on the unit square), which

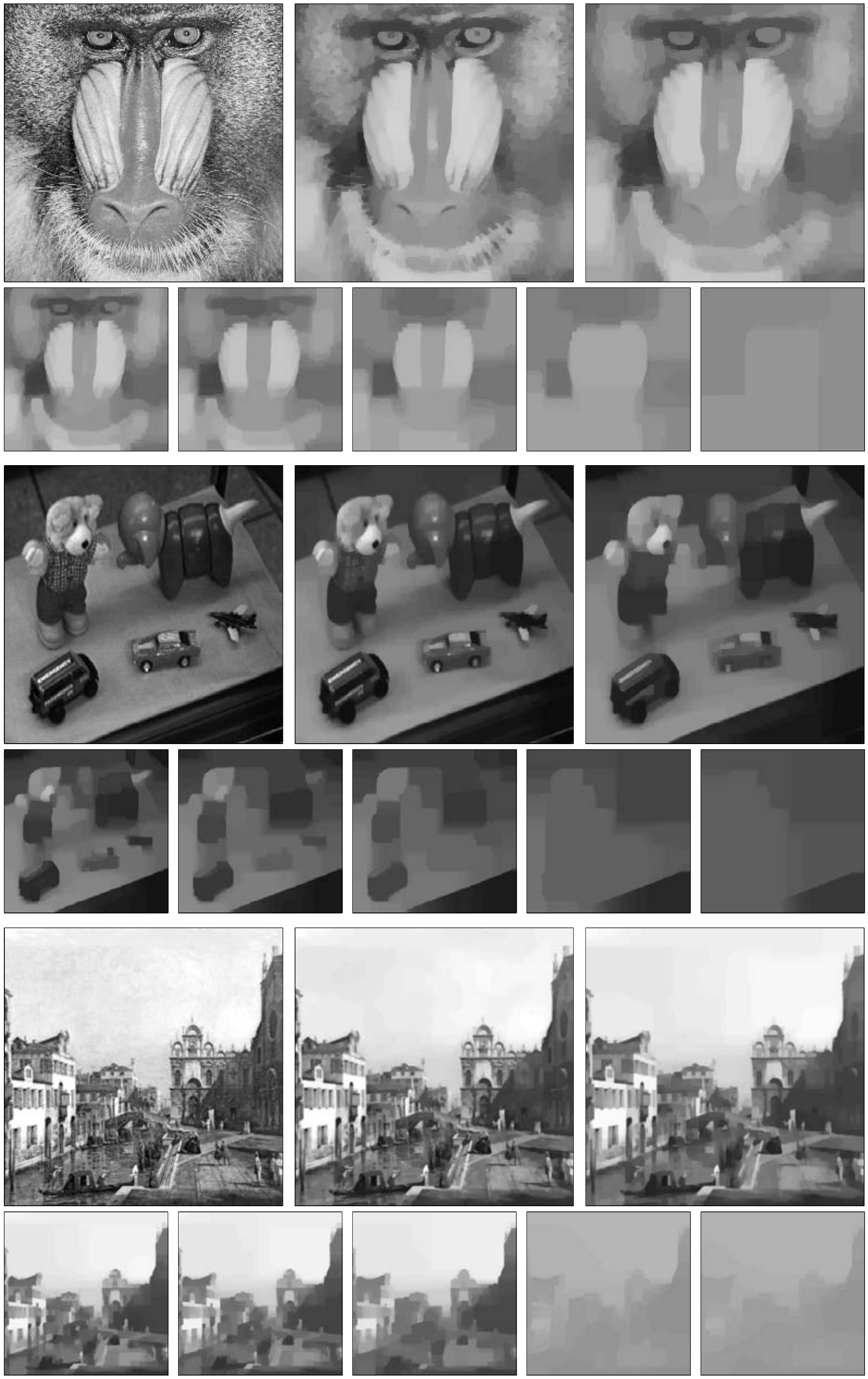


Figure 7: Results of the $\tilde{\alpha}$ Algorithm for scale thresholds of 2^{k-1} x 2^{k-1} pixels for $k = 1$ to 7. The actual scale threshold depends on the size of the image. Values of $\tilde{\alpha}$ found for each scale threshold for each image are given in Table 1 and are plotted in Figure 9. In each set is the original image followed by the seven regularized images.

Figure 8: The $\tilde{\alpha}$ Algorithm applied to three noisy images. The $\tilde{\alpha}$ values found using the $\tilde{\alpha}$ Algorithm, where the scale threshold was a single pixel, are given in the first table in Table 2. Noise levels, before and after regularization, are given in the second table in Table 2. For each pair of images, the top image is the noisy image and the bottom image is the regularized image from solving (2) using the $\tilde{\alpha}$ value found using the $\tilde{\alpha}$ Algorithm.

corresponds to a single pixel. The resulting images are given in Figure 8. The $\tilde{\alpha}$ values found and the old and new SNRs are given in Table 2.

The numerical results given for noisy images are not meant to demonstrate the basic effects of TV regularization on a noisy image, which are of course well known by now. What is novel about these results is that they were obtained without any knowledge of noise level being explicity incorporated into the process for finding the optimal value of α and the corresponding regularized image. The only information used by the $\tilde{\alpha}$ Algorithm was the scale threshold to use: we chose a scale of one pixel in all twelve cases (three images, four noise levels for each). Of course, the amount of noise in the image inherently influences the value of $\tilde{\alpha}$ found by the $\tilde{\alpha}$ Algorithm. As expected, applying the $\tilde{\alpha}$ Algorithm to noisier images results in larger $\tilde{\alpha}$ values, as seen in Table 2.

Obviously it is quite useful to have an approach to denoising that does not depend an accurate measure of noise present in the image, particularly since noise level often is unknown or is, at best, an estimate. As the $\tilde{\alpha}$ Algorithm is not necessarily a denoising algorithm, in this paper we do not further consider it from this point of view. We are currently investigating in more detail the usefulness of the $\tilde{\alpha}$ Algorithm as a denoising scheme, and we will give results in a separate paper.

5.4 $\tilde{\alpha}$ as a function of $scale_{thresh}$ and SNR

We conclude this section of numerical results by examining how $\tilde{\alpha}$ increases with $scale_{thresh}$ for the three noise-free images considered and how $\tilde{\alpha}$ increases with *noise* for the three noisy images considered. These $\tilde{\alpha}$ values were already given in Tables 1 and Table 2. The plots of these data are given in Figure 9.

The first plot in Figure 9 shows the values of $\tilde{\alpha}$ found as a function of $scale_{thresh}$ for the Mandrill, Toy and Canaletto images. Although each of the noise-free images is quite different from the other two, the values of $\tilde{\alpha}$ found for each $scale_{thresh}$ are quite similar. Also, the resolution of the image does not seem to significantly affect the relationship between $\tilde{\alpha}$ and the chosen scale threshold, as illustrated by the similar results of both of the 256 x 256 Mandrill and Toy images as compared to the 512 x 512 Canaletto image.

The other plot in Figure 9 shows the values of $\tilde{\alpha}$ found as a function of noise level. For all three images and for all four noise levels, we found the $\tilde{\alpha}$ corresponding to a scale threshold of $1/4n$, i.e. a single pixel. For all three images, $\tilde{\alpha}$ appears to increase as noise level increases at approximately the same rate. Quite interestingly, the relationship between $\tilde{\alpha}$ and noise level is nearly exactly linear for the given range of noise levels.

Observation The $\tilde{\alpha}$ values for the Blood Vessels image are larger than those for the Peppers and Elaine images because it is 140 x 140 as opposed to the Peppers and Elaine images being 256 x 256. Since the domain for all three images is the unit square, the scale a single pixel in the Peppers image is 140/256 the scale of a single pixel in the Elaine and Blood Vessel images. The ratios of the Peppers and Elaine $\tilde{\alpha}$ values to the Blood Vessels $\tilde{\alpha}$ values is close to 140/256 (of course since they are different images, we would not expect this to be exact).

Both plots of Figure 9 are very interesting, but it is not completely clear how to best interpret or generalize these results. It will certainly be worthwhile to further investigate these issues in future work.

6 Other applications of scale recognition

In this final section (prior to the summary and conclusions), we begin to consider other ways in which to exploit our understanding of TV regularization's natural ability to perceive scale in an image. As already seen above, we can measure the scale throughout the image in order to precisely find the minimum value of α required to remove all features at or below any given scale threshold. We briefly consider two other potential uses for this ability to measure scale. First, we can determine at exactly which locations there is a feature or a portion of a feature of or below any given scale. This leads to some insight on the multiscale effects of TV regularization, which we briefly examine in Section 6.2. Second, in Section 6.3 we use the ability to determine scale at each discrete location throughout the image to examine the rate at which scale is lost as

$scale_{thresh}$	Mandrill	Toys	Canaletto
1/2048	-	-	0.00012
1/1024	0.00052	0.00018	0.00031
1/512	0.00100	0.00066	0.00068
1/256	0.00132	0.00151	0.00099
1/128	0.00224	0.00308	0.00175
1/64	0.00418	0.00457	0.00256
1/32	0.00750	0.01007	0.00346
1/16	0.01793	0.01670	-

Table 1: The $\tilde{\alpha}$ values found for the several scale thresholds used when applying the $\tilde{\alpha}$ Algorithm to the (noise-free) Mandrill, Toys and Canaletto Images in 7. These values are plotted in Figure 9(a).

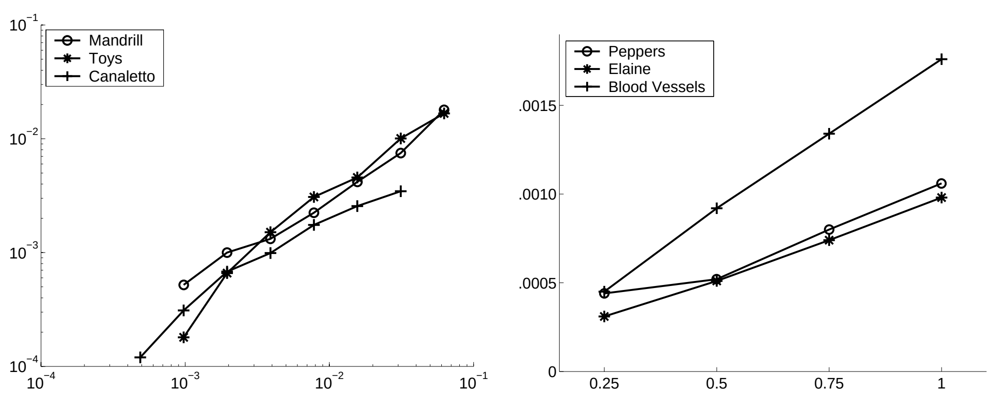
$\tilde{\alpha}$ values

Noise level	Peppers	Elaine	Blood Vessels
0.25	0.00044	0.00031	0.00045
0.50	0.00052	0.00051	0.00092
0.75	0.00080	0.00074	0.00134
1.00	0.00106	0.00098	0.00176

Signal-to-noise ratios
where SNR $= \sigma^2_{signal}/\sigma^2_{noise}$

Noise	Peppers		Elaine		Blood Vessels	
level	Old	New	Old	New	Old	New
0.25	11.11	22.23	11.56	24.54	10.18	31.29
0.50	2.78	17.18	2.89	15.78	2.55	14.36
0.75	1.23	11.29	1.28	11.60	1.13	9.81
1.00	0.69	8.55	0.72	9.20	0.64	7.53

Table 2: Data for the images seen in Figure 8. The first table gives the $\tilde{\alpha}$ values found for the four noise levels that were added to each image. The second table gives the corresponding signal-to-noise ratios for each image and noise level, both before and after regularization using the $\tilde{\alpha}$ value given in the first table. A scale threshold of one pixel was used in all cases. The values in the first table are plotted in Figure 9(b).



(a) $\tilde{\alpha}$ as function of $scale_{thresh}$.   (b) $\tilde{\alpha}$ as function of noise level.

Figure 9: Plots of $\tilde{\alpha}$ values as a function of $scale_{thresh}$ (first plot) and noise level (second plot). The data in the first plot are the $\tilde{\alpha}$ values in Table 1, which were used to obtain the results for the noise-free images in Figure 7. The data in the second plot are the $\tilde{\alpha}$ values in the first table of Table 2, which were used to obtain the results for the noisy images in Figure 8.

α increases for the two images considered in Section 6.2 and comment on how this might generalize to other images.

As mentioned earlier, the analysis in this paper is intended to help (rather than exhaustively) develop a precise understanding of how TV regularization perceives scale in an image and how TV regularization resolves an image into its various scales. Still, two basic behaviors, which we give next as axioms, should hold regardless of the complexity of the image.

6.1 Measuring scale $||u||_{scale}$ and contrast $||u||_{contrast}$

Let $||u||_{scale}$ be a seminorm of u with respect to the scales present in u, and let $||u||_{contrast}$ be a seminorm of u with respect to the contrast present in u. Given

$$u = arg \min_u \tfrac{1}{2}||u - u_0||^2 + \alpha \, TV(u) \tag{21}$$

$$u_i = arg \min_u \tfrac{1}{2}||u - u_0||^2 + \alpha_i \, TV(u) \tag{22}$$

$$\hat{\alpha} = ||u_0 - \bar{u}_0||_G \tag{23}$$

any measure $||u||_{scale}$ of scale and $||u||_{contrast}$ of contrast should satisfy the following axioms.

Axiom 1 (Increasing scale) *Where* $M = \sup_\alpha ||u||_{scale}$ *and given (21) - (23), we have:*

- *If* $\alpha_1 < \alpha_2$*, then* $||u_1||_{scale} \leq ||u_2||_{scale}$*.*
- *As* $\alpha \nearrow \hat{\alpha}$*,* $||u||_{scale} \nearrow M$*.*

In short, scale is non-decreasing and asymptotically increasing in α*.*

Axiom 2 (Decreasing contrast) *Given (21) - (23), we have:*

- *If* $\alpha_1 < \alpha_2$*, then* $||u_1||_{contrast} > ||u_2||_{contrast}$*.*
- *As* $\alpha \nearrow \hat{\alpha}$*,* $||u||_{contrast} \searrow 0$*, and for* $\alpha \geq \hat{\alpha}$*,* $||u||_{contrast} = 0$*.*

In short, contrast is decreasing in α*.*

Remark Axiom 1 basically says that the evolution of scale, both for the image as a whole and at any particular location in the image, is nonreversible. That is, there is sort of a scale entropy: as α increases, the scale, as measured in the image as a whole or at each location, increases asymptotically in finite time (i.e. for a finite value of α) to the limiting maximum scale where there is no variation in scale. Of course, we know that for $\alpha \geq \hat{\alpha}$ (whos value depends only on u_0 and the size of the image domain), the solution to (2) is simply a constant-valued image with value equal to the mean of the original image u_0. Axiom 2 describes a similar notion for contrast.

Definition 2 *Let* $||u||^{\vec{x}}_{scale}$ *be the smallest scale in u still present at* $\vec{x}$*.*

Remark For this definition, Axiom 1 becomes a property.

Example We saw in Figure 3 that the region $[9/27, 10/27]$ (the first of the 9 extrema) is part of features of three different scales, depending on the value of α used to regularized the image. More precisely, for $x \in [9/27, 10/27]$ (or $x \in (9/27, 10/27)$—the end points of the interval, having measure 0, are not important), we have

$$||u||^{x}_{scale} = \begin{cases} 1/54 & \text{if} \quad 0 \leq \alpha < \alpha_{9\to 3} \\ 1/18 & \text{if} \quad \alpha_{9\to 3} \leq \alpha < \alpha_{3\to 1} \\ 1/6 & \text{if} \quad \alpha_{3\to 1} \leq \alpha < \alpha_{1\to 0} \end{cases}$$

For $\alpha \geq \alpha_{1\to 0}$ there are no longer features present in the regularized image, which would be constant-valued.

Remark In addition to the above definition of $||u||^{\vec{x}}_{scale}$, there would be several other definitions of $||u||_{scale}$ that are natural measures of contrast in an image. Similarly, there would be a variety of natural definitions of $||u||_{contrast}$. Any global (i.e. over the entire image) or local (i.e. at a specific location in the image) definition of $||u||_{scale}$ or $||u||_{contrast}$ should satisfy the above axioms.

Since we are measuring scale using (6), Definition 2 naturally applies to our work. We will observe the scale entropy described in Axiom 1 in the following section.

6.2 Multiscale and scalespace effects of TV regularization

The multiscale and scalespace-generating effects of TV regularization are well known and are the subject of ongoing investigation. See, for example, [17], [21] and [26]. Of course, a more accurate and complete understanding of the multiscale and scalespace-generating nature of TV regularization is really only possible if there exists a precise and complete notion of scale as perceived by TV regularization. Therefore, we expect that the theory and discussion presented in the previous sections will lead to a better understanding of the multiscale and scalespace-generating effects of TV regularization. As mentioned earlier, as this is a fairly complex issue, we do not attempt to treat it in detail in this paper. Rather, we give two examples that lend some insight into the inherent ability of TV regularization to recognize scale, insight that we expect to lead to further discussion and development of theory.

6.2.1 Scalespace of Mandrill image

We consider in more detail the Mandrill image shown earlier in Figures 6 and 7. The image is 256 x 256, and as usual the domain is the unit square and we have normalized the image so that the minimum and maximum intensities are 0 and 1.

Earlier we found that $\tilde{\alpha} = 0.00052$ is the minimum value of α necessary to remove all features at or below a scale threshold corresponding to a single pixel. We now examine the results when solving (2) using a range of values between 0 and $\tilde{\alpha}, 0.1\,\tilde{\alpha}, 0.2\,\tilde{\alpha}, ..., \tilde{\alpha}$, to see in more detail the effects of the regularization. The resulting images are given in Figure 10. There are eleven sets of images, the first corresponding to the original image, and the other ten corresponding to the results of solving (2) using these ten values of α.

For each set (organized by columns), the top image is the image itself. The second image (second row of the set) shows the locations throughout the image at which there are features at or below the scale threshold of $1\,/\,4n$ (where $n = 256$), the scale of a single pixel. Similarly, the third and fourth rows of images show the locations in the image at which there are features at or below the scale thresholds of $1\,/\,3n$ and $1\,/\,2n$, the scales corresponding to 1 x 2 pixel and 2 x 2 pixel features, respectively. The remaining percentage of features at or below each of the given scale thresholds for each value of α is given in the first table in Table 3.

Observation In examining the images in Figure 10, it is apparent that most of the feature removal is relatively immediate, i.e. for the smaller values of α. For example, the second row of images shows the location of features whose scale corresponds to that of a single pixel. Although a value of $\tilde{\alpha} = 0.00052$ is needed to completely remove *all* features of this size from the image, even for $\alpha = 0.30\,\tilde{\alpha}$ or $\alpha = 0.40\,\tilde{\alpha}$, the image is almost entirely devoid of these one pixel features. We demonstrate this in more detail for a portion of this image in Figure 11. Notice, in particular, that the one feature that is still present until the end is the center of pupil of the Mandrill's left eye (the right eye, from our perspective). So if the goal is to remove all single-pixel features, perhaps a smaller value of α should be used, even if there are a few single-pixel features still remaining in order to better preserve the (wanted) larger features. This decision will depend on the image and the reason for applying regularization. It is not completely clear how to best evaluate the results in Figures 10 and 11; still, they are enlightening and shed some new light on how TV regularization has a multiscale effect on images, dependent on the value of α used in solving (2). It is clear that further investigation of TV regularization multiscale effects is warranted.

Remark In the images shown in Figure 10 and especially in the images shown in Figure 11, it is clear that once scale at any given location is recognized as being at or above a certain thresold, it will never

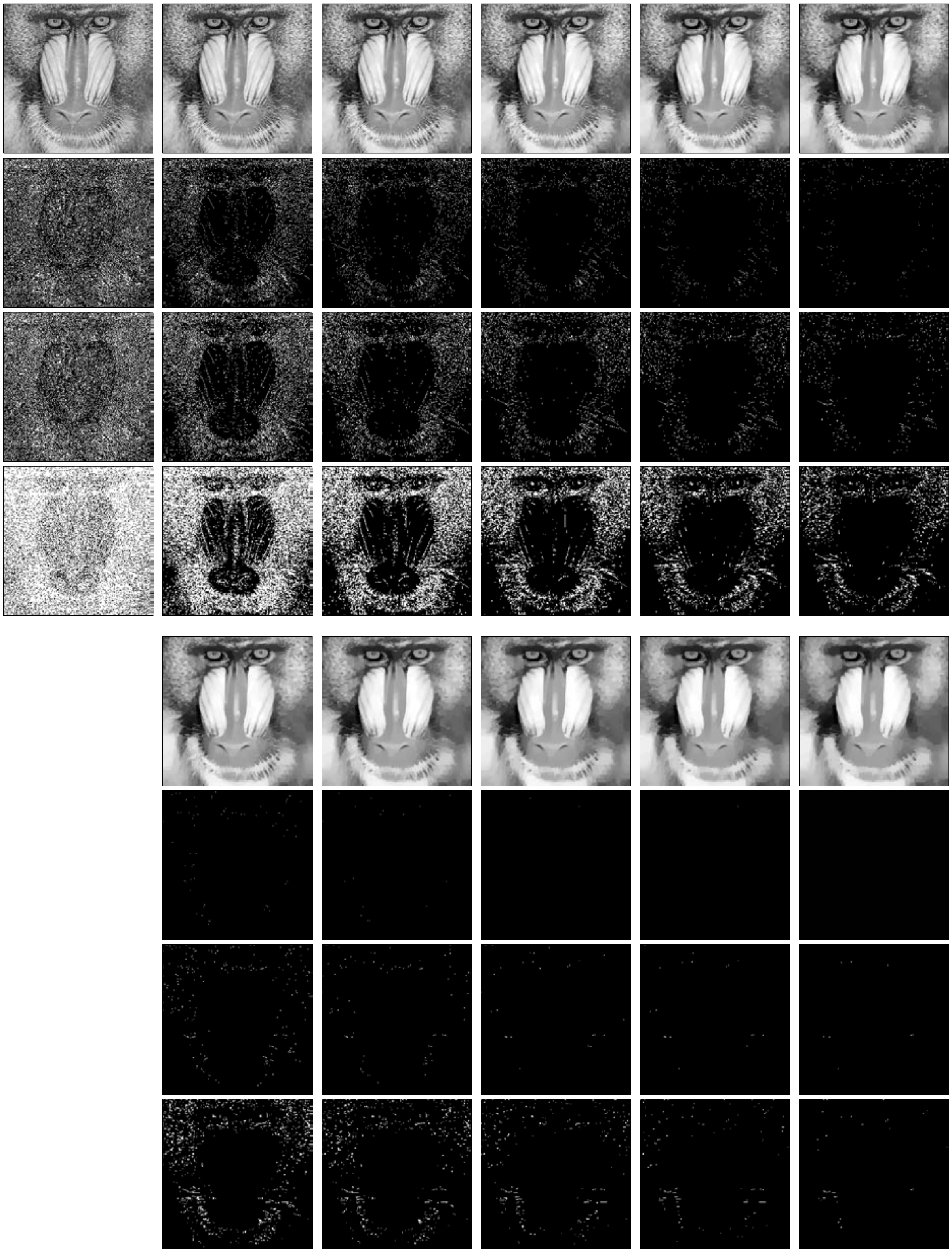


Figure 10: Results of applying TV regularization (2) to the Mandrill image. The eleven sets of images correspond to the original image plus the ten images resulting from solving (2) using $\alpha = 0.1\,\tilde{\alpha}$, $0.2\,\tilde{\alpha}$, $0.3\,\tilde{\alpha}$, $0.4\,\tilde{\alpha}$ and $0.5\,\tilde{\alpha}$ (top set of images) and $\alpha = 0.6\,\tilde{\alpha}$, $0.7\,\tilde{\alpha}$, $0.8\,\tilde{\alpha}$, $0.9\,\tilde{\alpha}$ and $1.0\,\tilde{\alpha}$ (bottom set). For each set of images, the top image is the image itself, while the second through fourth images show the locations of all (portions of) features with scale at or below $1\,/\,4n$ (1 x 1 pixel), $1\,/\,3n$ (1 x 2) and $1\,/\,2n$ (2 x 2), respectively.

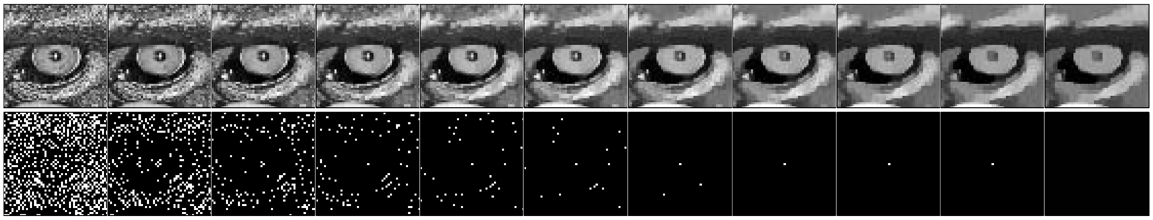


Figure 11: Results of applying TV Regularization using α. The first row contains the original image, plus results of solving (2) using $\alpha = 0.1\,\tilde{\alpha}, 0.2\,\tilde{\alpha}, ..., \tilde{\alpha}$. The second row contains corresponding images showing the locations of still-remaining features at or below the one-pixel scale.

drop below that threshold, and in fact, as described in Axiom 1, the scale at every location throughout the image will increase asymptotically to a maximum scale as α increases. The white "dots" in Figures 10 and Figure 11 are the locations at which are there features at or below a given scale. Notice that you see only the disappearance of the dots, but no reappearance of dots or appearance of new dots anywhere. We can formalize this phenomenum with the following proposition.

Proposition 3 *Given image u_0, for a given scale and α, define*

$$S^{\alpha}_{scale} = \left\{ \vec{x} : scale_u(\vec{x}) \leq scale, u = arg \min_u \tfrac{1}{2}||u - u_0||^2 + \alpha\, TV(u) \right\}$$

then we have:

- *For any α, if $scale_1 < scale_2$ then $S^{\alpha}_{scale_1} \subseteq S^{\alpha}_{scale_2}$.*
- *For any scale, if $\alpha_1 > \alpha_2$ then $S^{\alpha_1}_{scale} \subseteq S^{\alpha_2}_{scale}$.*

Remark This proposition is directly related to Axiom 1 given in Section 6.1 and to the notion of scale entropy. Although we do not prove the proposition, the principles conveyed by both statements are apparent in Figure 10. The second statement, in which scale is fixed, is also illustrated quite nicely in Figure 11.

6.3 Rate of loss of features

We last briefly examine the decay (rate of loss) of features of any given scale in an image. In the previous section we saw that we can recognize scale throughout the image. It is illuminating to look at the rate of decay of the remaining scale for increasing values of α. Table 3 gives us the percentage of all features at or below a given scale remaining for each value of $\tilde{\alpha}$, as illustrated in Figure 10. These data are plotted in Figure 12(a).

As a second example we find the same information about remaining percentages at the same three scale levels for the Canaletto image. In this second case, since most of the features for each of the three scales in the Mandrill image seemed to be removed rather quickly, we now use more values of α, particularly smaller values, in order to observe more gradually the decrease in percentages. These data are listed in the second table in Table 3 and are plotted in Figure 12(b).

Remark For both images, in Figure 12, we first plot the standard (linear) plot of each scale percentage, and then we give the log (in the y axis, the percentage of features at or below a certain scale) plot of the same data. From these plots, we see that the rate of loss or decay of the features at or below the three given scales seems nearly exponential for both images. Of course we could easily contrive an image for which scale decay is not exponential. Still, it may be that for a variety of natural images, scale decay would be exponential. That is, most of the features at or below a given scale disappear rapidly, while there are a few features that still remain for a while until α is too large. This was especially evident in Figures 10 and 11. This decay of scale and our ability to measure it using TV regularization certainly merit further investigation.

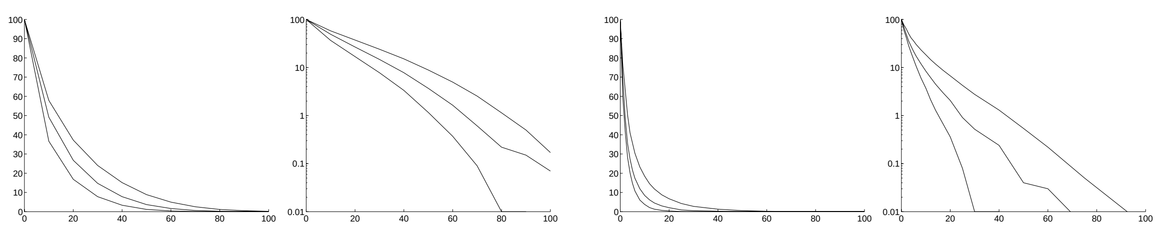
Mandrill Image

α, as %	% of scale remaining		
of $\tilde{\alpha}$	$1 / 4n$	$1 / 3n$	$1 / 2n$
0	100.00	100.00	100.00
10	36.77	49.15	57.88
20	16.85	26.73	37.25
30	7.79	14.74	24.03
40	3.34	7.79	15.11
50	1.16	3.68	8.87
60	0.37	1.64	4.96
70	0.09	0.61	2.54
80	0.01	0.22	1.14
90	0.01	0.15	0.50
100	0.00	0.07	0.17

Canaletto Image

α, as %	% of scale remaining		
of $\tilde{\alpha}$	$1 / 4n$	$1 / 3n$	$1 / 2n$
0	100.00	100.00	100.00
1	60.39	70.30	77.39
2	41.95	49.31	63.76
3	28.28	35.50	50.32
4	20.22	27.42	41.07
5	14.77	21.67	35.75
6	10.79	17.38	30.41
8	6.08	11.97	23.32
10	3.74	8.44	18.46
12	2.11	6.14	14.54
14	1.29	4.50	11.80
17	0.68	3.00	8.78
20	0.36	2.06	6.71
25	0.08	0.91	4.25
30	0.01	0.52	2.76
40	0.01	0.24	1.30
50	0.00	0.04	0.54
60	0.00	0.03	0.22
75	0.00	0.00	0.05
100	0.00	0.00	0.00

Table 3: The percentage of features at or below three specified scales remaining after applying TV regularization (2) to the Mandrill image (shown in Figure 10) and the Canaletto image for various values of α. The three scales considered are $1 / 4n$, $1 / 3n$ and $1 / 2n$, which correspond to scales of 1 x 1, 2 x 1 and 2 x 2 pixel features, respectively. Each column shows the percentage of the orginal pixel locations recognized as being at or below the specific scale for the given value of α. We found $\tilde{\alpha}$ using the $\tilde{\alpha}$ Algorithm. For the Mandrill image, $\tilde{\alpha}$ corresponded to removing all features at or below scale $1 / 4n$: notice the $1 / 4n$ column of Mandrill results. For the Canaletto image, $\tilde{\alpha}$ corresponds to removing all features at or below scale $1 / 2n$: notice this in the $1 / 2n$ column of Canaletto results.



(a) Decay of features in Mandrill image.

(b) Decay of features in Canaletto image.

Figure 12: A plot of the remaining percentages of scales listed in Table 3. For each pair of images, the first plot is the linear plot of the data, and the second plot is the log plot of the data. The nearly linear behavior seen in the log plots illustrates the nearly exponential decay of the image features of the three scales considered. For each image, the three curves, from top to bottom, show the percentage of features at or below scale thresholds of $1 / 4n$, $1 / 3n$ and $1 / 2n$, respectively.

7 Summary and Conclusions

TV regularization naturally recognizes scale in an image. This gives us great insight into how TV regularizations works, and it leads to a number of ways in which this ability to recognize scale can be exploited. As shown, we can automatically and precisely determine how much regularization is needed (i.e. what value of α to choose) to remove all features at or below a given scale threshold from an image. There is a nice connection between Meyer's G Norm and both our notion of scale and our $\tilde{\alpha}$ Algorithm. This connection leads to a more intuitve explanation of the G norm and how it relates to scale in an image. The ability to recognize scale leads to a better understanding of already known TV-based ideas and schemes, including scalespace, and it leads to a number of new and potentially very useful tasks for manipulating and understanding images, including measuring the decay of features of various scales in an image. Using this ability to measure scale, for the examples we considered, we have seen that most features at a given scale tend to disappear quickly, while a relatively small fraction persists longer. Some of the ideas investigated in this paper are complete, and some of the work was intended to show how more possible avenues of investigation have been opened due to this ability to recognize scale. Finally, although this work is done for images in $\mathbf{R}^2$, the theory developed can be extended to any function in any dimension. Other work that naturally stems from the work done in this paper includes a spatially adaptive $\tilde{\alpha}$ Algorithm and more efficient approaches to finding $\tilde{\alpha}$, such as multigrid and domain decomposition approaches to the $\tilde{\alpha}$ Algorithm, which we are currently developing.

References

[1] R. Acar and C. Vogel. Analysis of total variation penalty methods. *Inverse Problems*, 10:1217–1229, 1994.

[2] H. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Trans. Image Processing*, 2005. to appear.

[3] F. Andreu, V. Caselles, J. Diaz, and J. Mazon. Some qualitative properties for the total variation flow. *J. Functional Analysis*, 188:516–547, 2002.

[4] G. Aubert and J-F. Aujol. Modeling very oscillating signals. Application to image processing. *Applied Mathematics and Optimization*, 51(2), March/April 2005.

[5] J-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *J. Mathematical Imaging and Vision*, 22(1), January 2005.

[6] J-F. Aujol and A. Chambolle. Dual norms and image decomposition models. *International J. Computer Vision*, 63(1), June 2005.

[7] G. Bellettini, V. Caselles, and M. Novaga. The total variation flow in $\mathbf{R}^n$. *J. Differential Equations*, 184:475–525, 2002.

[8] P. Blomgren and T. Chan. Color TV: total variation methods for restoration of vector-valued images. *IEEE Trans. Image Processing*, 7:304–309, 1998.

[9] A. Chambolle. An algorithm for total variation minimization and applications. *J. Mathematical Imaging and Vision*, 20:89–97, January 2004.

[10] A. Chambolle and P-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76:167–188, 1997.

[11] T. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Scientic Computing*, 20:1964–1977, 1999.

[12] T. Chan and J. Shen. Mathematical models for local non-texture inpaintings. *SIAM J. Applied Mathematics*, 62:1019–1043, 2001.

[13] T. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Processing*, 7:370–375, 1998.

[14] Q. Chang and I-L. Chern. Acceleration methods for total variation-based image denoising. *SIAM J. Applied Mathematics*, 25:982–994, 2003.

[15] D. Dobson and F. Santosa. Recovery of blocky images from noisy and blurred data. *SIAM Journal on Applied Mathematics*, 56:1181–1198, 1996.

[16] D. Dobson and C.R. Vogel. Convergence of an iterative method for total variation denoising. *SIAM Journal on Numerical Analysis*, 34:1779–1971, 1997.

[17] G. Gilboa, N. Sochen, and Y. Y. Zeevi. PDE-based denoising of complex scenes using a spatially-varying fidelity term. In *Proc. ICIP 2003, Barcelona, Spain*, volume 1, pages 865–868, 2003.

[18] Y. Gousseau and J-M. Morel. Are natural images of bounded variation? *SIAM J. Mathematical Analysis*, 33:634–648, 2001.

[19] Yves Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations. Providence, RI, 2001. The Fifteenth Dean Jacquelines B. Lewis Memorial Lectures.

[20] S. Osher, A. Sole, and L. Vese. Image decomposition and restoration using total variation minimization and the h^{-1} norm. *Multiscale Model. Simul.*, 1:349–370, 2003.

[21] E. Radmoser, O. Scherzer, and J. Weickert. Scale-space properties of regularization methods. In *Scale-space theories in computer vision, Lecture Notes in Computer Science, Vol. 1682, Springer, Berlin, M. Nielsen, P. Johansen, O.F. Olsen, J. Weickert (Eds.)*, pages 211–222, 1999.

[22] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[23] G. Strang. L^1 and L^∞ approximation of vector fields in the plane, 1982. Lecture Notes in Num. Appl. Anal, Volume 5.

[24] D. Strong. *Adaptive Total Variation Minimizing Image Restoration*. PhD thesis, UCLA Math Department, August 1997. CAM Report 97-38.

[25] D. Strong and T. Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, 19(6):165–187, 2003.

[26] E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (BV, L^2) decompositions. *SIAM journal on Multiscale Modeling and Simulation*, 2(4):554–579, 2003.

[27] A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. Winston and Sons, Washington D.C, 1977.

[28] L. Vese and S. Osher. The level set method links active contours, mumford-shah segmentation, and total variation restoration, February 2002. UCLA CAM Report 02-05, available at http://www.math.ucla.edu/applied/cam/index.html.

[29] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17:227–238, 1996.