Segmentation of Pulmonary Nodule Images Using Total Variation Minimization*

Thomas F. Coleman† Yuying Li‡ Adrian Mariano§

September 8, 1998

Abstract

Total variation minimization has edge preserving and enhancing properties which make it suitable for image segmentation. We present Image Simplification, a new formulation and algorithm for image segmentation. We illustrate the edge enhancing properties of total variation minimization in a discrete setting by giving exact solutions to the problem for piecewise constant functions in the presence of noise. In this case, edges can be exactly recovered if the noise is sufficiently small. After optimization, segmentation is completed using edge detection. We find that our image segmentation approach yields good results when applied to the segmentation of pulmonary nodules.

1 Introduction

Image segmentation is the partitioning of an image into regions so that each region corresponds to one object in the image. Due to the importance of the segmentation problem in computer vision, numerous strategies have been proposed. See the surveys [7, 9, 14] for some examples. We propose Image Simplification, a new total variation minimization approach for image segmentation.

An *image* is a matrix whose entries are called *pixels*. One segmentation approach is to categorize pixels according to their numerical values. A histogram is used to find clusters of pixels with similar values, and a final segmentation is obtained by finding connected sets of pixels in the same clusters. Methods which divide up pixels based on their numerical values are called *thresholding* methods and were surveyed in [18]. Because spatial information is poorly exploited, these approaches work best when the objects have sharp contrast and appear on a

*This research was supported by the NIH Parallel Processing Resource for Biomedical Scientists, SSS-4(E) 1 P41 RR04293-01A3.

†Computer Science Department and Cornell Theory Center, Cornell University, Ithaca, New York, 14853.

‡Computer Science Department, Cornell University, Ithaca, New York, 14853.

§Center for Applied Mathematics, Cornell University, Ithaca, New York, 14853.

non-varying background [9]. These methods often require a priori knowledge of the number of objects in the image [7].

Region growing is a spatial approach in which pixels are regarded as the nodes of a graph. They are connected to their neighbors according to some criterion such as the difference in value [3, 1]. A segmentation is obtained by identifying the connected components of the graph. However, it only takes one extra edge in the graph to erroneously join two regions; therefore, this method is particularly susceptible to noise [9]. An image is corrupted by *additive noise* when the pixels are perturbed from their correct values. Noise is inevitably introduced in the process of acquiring images, but sometimes the noise is very small and can be ignored. Usually noise is modeled as a random image whose pixels are independent identically distributed Gaussian random variables.

A more sophisticated approach is to associate with each region of the image some value such as the average pixel value over the region. Adjacent regions are merged if the values are close enough to each other. The algorithm can be started with every pixel in a region by itself. Often several passes are required to obtain a segmentation. Algorithms of this type were surveyed in [19]. This process is inherently sequential and does not use global information [7].

Edge detection is another technique that has been applied to image segmentation. Objects in the image are thought to have boundaries where pixel values change rapidly. Many edge detection approaches are based on discrete approximations to the gradient [7]. Boundary points are identified and assembled into paths using methods such as heuristic search [11] or line fitting [6]. These paths divide the image into regions. A major difficulty with this approach is that edge detection methods often produce edges with gaps, and they tend to produce false edges, particularly in the presence of noise.

Image segmentation has been formulated as an optimization problem, e.g. Mumford and Shah [13]. Their strategy is to seek an approximation to the given image along with a set of regions. This approximation should be flat on each region and the length of the boundaries of all the regions should be small. This set of conditions is obtained by formulating an optimization problem with a two part objective function. One part imposes a penalty for deviation from the original function; the other part imposes a penalty proportional to the length of the boundary of the regions. This optimization problem has both discrete variables that specify where region boundaries lie, and continuous variables that give the value of the approximation. A major disadvantage of this approach is that no algorithm is known that provably converges to the solution. Blake and Zisserman propose a heuristic in [2]. In [12], the idea of solving image segmentation with this type of optimization formulation is considered in great detail.

Recent work has indicated the value of total variation, a technique using the ℓ_1 norm, in image processing. Total variation was introduced by Rudin and Osher [15] for image restoration and noise removal. In [10], Li and Santosa present a computationally simpler total variation formulation for image restoration. Total variation measures are useful in image processing because they are able to produce sharp edges in the output images. This ability has applications to image segmentation where identification of edges is very important.

The ability of total variation methods to preserve edges can be understood through analysis of special cases. Strong and Chan found exact solutions to special cases of continuous total variation problems [16]. These exact solutions show that in certain special cases in the presence of noise, edges are perfectly restored by the algorithm. We present the exact solution in a discrete setting using our formulation.

We apply the formulation of Li and Santosa to image segmentation by using a noise removal method. The typical goal of noise removal is to restore an image to a noise-free state. We wish to go further, removing from the image small scale details which impede image segmentation. Unlike many of the traditional image segmentation approaches, our method is a global one which operates on the entire image at once. It can be used effectively on very noisy images. Unlike other variational formulations, our proposed formulation is a convex programming problem and a solution can be computed.

We apply our segmentation approach to the problem of identifying pulmonary nodules in X-ray computed tomography (CT) scans (see for example [17, 8]). Pulmonary nodules are abnormal growths in the lungs which often, but not always, lead to lung cancer. Early identification of malignant nodules is crucial for effective treatment of lung cancer. Automation is sought because CT scans produce an enormous amount of information, and searching through the information by hand is time consuming. Precise identification of nodule boundaries is important for diagnostic purposes. Nodule boundary information can be used to estimate nodule volume, making determination of the nodule growth rate possible. Features of the boundary of pulmonary nodules are important for distinguishing between malignant and benign nodules. The main difficulty of the pulmonary nodule segmentation problem involves the separation of nodules and blood vessels, which can have very similar intensity values near boundaries.

In Section 2 of this paper we describe Image Simplification, an ℓ_1 optimization method using the total variation. We prove that for certain piecewise constant images that have been corrupted by a sufficiently small noise image, the edges in the original image will be recovered exactly. Section 3 of the paper explains a computational method for solving the Image Simplification problem. Section 4 describes how we extract the boundaries of pulmonary nodules from the image after Image Simplification.

2 Total Variation for Image Simplification

Image Simplification comprises an optimization problem and an algorithm for solving that problem. In this section we consider the formulation of the Image Simplification optimization problem. An image in a continuous setting is a real valued function. Let $\Omega \subset R^n$, let μ be Lebesgue measure, and let f be a differentiable image defined on Ω. The definition used by Rudin and Osher in [15] is

$$\mathrm{TV}(f) = \int_\Omega \|\nabla f\|_2 \, d\mu. \tag{1}$$

We define the total variation of an image by

$$\mathrm{TV}(f) = \int_\Omega \|\nabla f\|_1 \, d\mu. \tag{2}$$

We have chosen this definition because it leads to simpler computations in a discrete setting . A one-dimensional image is called a *signal*. The total variation of a signal x in the discrete setting is given by

$$\mathrm{TV}(x) = \sum_i |x_i - x_{i-1}| \, .$$

In two dimensions, the total variation involves terms for the horizontal, vertical, and possibly two diagonal directions. In general, the discrete total variation can be formulated in n dimensions in two different ways: differences can be taken in orthogonal directions, or differences can be taken in both orthogonal and diagonal directions. In n-dimensional space there are n orthogonal directions to consider, and there are $(3^n - 1)/2$ diagonal directions. The total variation can be represented generally in a discrete setting as

$$\mathrm{TV}(x) = \|Bx\|_1$$

where B is a matrix which takes differences between adjacent pixels in the image, possibly with spatially dependent weights. For one-dimensional problems, we define B_{1D} so that an increase in the signal x from left to right results in a positive entry in $B_{\mathrm{1D}}x$. For example if we deal with length 6 signals, then

$$B_{\mathrm{1D}} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

For two-dimensional problems, we define B_{2D} so that an increase in an image x from left to right or from top to bottom results in a positive entry in $B_{\mathrm{2D}}x$. The B_{2D} matrix will have a block for horizontal differences and one for vertical differences. If diagonal differencing is desired, B_{2D} will have four blocks.

Unlike more traditional least squares measures, the total variation assigns the same value to edges as it does to other monotonic functions (see Figure 1). This edge tolerance is crucial for image segmentation because it enables total variation methods to preserve or enhance edges; least squares methods usually smooth out edges.

For computational purposes, we represent $m \times n$ images as vectors in R^{mn}. Let $x_0 \in R^{mn}$ be the original image. We want to find a new image x which is

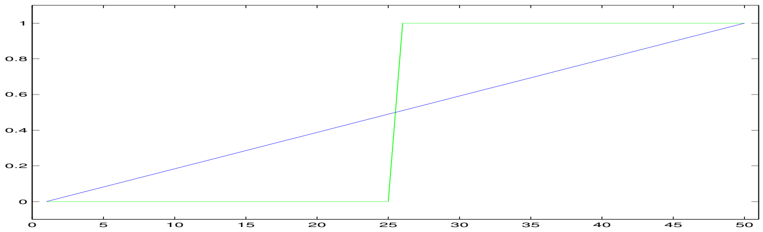


Figure 1: These two signals have the same total variation. Least squares methods will strongly favor the ramp signal because it has a much smaller ℓ_2 norm than the step signal.

close to x_0 but which has the smallest total variation. In order to do this, we use the Image Simplification formulation

$$\min_x \|Bx\|_1 \quad \text{s.t.} \quad \|x - x_0\|_2 \leq \sigma \tag{3}$$

where σ is a parameter which specifies how far x will be from the original image. We select σ by choosing it to be some fraction of the adjusted norm of the original image, $\|x_0 - \text{avg}(x_0)\|_2$ where $\text{avg}(x_0)$ is the average value of x_0.

The ability of the Image Simplification formulation (3) to retain edges relies on the interaction between the objective function and the constraint. We start with a simple piecewise constant example and follow it with some theorems proving that Image Simplification preserves edge location in piecewise constant data. Consider the situation where the original data is a step function defined on $2n$ points by

$$x_0(i) = \begin{cases} 0 & \text{if } 0 \leq i \leq n-1 \\ 1 & \text{if } n \leq i \leq 2n-1. \end{cases}$$

As noted above, the two signals shown in Figure 1 have the same total variation, leading to the concern that Image Simplification (3) could produce a ramp function like $r(i) = i/(2n-1)$ which obliterates the sharp edge in the data. The constraint prevents this ramp function from occuring. Even if σ is chosen large enough to admit the ramp function into the feasible region, Image Simplification will not produce the ramp function because the function

$$x_\delta(i) = \begin{cases} \delta & \text{if } 0 \leq i \leq n-1 \\ 1-\delta & \text{if } n \leq i \leq 2n-1 \end{cases}$$

has a smaller total variation than r, but $\|r - x_0\|_2 = \|x_\delta - x_0\|_2$ for an appropriate δ. In fact, x_δ is the unique optimal solution to the Image Simplification problem when $\delta = \sigma/\sqrt{2n}$. In this simple example, Image Simplification preserves edges. In Theorem 1 we prove that in the noise free case Image Simplification will preserve the edges in any piecewise constant signal. By studying some simple cases, we hope to gain some insight into the general behavior of

Image Simplification. Strong and Chan in [16] proved similar results for continuous functions using (1). We prove our results in a discrete setting with the total variation given by a discretized form of (2).

In order to state the theorems we need some definitions. Different constant sections of a piecewise constant signal will be shifted in different directions by Image Simplification depending on values taken on the surrounding sections. Sections of the signal that are above neighboring sections will be lowered. Sections which are below neighboring sections will be raised. Sections which have one neighbor above and one below will be unchanged. In order to express this formally, we make the following definition. Let x be a piecewise constant image whose domain is partitioned into N regions $R_1, \ldots, R_N$ such that R_i is adjacent only to R_{i+1} and R_{i-1}. Assume that x is constant on R_i where it takes the value $x(R_i)$. If $x(R_i) > x(R_{i-1})$ and $x(R_i) > x(R_{i+1})$, then R_i is a *maximal region.* If $x(R_i) < x(R_{i-1})$ and $x(R_i) < x(R_{i+1})$, then R_i is a *minimal region.* If R_i is a maximal region or a minimal region then it is also an *extreme region.* Define the index set for the extremal regions by

$$\mathcal{E} = \{i \mid R_i \text{ is an extreme region}\}. \tag{4}$$

We also establish a proportionality result for the shifts that Image Simplification produces. In order to express this proportionality result, we need some notation:

$$\begin{aligned} \alpha_i &= \text{area of } R_i \\ \beta_i &= \text{size of the boundary of } R_i. \end{aligned} \tag{5}$$

For piecewise constant signals, α_i is the length of a region, $\beta_i = 2$ for interior regions and $\beta_i = 1$ for the two edge regions. We show in Theorem 1 that the size of the shift produced by Image Simplification is proportional to the size of the boundary and inversely proportional to the size of the area. Intuitively this is because the objective function favors shrinking the size of the jumps at the boundary, and the constraint keeps large regions from shifting up or down too far.

Theorem 1 (Piecewise Constant Signals) *Suppose x_0 is a piecewise constant signal defined on segments R_i, $i = 1 \ldots N$ (where R_i is adjacent only to R_{i+1} and R_{i-1}) and σ is a sufficiently small positive constant. Let*

$$\delta_i = \begin{cases} \frac{\beta_i}{2|\lambda_1|\alpha_i} & \textit{if } R_i \textit{ is a minimal region} \\ \frac{-\beta_i}{2|\lambda_1|\alpha_i} & \textit{if } R_i \textit{ is a maximal region} \\ 0 & \textit{otherwise} \end{cases}$$

where α_i and β_i are defined by (5). The value of $|\lambda_1|$ is specified by

$$4\lambda_1^2 = \frac{\sum_{i \in \mathcal{E}} \frac{\beta_i^2}{\alpha_i}}{\sigma^2},$$

with $\mathcal{E}$ given by (4). Let $x_* = x_0 + \delta_i$ on R_i. Then x_* is the unique optimal solution to the Image Simplification problem (3) where $B = B_{1D}$.

The proof can be found in the appendix. A similar result holds in two dimensions: symmetric images of nested squares are preserved by Image Simplification. An image, x, is an *image of nested squares* if it can be obtained as the result of the following construction. Let $\{S_i\}$, $i = 1 \ldots N$, be a family of squares on a grid such that $S_i \subset S_{i+1}$ and all of the squares have the same center point. Set $S_0 = \emptyset$ and define $R_i = S_i \setminus S_{i-1}$ for $i = 1 \ldots N$. Let x be the vector representation of an image defined on the grid and constant on each R_i.

Theorem 2 (Symmetric Nested Square Images) *Suppose x_0 is an image of nested squares, with R_i, $i = 1 \ldots N$, the regions on which it is constant, and σ is a sufficiently small positive constant. Let*

$$\delta_i = \begin{cases} \frac{\beta_i}{2|\lambda_1|\alpha_i} & \textit{if } R_i \textit{ is a minimal region} \\ \frac{-\beta_i}{2|\lambda_1|\alpha_i} & \textit{if } R_i \textit{ is a maximal region} \\ 0 & \textit{otherwise} \end{cases}$$

where α_i and β_i are defined by (5). The value of $|\lambda_1|$ is specified by

$$4\lambda_1^2 = \frac{\sum_{i \in \mathcal{E}} \frac{\beta_i^2}{\alpha_i}}{\sigma^2},$$

with $\mathcal{E}$ given by (4). Let $x_* = x_0 + \delta_i$ on R_i. Then x_* is the unique optimal solution to the Image Simplification problem (3) where $B = B_{2D}$.

Real images contain noise. In the presence of noise, we can establish similar results. We do this by using a very general theorem concerning solutions to perturbed problems which have the same form as the Image Simplification problem. Let $\operatorname{sgn}(x)$ denote sign of its argument with the convention that $\operatorname{sgn}(0) = 0$.

Theorem 3 (Noise Perturbed Solutions) *Suppose x_* is the optimal solution to*

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - x_0\|_2 \leq \sigma$$

where B is a matrix and $\sigma > 0$. Let b_i^T be the ith row of B. Let $\mathcal{A} = \{i \mid b_i^T x_* = 0\}$ and let $\lambda_{\mathcal{A}}$ be the Langrange multiplier vector associated with entries in $\mathcal{A}$. Assume that the entries of $\lambda_{\mathcal{A}}$ are strictly between 1 and -1. Let $B_{\mathcal{A}}$ be the matrix containing rows whose indices are in $\mathcal{A}$. Now set $\tilde{x}_0 = x_0 + n$ where n is a perturbation. Let $\tilde{x}_*$ be the optimal solution to

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - \tilde{x}_0\|_2 \leq \tilde{\sigma}.$$

Then if n is sufficiently small there exists a scalar, $\tilde{\sigma}$, so that the solution $\tilde{x}_*$ to the perturbed problem satisfies

$$B_{\mathcal{A}} \tilde{x}_* = B_{\mathcal{A}} x_*$$

and

$$\operatorname{sgn}(b_i^T \tilde{x}_*) = \operatorname{sgn}(b_i^T x_*) \ \textit{if } i \in \mathcal{A}^C.$$

Combining Theorem 3 with Theorems 1 and 2 leads to a generalization of the noiseless results to the noisy situation. If the noise is sufficiently small, then there is a range of σ values such that Image Simplification preserves the locations of edges in signals and squares in images.

Corollary 1 *Let x_0 be either a piecewise constant function or an image of nested squares, with R_i the regions on which x_0 is constant. Assume that R_i is an extremal region for every i. Suppose that x_0 is corrupted by noise to give $\tilde{x}_0 = x_0 + n$, and define $\tilde{x}_*$ so that $\tilde{x}_* - x_0 = \tilde{\delta}_i$ on R_i. Suppose n is sufficiently small. Then there exist $\tilde{\sigma}_{min}$ and $\tilde{\sigma}_{max}$ so that if $\tilde{\sigma}_{min} < \tilde{\sigma} < \tilde{\sigma}_{max}$ then $\tilde{x}_*$ is the optimal solution to*

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - \tilde{x}_0\|_2 \leq \tilde{\sigma}$$

where $B = B_{1D}$ or $B = B_{2D}$ as appropriate and $\tilde{x}_*$ depends on $\tilde{\sigma}$. Furthermore the edges in $\tilde{x}_*$ have the same direction as the edges in x_0.

Corollary 1 requires that all of the regions be extremal. This means that the signal or image must alternate "hills" and "valleys" rather than having an ascending or descending staircase pattern. Without this assumption, the corollary does not hold. All of these results are proved in the appendix. The results obtained by Strong and Chan for the continuous case using (1) are very similar; our results are different because we use formulation (3) in the discrete setting. The two-dimensional result uses only orthogonal differences. When both orthogonal and diagonal differences are used, computations have revealed that Image Simplification does not preserve squares. We conjecture that in this situation, Image simplification preserves octagons. We have also observed from our computations that if σ is too large, then adjacent intervals of the signal, or adjacent regions in the image will be combined into a single region. All of the remaining edges will still be accurately preserved. Although we have only shown that squares are preserved, in practice, edges in two-dimensional images are preserved very well for a a variety of shapes.

Sharpness of edges can sometimes be further improved by introducing weights in the total variation. In one dimension, the weighted formulation is

$$\min \sum_i w_i |x_i - x_{i-1}| \quad \text{st} \quad \|x - x_0\|_2 \leq \sigma.$$

The weighted formulation produces sharper edges when the signal to noise ratio is high (See Figure 2). We select the weights based on the initial image x_0. Where the difference between two pixels in x_0 is large, we choose a small weight; where this difference is small, we use a large weight. Our experiments have shown weighted Image Simplification to be weakly dependent on the exact function used to compute these weights; most strictly positive concave functions produce similar results. We have therefore chosen to use $w(d) = e^{-\alpha d}$ because of its simple form. When the signal to noise ratio is small, weighting data obtained from the original image is noisy and interferes with edge identification. Experiments have confirmed that, in this case, the unweighted algorithm is superior.

Figure 2: On the left is the initial image x_0, a slice from a lung nodule. In the center is the result of the unweighted algorithm. On the right is the result of the weighted algorithm.

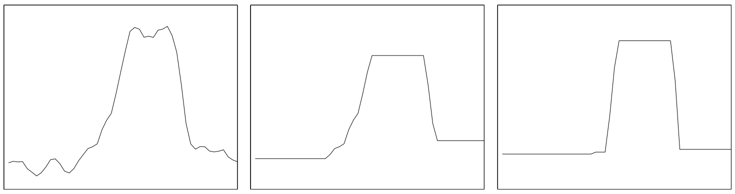
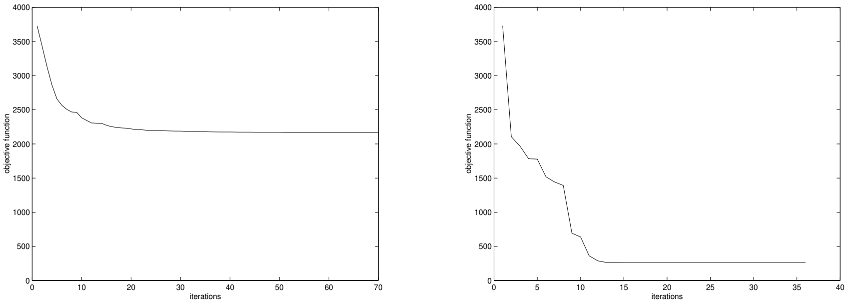


Figure 3: On the left is the convergence of a steepest descent method; on the right is the convergence of an affine scaling method for the same problem. The steepest descent method gets stuck at an objective function value much larger than the value achieved by the affine scaling method.



3 Solving the Image Simplification problem

The Image Simplification optimization problem (3) cannot be solved with standard optimization approaches. The objective function is piecewise linear, and there is a quadratic constraint. A steepest descent type method can get stuck far from the optimum (see Figure 3); therefore, we use a modification of affine scaling, a method introduced by Coleman and Li for solving unconstrained linear ℓ_1 problems [5]. The dimension of the problem can be very large. In order to process an $n \times n$ image, we need to handle n^2 variables. Therefore it is crucial to exploit sparsity.

The weighted total variation can be written as $\|Bx\|_1$ for an appropriate matrix B. The matrix B is very sparse: it has only two nonzero entries per row. With the definitions

$$
\begin{aligned}
r &= \begin{bmatrix} \sigma^2 - (x - x_0)^T (x - x_0) \\ Bx \end{bmatrix} & \qquad J &= -2(x - x_0) \\
g &= \begin{bmatrix} 0 \\ \operatorname{sgn}(Bx) \end{bmatrix} & \qquad M &= \begin{bmatrix} J & B^T \end{bmatrix}
\end{aligned}
\tag{6}
$$

the complementarity conditions for (3) can be written

$$\begin{aligned}&M\lambda = 0\\ &\mathrm{diag}(r)(g-\lambda) = 0.\end{aligned}$$

Applying Newton's method to these equations gives

$$\begin{bmatrix} -2\lambda_1 I & M \\ \mathrm{diag}(g-\lambda)M^T & -\mathrm{diag}(r) \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda + \Delta\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathrm{diag}(r)g \end{bmatrix} \tag{7}$$

where I is the identity matrix. The second equation can be solved for $\lambda + \Delta\lambda$ to obtain

$$\lambda + \Delta\lambda = g + \frac{\mathrm{diag}(g-\lambda)}{\mathrm{diag}(r)} M^T \Delta x. \tag{8}$$

Set $s_i = (g_i - \lambda_i)/r_i$. The first entry of s is s_1 and the rest of s is s_{rest}. The multiplier λ_1 cannot be computed accurately from (8) because $g_1 = 0$ and r_1 approaches zero as an optimal solution is approached. Therefore, eliminate all of the λ values except λ_1 from (7) to get

$$\begin{bmatrix} -2\lambda_1 I + B^T \mathrm{diag}(s_{\mathrm{rest}})B & J \\ \lambda_1 J^T & r_1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda_1 + \Delta\lambda_1 \end{bmatrix} = \begin{bmatrix} -B^T g_{\mathrm{rest}} \\ 0 \end{bmatrix}.$$

In order to solve this system, write $-2\lambda_1 I + B^T\mathrm{diag}(s_{\mathrm{rest}})B = R^T R$ using a sparse Cholesky factorization. Then apply Gaussian elimination to obtain

$$\lambda_1 + \Delta\lambda_1 = \frac{\lambda_1 J^T (R^T R)^{-1}(-B^T g_{\mathrm{rest}})}{\lambda_1 J^T (R^T R)^{-1} J - r_1}.$$

To solve an Image Simplification problem, we begin by using a steepest descent method, using a line search which avoids points of nondifferentiability. The steepest descent iteration is much less expensive than the Newton type iteration, and it usually makes good progress initially. When the objective function is decreasing slowly, or after a certain number of iterations, we switch to the Newton type iteration described above. We have applied this algorithm to a variety of images.

Our first example illustrates the ability of Image Simplification to recover edges in a simple noisy image. In image processing, the noise level is often characterized by the *signal to noise ratio* which is the ratio of the signal power to the noise power. This ratio is generally expressed in decibels (dB):

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\text{signal power}}{\text{noise power}}\right) \text{ dB}.$$

The power of a signal or image x is $\|x - \mathrm{avg}(x)\|_2^2$. The power of random noise is the variance.

We exhibit Image Simplification on a 64×64 image consisting of alternating bands of values 115 and 140. The image is corrupted by Gaussian noise n with variance 20. This gives a SNR of -4.5dB. This corrupted image is shown in the top left of Figure 4. Below it is the graph of a slice of the original image. We applied edge detection (as described in Section 4) to the original image; the result is a dense cloud of edge points shown at the bottom left of Figure 4.

Image Simplification removes the noise and retains a low contrast version of the original image. The result is displayed in the top right of Figure 4. Beneath that is the cross section where the three stripes are clearly visible. The result of edge detection appears at the bottom right. The edges of the rings in the original image are completely restored. In this case, we obtain a good restoration of the edge locations in the image even though the image is not one of the special cases we have analyzed.

In Figure 5, the results of applying Image Simplification to a pulmonary nodule image are displayed. The top section shows a surface plot of the original image. It has a bumpy surface and various details around the base. The two slices show that there are lots of small edges both at the top of the nodule and around it. After Image Simplification, these extraneous details have been smoothed out. The two cross sections show a simple rectangular structure.

4 Segmentation of a processed image

After image simplification is completed, it is necessary to perform a second phase, *object extraction*, to extract the objects from the image. Our object extraction approach begins with edge detection. Objects are bounded by loops, so we identify loops in the edge data. When edge points do not lie on loops, we close off the loops by using level sets of the image. This procedure sometimes produces duplicate loops surrounding the same object. To associate a single loop with each object, we define a distance measure and group loops together if they are close to each other. We then collapse each group of loops into a single loop which bounds the segmented object. The object extraction phase can be divided into the following five steps: edge detection, extraction of loops, edge fragment closure, loop classification, and loop collapsing.

Edge detection In order to identify edges in the image x, we convolve the image with the Prewitt derivative approximation matrices which produce the components of the gradient:

$$P_{\text{horiz}} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad P_{\text{vert}} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

Next we identify local maxima in the gradient approximation. If a pixel's gradient magnitude value is larger than the interpolated values along the gradient direction, we identify it as a local maximum. Because pulmonary nodules can have faintly defined edges where they meet blood vessels, we do not threshold based on edge strength. We are interested in the nodules rather than other features of the lung images so we use knowledge of the range of intensity values for pulmonary nodules to select a cutoff. We then consider connected clusters

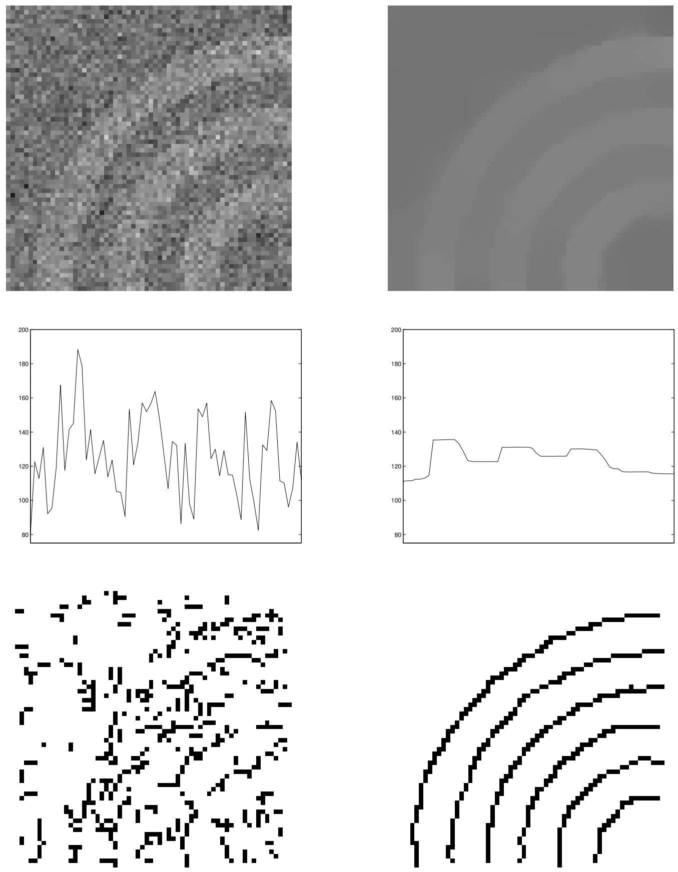


Figure 4: The first image in the left column is the noisy original image. The second picture in the left column shows the values on a slice of that image. The bottom picture in the left column shows the results of edge detection applied to this image. The right hand column shows the result of Image Simplification.

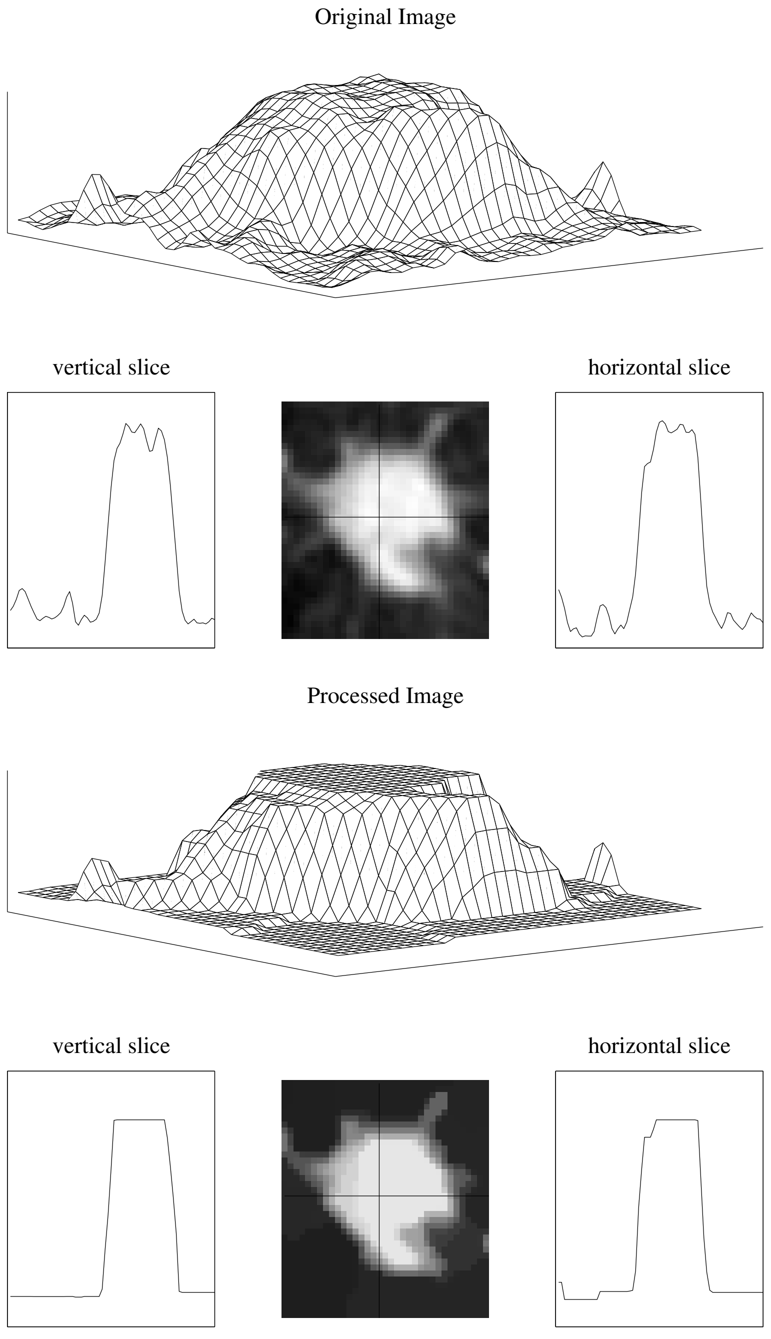


Figure 5: At the top is the original image. The bottom shows the result of Image Simplification applied to the top image.

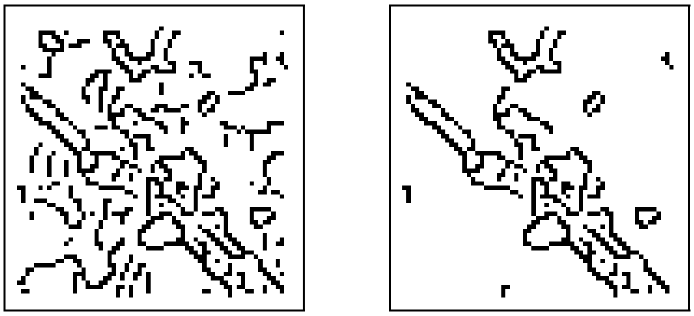


Figure 6: On the left is the result of edge detection applied to a pulmonary nodule image without thresholding. On the right is the same edge detection method with thresholding on the intensity values on the original image.

A	B

	B
A	

	B
A	C

Figure 7: Pixels containing letters are on; empty pixels are off. At the left, pixels A and B are connected by an edge in the graph because they are horizontally adjacent. In the center, pixels A and B are connected by an edge. But at the right, the addition of pixel C breaks the direct edge connection. Pixel C is a mutual neighbor to A and B, so A and B are not connected by an edge in the graph; pixels A and C are connected, as are pixels C and B.

of edges and retain those connected clusters which have at least one edge point with image intensity value above the cutoff (see Figure 6). Once edge detection is completed, it is necessary to organize the edge pixels into meaningful structures.

Loop extraction The second step of the object extraction algorithm is to extract loops from the edge data. We treat each cluster as a graph G by letting the nonzero pixels be the vertices of the graph. We define the edge set of the graph by connecting any two vertically or horizontally adjacent pixels with an edge. We connect diagonally adjacent pixels when they do not have a mutual neighbor defined by vertical or horizontal connections. See Figure 7.

Objects in the edge data correspond to circuits in the graph. The circuits of primary interest are those circuits which correspond to only one object rather than a collection of objects. Below we define the notion of a minimal circuit to identify these circuits. Let L be the set of pixels that form a circuit and define $\text{int}(L)$ to be the finite set of pixels surrounded by L. A circuit L is *minimal* in the graph G if the only circuit in $G \cap (L \cup \text{int}(L))$ is L itself. See Figure 8 for an example.

In order to identify the circuits of interest, we first form a fundamental set of circuits. A set of *fundamental circuits* of a graph is obtained from a spanning

Figure 8: The circuit ACDF is not minimal in the graph because of the circuits ABEF and BCDE which make use of segment BE in the interior of ACDF.

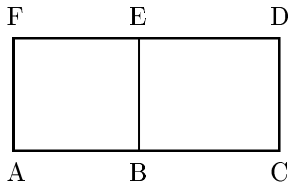
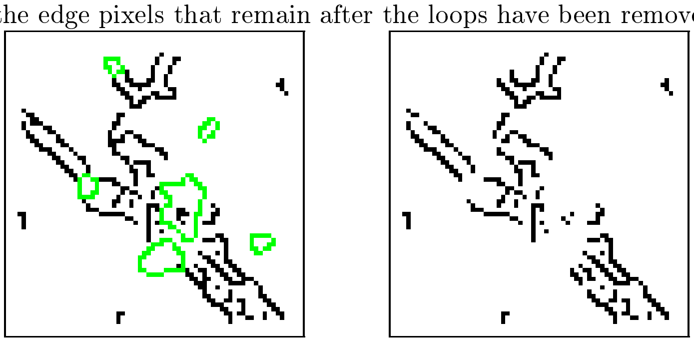


Figure 9: On the left the loops identified by the algorithm are shown. On the right are the edge pixels that remain after the loops have been removed.

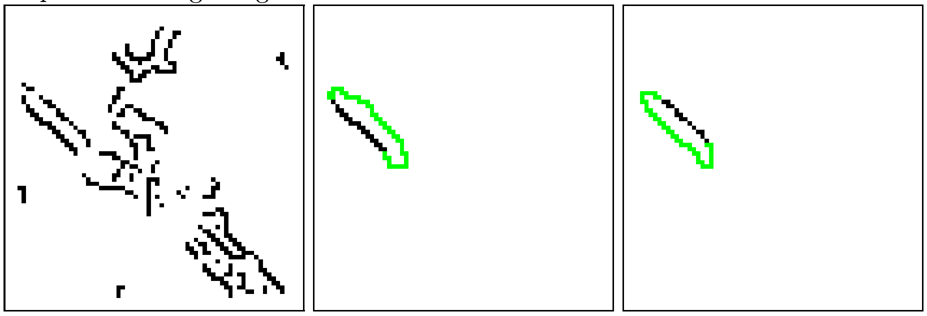


tree of the graph. Adding any edge of the graph to the spanning tree creates one of the fundamental circuits. Write circuits in binary vectors by setting the ith component equal to 1 if edge i is in the circuit, and zero otherwise. With this representation, all circuits in a graph can be represented as sums modulo 2 of the vectors corresponding to the fundamental circuits [4].

To produce a final list of circuits, we use a greedy heuristic to approximate the set of minimal circuits of the graph. Start with C equal to the empty set, and consider every circuit of the graph starting with the shortest one and proceeding in order by size. Whenever a circuit contains edges not already represented in C, add that circuit to C. We remove all of the pixels represented in C from the edge data. See Figure 9.

The algorithm described so far fails to extract loops which are visually obvious but fail to close because they lack one or two pixels. We search for partial loops by considering the leaves, or endpoints, of each edge tree. For a given edge tree, if there is a path which starts at an endpoint, is sufficiently long, and has its other endpoint close enough to the starting point, then it is identified as a partial loop and closed off. We remove all of the pixels which lie on partial loops from the edge data. Removing these pixels leaves behind a collection of edge pixels which are not associated with any loops; however, these edge fragments may mark the presence of important image features.

Figure 10: On the left is the set of edge data. The two right images show how the same object is encircled in two different ways by using level sets to make a loop from an edge fragment.



Edge fragment closure The next step is to make sense of the remaining edge data. For a given cluster of edge pixels, we pick an endpoint e_0 from the set E of endpoints. We then find a contour line passing through e_0. We select a second endpoint e_1 from E which minimizes the distance to the contour. Let c be the point on the contour closest to e_1. There is a unique path from e_0 to e_1 in the edge cluster, and there are two paths from e_0 to c. We select one of those two paths and join it together with the path from e_0 to e_1 and the line segment from e_1 to c. The result is a closed loop. Having finished with our selected endpoint e_0, we now continue with the remaining endpoints in E (including e_1), until we have constructed $|E|$ loops. See Figure 10 for an example. This procedure can produce several loops that correspond to a single object in the image. In order to make sense of this data, we use a loop classification process.

Loop classification The loops constructed in the last two steps cover all of the edge pixels found by the edge detection algorithm. However, some objects will be identified with multiple boundary loops as shown in Figure 10. In order to group together contours which encircle the same object, we place a distance measure on loops.

For a loop L, define $\text{ext}(L)$ to be all the pixels which are not in L and not in $\text{int}(L)$. Use $\mu(X)$ for the area of the set X. Then the distance between two loops can be defined as the symmetric difference of the interiors:

$$d(L_1, L_2) = \mu\left((\text{int}\, L_1 \setminus \text{ext}\, L_2) \cup (\text{int}\, L_2 \setminus \text{ext}\, L_1)\right).$$

We define the following relative distance measure:

$$\hat{d}(L_1, L_2) = \begin{cases} d(L_1, L_2) / \left(|L_1| + |L_2|\right) & \text{when } \text{int}\, L_1 \cap \text{int}\, L_2 \neq \emptyset \\ \infty & \text{when } \text{int}\, L_1 \cap \text{int}\, L_2 = \emptyset \end{cases}$$

For concentric circles, $\hat{d}$ gives the distance between the loops. For non-overlapping loops, the distance measure uses the area of the loops to measure separation. This can be problematic when the loops are small: all small loops will be close to each other. The infinite weight for non-overlapping loops eliminates this problem.

Using $\hat{d}$, we group the loops together into sets such that the relative distance between loop sets is larger than a cutoff value c using the greedy algorithm given below. The parameter c should be chosen so that each set of loops encircles one object. Loops with no interior are discarded at this point.

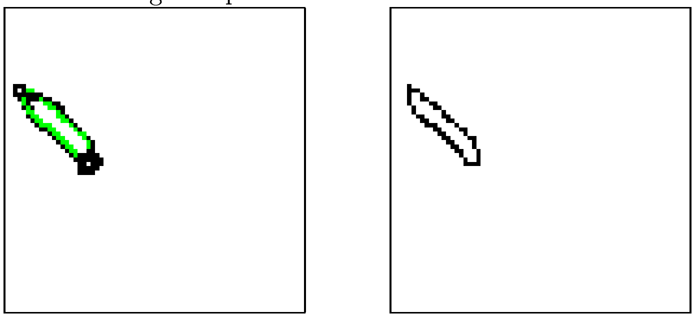
$i = 0$
$\mathcal{L} = \{\text{all loops with nonempty interior}\}$
while $\mathcal{L} \neq \emptyset$
  pick some $L \in \mathcal{L}$
  $G_i = \{L\}$
  repeat
    $N = \{A \in \mathcal{L}\colon \hat{d}(A, L) \leq c\}$
    $\mathcal{L} = \mathcal{L} \setminus N$
    $G_i = G_i \cup N$
  until $N = \emptyset$
  $i = i + 1$
end

The result of this algorithm is a collection of loops sets $\{G_i\}$. Each loop set corresponds to an object. If a loop set contains multiple loops then they must be combined. The next step is to combine these multiple loops into a single loop that identifies the boundary of an object.

Loop collapsing In order to combine loop sets together, we select the longest loop L from a loop set and pick a starting pixel s on that loop. Then we align the other loops with L by finding the pixel in each loop which is nearest to s. We map all of the loops in the loop set onto L, starting at s and using arc length, scaled for loop lengths, to define the map. Then for every pixel in the L, there is a point (possibly interpolated between pixels) on each loop in the loop set. We consider this set of points. If it contains any points which are edge pixels (or lie between edge pixels), then we take the average of those edge points. Otherwise we take the average of all of the points. The result is a single point corresponding to each pixel in L. We connect these points together to produce the final collapsed boundary corresponding to the loop set. Figure 11 shows an example. This step of the algorithm produces a segmentation: a collection of loops which specifies the boundaries of the objects in the image.

Figure 12 shows the results of applying our segmentation algorithm to an entire nodule. This nodule image spans six slices and includes one slice on which the nodule has an internal hole. The algorithm produces excellent segmentations on all slices, with the identified boundary neatly following the boundary visible in the original image. The algorithm is able to separate the nodule from nearby blood vessels in the third and fourth slices.

Figure 11: On the left is a collection of loops. Points which are not edge points appear in black. Edge points are shaded. On the right is the result of collapsing this loop set into a single loop.



5 Concluding Remarks

Our ℓ_1 based total variation method, Image Simplification, has features that make it useful for image segmentation. For two classes of simple functions—piecewise constant signals and nested square images—we have analytically identified the optimal solutions. When Image Simplification is applied to these functions, maximal regions shift downward, minimal regions shift upward, and other regions stay fixed. In the presence of small amounts of noise, edges can be recovered exactly, and the noise can be completely eliminated.

The Image Simplification problem can be attacked by using a steepest descent method. However, this method can get stuck and fail to reach the optimal solution due to nondifferentiability of the objective function. We have described a Newton type method which does not have this problem. Image Simplification problems can be solved more efficiently by using a hybrid method involving both steepest descent and Newton type iterations. The result of Image Simplification is an image that is easy to segment, with sharply defined edges and flat regions. Unlike other variational segmentation approaches, the optimization problem used for Image Simplification is a convex problem and a solution can be computed. In addition, Image Simplification is a global method which operates on the entire image.

We obtain a final segmentation by using edge detection. Edges are grouped into paths and closed into loops by following contours. Contours are classified and collapsed into boundaries of objects.

6 Acknowledgments

We thank David Yankelevitz, Claudia Henschke and Bingshen Zhao of the Cornell Medical School for introducing us to the pulmonary nodule segmentation problem and providing lung nodule data.

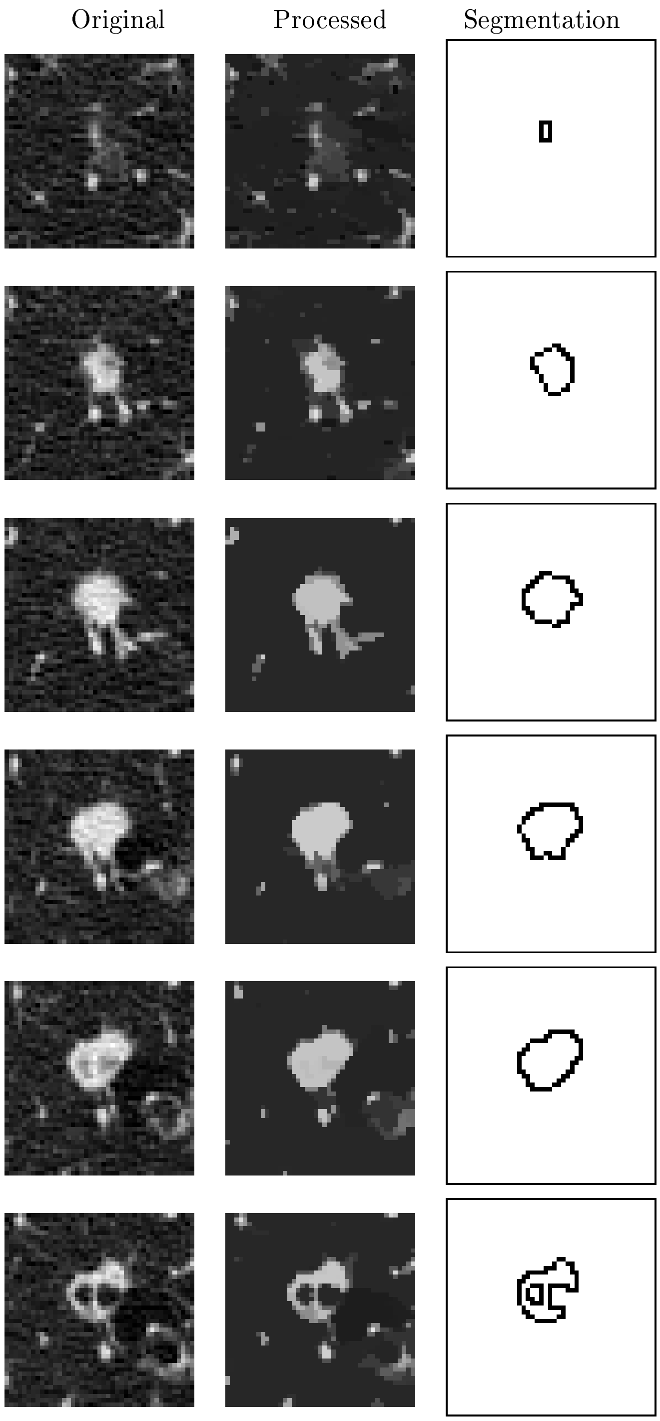


Figure 12: A three-dimensional segmentation.

7 Appendix: proofs

The proofs in this section are all based on the KKT conditions for the optimization problem (3). These conditions can be written:

$$\begin{aligned}&M\lambda = 0\\ &\operatorname{diag}(r)(g-\lambda) = 0\\ &\|Ax - x_0\|_2 \le \sigma\\ &\lambda_1 \le 0\\ &|\lambda_i| \le 1 \text{ for } i > 1\end{aligned} \tag{9}$$

where M, J, r, and g are defined by equation (6). The optimization problem (3) is convex; therefore, to establish that a particular x is a solution to (3) it suffices to exhibit λ such that (9) holds for that x.

Lemma 1 (Uniqueness of Solutions) *Suppose x_* satisfies the optimality conditions (9) for the problem*

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - x_0\|_2 \le \sigma. \tag{10}$$

Let b_i^T be the ith row of B and let $\mathcal{A} = \{i \mid b_i^T x_* = 0\}$. For an $m \times n$ image, let $\dim(x) = mn$. *Then if either*

$$\|x_* - x_0\|_2 = \sigma \text{ and } \lambda_1 \neq 0 \tag{11}$$

or

$$|\lambda_i| < 1 \text{ for } i \in \mathcal{A}^C \text{ and } \operatorname{rank}\{b_i \mid i \in \mathcal{A}\} = \dim(x) \tag{12}$$

then x_* is the unique solution to (10).

Proof: Pick d to be a nonzero feasible direction at x_*. If (11) holds this means that $J^T d > 0$; otherwise d can be any nonzero direction.

Now consider taking a small step in the direction d and evaluating the objective function:

$$\|B(x_* + \epsilon d)\|_1 = \sum_{i \in \mathcal{A}} \left|b_i^T x_* + \epsilon b_i^T d\right| + \sum_{i \in \mathcal{A}^C} \left|b_i^T x_* + \epsilon b_i^T d\right|.$$

In $\mathcal{A}$, we know that $b_i^T x_* = 0$, so $\left|b_i^T x_* + \epsilon b_i^T d\right| = \left|b_i^T x_*\right| + \left|\epsilon b_i^T d\right|$. In $\mathcal{A}^C$, the absolute value term can be rewritten as $\left|b_i^T x_*\right| + \operatorname{sgn}(b_i^T x_*) b_i^T d\epsilon$ for ϵ sufficiently small. This gives the following equality:

$$\|B(x_* + \epsilon d)\|_1 = \|Bx_*\|_1 + \epsilon \left(\sum_{i \in \mathcal{A}} \left|b_i^T d\right| + \sum_{i \in \mathcal{A}^C} g_i b_i^T d \right) \tag{13}$$

where $g_i = \operatorname{sgn}(b_i^T x_*)$.

Because the optimality conditions (9) are satisfied we know that

$$J\lambda_1 + B^T \lambda_{\text{rest}} = J\lambda_1 + \sum_{i>1} b_i \lambda_i = 0.$$

If $i \in \mathcal{A}^C$ then the equation $\text{diag}(r)(g-\lambda) = 0$ from (9) forces $\lambda_i = g_i$. Using this and taking the inner product with d gives

$$\lambda_1 J^T d + \sum_{i \in \mathcal{A}^C} g_i b_i^T d + \sum_{i \in \mathcal{A}} \lambda_i b_i^T d = 0.$$

Taking absolute values inside the rightmost summation gives the inequality

$$\lambda_1 J^T d + \sum_{i \in \mathcal{A}^C} g_i b_i^T d + \sum_{i \in \mathcal{A}} \left|\lambda_i b_i^T d\right| \geq 0,$$

and taking the maximum of the $|\lambda_i|$ coefficients widens the gap to give

$$\lambda_1 J^T d + \sum_{i \in \mathcal{A}^C} g_i b_i^T d + \left(\max_{i \in \mathcal{A}} |\lambda_i|\right) \sum_{i \in \mathcal{A}} \left|b_i^T d\right| \geq 0.$$

Rearranging terms and adding $\sum_{i \in \mathcal{A}} \left|b_i^T d\right|$ to both sides gives

$$\sum_{i \in \mathcal{A}^C} g_i b_i^T d + \sum_{i \in \mathcal{A}} \left|b_i^T d\right| \geq -\lambda_1 J^T d + \left(1 - \max_{i \in \mathcal{A}} |\lambda_i|\right) \sum_{i \in \mathcal{A}} \left|b_i^T d\right|.$$

Call the right hand side of this inequality y. If (11) holds then $\lambda_1 < 0$ by (9) so $-\lambda_1 J^T d > 0$. The second term in y is a sum of nonnegative values, so $y > 0$. If (12) holds then $\lambda_1 = 0$. However, $(1 - \max_{i \in \mathcal{A}} |\lambda_i|) > 0$. Because the rank of $\{b_i \mid i \in \mathcal{A}\}$ is equal to the dimension of the space, mn, there must be some i for which $b_i^T d \neq 0$. Therefore, we can conclude that $y > 0$. Inserting this result into equation (13) reveals that

$$\|B(x_* + \epsilon d)\|_1 \geq \|Bx_*\|_1 + \epsilon y > \|Bx\|_1.$$

Since (10) is a convex optimization problem, we can conclude that x_* is the unique solution to (10). ■

Theorem 1 (Piecewise Constant Signals) *Suppose x_0 is a piecewise constant signal defined on segments R_i, $i = 1 \ldots N$ (where R_i is adjacent only to R_{i+1} and R_{i-1}) and σ is a sufficiently small positive constant. Let*

$$\delta_i = \begin{cases} \frac{\beta_i}{2|\lambda_1|\alpha_i} & \text{if } R_i \text{ is a minimal region} \\ \frac{-\beta_i}{2|\lambda_1|\alpha_i} & \text{if } R_i \text{ is a maximal region} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where α_i and β_i are defined by (5), and the value of $|\lambda_1|$ is specified by

$$4\lambda_1^2 = \frac{\sum_{i \in \mathcal{E}} \frac{\beta_i^2}{\alpha_i}}{\sigma^2}, \tag{15}$$

with $\mathcal{E}$ given by (4). Let $x_* = x_0 + \delta_i$ on R_i.

Then x_* is the unique optimal solution to

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - x_0\|_2 \leq \sigma$$

where $B = B_{1D}$.

Proof: Since Bx_0 and Bx_* are zero at the same points, we can assume that σ is small enough so that $\text{sgn}(Bx_0) = \text{sgn}(Bx_*)$. We will produce a λ which satisfies the optimality conditions (9). First consider the complementarity condition

$$\text{diag}(r)(g - \lambda) = 0. \tag{16}$$

The first entry in r is $\sigma^2 - \|x_* - x_0\|_2^2$ which equals $\sigma^2 - \sum_{i=1}^{N} \alpha_i \delta_i^2$. It's easy to verify from (14) and (15) that $r_1 = 0$. The rest of r is equal to Bx_*, which is zero everywhere except at points of discontinuity where jumps occur in x. For each such point, d, set $\lambda_d = g_d$. Then $r_d(g_d - \lambda_d) = 0$, so (16) holds.

Next we will define the rest of λ so that

$$M\lambda = 0 \tag{17}$$

and

$$\lambda_1 \leq 0 \text{ and } |\lambda_i| \leq 1 \text{ for } i > 1. \tag{18}$$

The matrix M is equal to $[J\ B^T]$. The vector J is defined in (6) by $J = 2(x_0 - x_*)$. Equation (17) can be rewritten as $B^T \lambda_{\text{rest}} = -\lambda_1 J$. Define the Lagrange multiplier λ_1 according to (15) with $\lambda_1 < 0$ in accordance with (18).

In order to help define λ_{rest}, we first examine the B^T operator in detail. Let $\xi = B^T \lambda_{\text{rest}}$. The operator B takes a signal as input and produces differences between adjacent entries; these can be thought of as points between the pixels in the signal. The B^T operator is a map from the points between the pixels to the pixels, so ξ is a signal. The value of a pixel in ξ depends only on the λ values associated with the two points bounding that pixel. Suppose these bounding λ entries are λ_i and λ_{i+1}. The B^T operator takes differences in the opposite direction than B so the value of the entry of ξ will be $\lambda_i - \lambda_{i+1}$. When a pixel is at the end of the signal, one of its bounding points will have no associated entry in λ. Set λ to 0 at these locations to handle this situation without special cases.

The value of λ is already defined on the points bounding R_i. Define it on the interior by linearly interpolating between the two boundary values. These boundary values are always between 1 and -1, so (18) is satisfied. Next we need to verify that $\xi = -\lambda_1 J$. On R_i we know that $-\lambda_1 J = 2\lambda_1 \delta_i$ since $J = 2(x_0 - x_*)$. By using (14) we can write $2\lambda_1 \delta_i = -\text{sgn}(\delta_i)\beta_i/\alpha_i$. Therefore, to complete the proof we need to verify that

$$\xi = \frac{-\text{sgn}(\delta_i)\beta_i}{\alpha_i} \text{ on } R_i. \tag{19}$$

If $1 < i < N$ and R_i is a maximal region, then $\lambda = 1$ on the left side and $\lambda = -1$ on the right side. Therefore, $\xi = 2/\alpha_i$ on R_i which satisfies (19) since $\beta_i = 2$. The sign is correct because $\delta_i < 0$ in R_i. If R_i is minimal then (19) holds as well. If R_i is not an extreme region then the two λ values are equal on both sides of R_i and hence $\xi = 0$ as required. If $i = 1$ or $i = N$ then one of the bounding λ values is zero, so $\xi = \pm 1/\alpha_i$ which satisfies (19) with $\beta_i = 1$. Since (19) holds, we have satisfied the last of the optimality conditions. For this problem, the constraint is tight and $\lambda_1 \neq 0$. Therefore, by Lemma 1, x_* is the unique optimal solution. ■

Theorem 2 (Symmetric Nested Square Images) *Suppose x_0 is an image of nested squares, with R_i, $i = 1 \dots N$, the regions on which it is constant, and σ is a sufficiently small positive constant. Let*

$$\delta_i = \begin{cases} \frac{\beta_i}{2|\lambda_1|\alpha_i} & \text{if } R_i \text{ is a minimal region} \\ \frac{-\beta_i}{2|\lambda_1|\alpha_i} & \text{if } R_i \text{ is a maximal region} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where α_i and β_i are defined by (5), and the value of $|\lambda_1|$ is specified by

$$4\lambda_1^2 = \frac{\sum_{i\in\mathcal{E}} \frac{\beta_i^2}{\alpha_i}}{\sigma^2}, \tag{21}$$

with $\mathcal{E}$ given by (4). Let $x_* = x_0 + \delta_i$ on R_i.

Then x_* is the unique optimal solution to

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - x_0\|_2 \leq \sigma$$

where $B = B_{2D}$.

Proof: Assume that R_1 is the central square region and R_N is the outermost region. Let n_i be the outer side length of R_i and k_i the inner side length of R_i. Since Bx_0 and Bx_* are zero at the same points, we can assume that σ is small enough so that $\mathrm{sgn}(Bx_0) = \mathrm{sgn}(Bx_*)$. We will produce a λ which satisfies the optimality conditions (9). First consider the complementarity condition

$$\mathrm{diag}(r)(g - \lambda) = 0. \tag{22}$$

The first entry in r is $\sigma^2 - \|x_* - x_0\|_2^2$ which equals $\sigma^2 - \sum_{i=1}^N \alpha_i \delta_i^2$. It's easy to verify from (20) and (21) that $r_1 = 0$. The rest of r is equal to Bx_*, which is zero everywhere except at points of discontinuity where jumps occur in x. For each such point, d, set $\lambda_d = g_d$. Then $r_d(g_d - \lambda_d) = 0$, so (22) holds.

Next we will define the rest of λ so that

$$M\lambda = 0 \tag{23}$$

and

$$\lambda_1 \leq 0 \text{ and } |\lambda_i| \leq 1 \text{ for } i > 1. \tag{24}$$

The matrix M is equal to $[J \ B^T]$. The vector J is defined in (6) by $J = 2(x_0 - x_*)$. Equation (23) can be rewritten as $B^T\lambda_{\text{rest}} = -\lambda_1 J$. Define the Lagrange multiplier λ_1 according to (21) with $\lambda_1 < 0$ in accordance with (24).

In order to help define λ_{rest}, we will first examine the B^T operator in detail. Let $\xi = B^T\lambda_{\text{rest}}$. The operator B takes as input an image made of square pixels and produces horizontal and vertical differences which can be regarded as lying on the lines between the pixels. The B^T operator is a map from the lines between the pixels to the pixels, so ξ is an image. The value of a pixel in ξ depends only on the λ values associated with the four lines bounding that pixel:

$$\begin{array}{ccc} & \lambda_i^{\mathrm{v}} & \\ \lambda_i^{\mathrm{h}} & & \lambda_{i+1}^{\mathrm{h}} \\ & \lambda_{i+1}^{\mathrm{v}} & \end{array}$$

The B^T operator takes differences in the opposite direction than B so the pixel in ξ will have the value $\lambda_i^{\mathrm{v}} - \lambda_{i+1}^{\mathrm{v}} + \lambda_i^{\mathrm{h}} - \lambda_{i+1}^{\mathrm{h}}$. When a pixel is at the edge of an image some of its bounding lines will have no associated entry in λ. We assign λ values of 0 to these locations to handle this situation without special cases.

The λ values we select must satisfy $\xi = -\lambda_1 J$. On R_i we know that $-\lambda_1 J = 2\lambda_1\delta_i$. Combining this with (20), we find $2\lambda_1\delta_i = -\mathrm{sgn}(\delta_i)\beta_i/\alpha_i$. Therefore, as we specify values of λ on R_i, we will check that

$$\xi = \frac{-\mathrm{sgn}(\delta_i)\beta_i}{\alpha_i} \text{ on } R_i. \tag{25}$$

We now consider the values of λ in the different regions. Figure 13 may clarify the argument. Since i is fixed we drop the subscripts on n_i and k_i. First consider the values of λ in R_1, and assume that R_1 is a maximal region. (Note that it must be an extreme region because it is completely surrounded by a region with a different value.) This region is bounded on the top and left by $\lambda = 1$ and on the bottom and right by $\lambda = -1$. We define λ to step uniformly in both directions from 1 down to -1. In order to do this, the step size must be $-2/n$. The corresponding value of ξ is $4/n$, with the sign change occuring because we subtract the bottom from the top and the right from the left. The sign of δ_i is -1 in this region, so we can verify (25):

$$\frac{4}{n} = \frac{-(-1)4n}{n^2}.$$

If R_1 is a minimal region then the condition still holds.

Next consider an interior region R_i where $1 < i < N$. If this region is not an extreme region, then it is bounded by λ values which are either all 1 or all -1. Set λ to be uniformly equal to the value on the boundary. All of the differences in both directions are zero, so $\xi = 0$ on R_i as required. Otherwise, assume that R_i is a maximal region (the minimal case being analogous). Because the squares are centered, we can fill in the λ values using symmetry. Therefore, it suffices to give values for one strip of the hollow square (labeled R_i in Figure 13). At the top and left of the square, λ is equal to 1. Fill in the $(n-k)/2 \times (n-k)/2$ square in the upper left with uniform steps of $-2/(n-k)$. This sets $\lambda = 0$ on the lower and right boundaries of that square. It gives a ξ value of $4/(n-k)$ for the region. In the strip to the left of the central hole, set the vertical λ values to zero. Let the horizontal λ values descend from 1 on the left to -1 on the right by steps of $-4/(n-k)$. The ξ values in this area are also $4/(n-k)$. Continuing down the strip, we fill in the vertical λ values just as we did in the top corner by stepping from 0 down to -1 in steps of $2/(n-k)$. The remaining portions of λ are filled in symmetrically. The value of ξ is uniformly $4/(n-k)$ on R_i. The sign of δ_i is -1. Checking equation (25) we have

$$\frac{4}{n-k} = \frac{-(-1)4(n+k)}{n^2-k^2}$$

so the equation is satisfied.

The last region to consider is R_N. This region must be an extreme region; again assume it is a maximal region. Unlike the other regions, this one is bounded by $\lambda = 0$ around the outside. In the top left corner, define λ in both directions to descend from 0 to $-k/(n+k)$ which requires steps of $-2k/((n-k)(n+k))$ since this square has side $(n-k)/2$. Moving down the strip to the left of the hole, define λ to rise from $-k/(n+k)$ up to $+k/(n+k)$ in the vertical direction, which requires steps of length $2/(n+k)$. In this region set the horizontal steps to be the linear interpolation from the zero on the left to the -1 on the right; this means steps of $-2/(n-k)$. Continuing down to the bottom corner, we go from $k/(n+k)$ down to zero in steps of $-2k/((n-k)(n+k))$. The rest of λ can be filled in by symmetry. The resulting value of ξ along the side is

$$\frac{2}{n-k} - \frac{2}{n+k} = \frac{4k}{(n-k)(n+k)} = \frac{4k}{n^2-k^2}.$$

It's easy to see that the same ξ value arises in the corners, and easy to verify (25).

Since (25) holds for all R_i we have established (23). Throughout the construction of λ, we have defined λ by interpolating linearly between 1 and -1 or between 0 and ± 1. Therefore, the resulting entries must all lie between 1 and -1, so we have satisfied (24). The constraint is tight and $\lambda_1 \neq 0$, so by Lemma 1, x_* is the unique optimal solution. ■

Theorem 3 (Noise Perturbed Solutions) *Suppose x_* is the optimal solution to*

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - x_0\|_2 \leq \sigma$$

where B is a matrix and $\sigma > 0$. Let b_i^T be the ith row of B. Let $\mathcal{A} = \{i \mid b_i^T x_* = 0\}$ and let $\lambda_{\mathcal{A}}$ be the Lagrange multiplier vector associated with entries in $\mathcal{A}$. Assume that the entries of $\lambda_{\mathcal{A}}$ are strictly between 1 and -1. Let $B_{\mathcal{A}}$ be the matrix containing rows whose indices are in $\mathcal{A}$. Now set $\tilde{x}_0 = x_0 + n$ where n is a perturbation. Let $\tilde{x}_*$ be the optimal solution to

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - \tilde{x}_0\|_2 \leq \tilde{\sigma}.$$

Then if n is sufficiently small there exists a positive scalar, $\tilde{\sigma}$, so that the solution $\tilde{x}_*$ to the perturbed problem satisfies

$$B_{\mathcal{A}}\tilde{x}_* = B_{\mathcal{A}}x_* \tag{26}$$

and

$$\operatorname{sgn}(b_i^T \tilde{x}_*) = \operatorname{sgn}(b_i^T x_*) \ \textit{if}\ i \in \mathcal{A}^C. \tag{27}$$

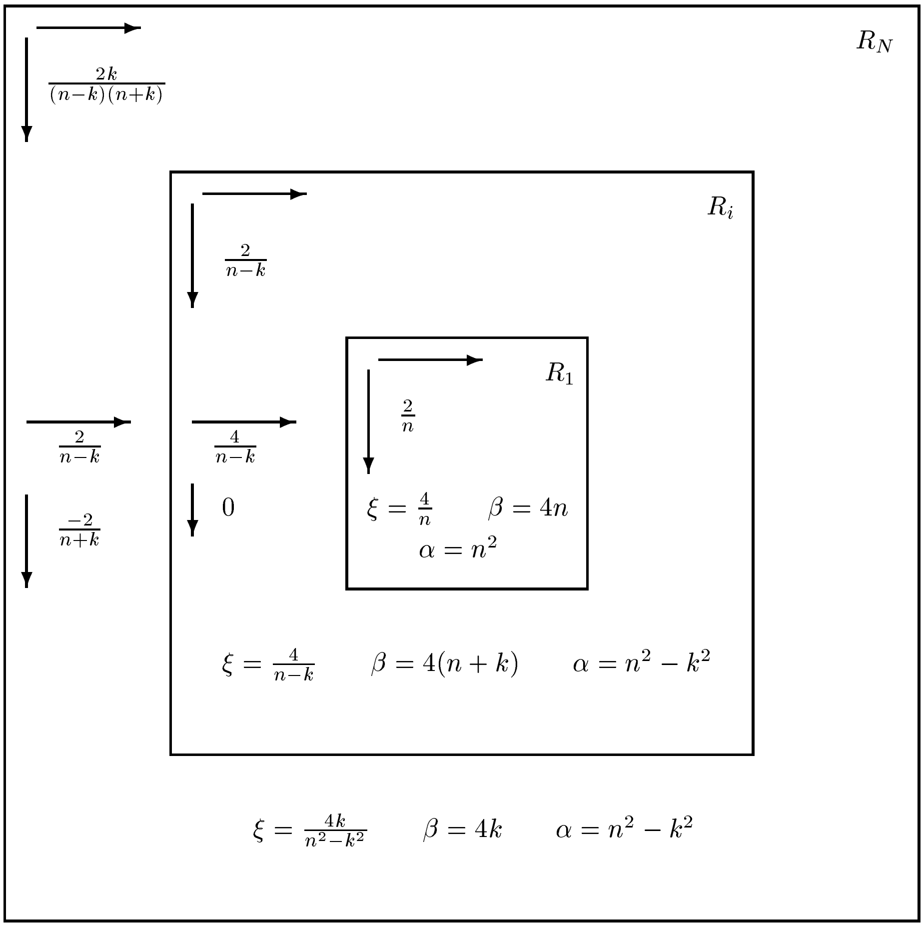


Figure 13: This diagram shows the step size which is used to select λ in different regions of the the image. The numbers next to the arrows indicate the difference between values of λ in that direction. In each region, the sum of the vertical difference and the horizontal difference is ξ, and $\mathrm{sgn}(\delta)\,|\xi|\,\alpha/\beta = 1$.

Proof: Let $\Delta x_* = n - B_{\mathcal{A}}^T(B_{\mathcal{A}}^T)^+ n$ where $(B_{\mathcal{A}}^T)^+$ is the pseudoinverse of $B_{\mathcal{A}}^T$. We will show that $\tilde{x}_* = x_* + \Delta x_*$ by considering the optimality conditions (9). First note that it is easy to verify that $B_{\mathcal{A}}\Delta x_* = 0$ using the Moore–Penrose properties of the pseudoinverse, hence establishing (26) for our choice of $\tilde{x}_*$.

The equation $M\lambda = 0$ can be rewritten as

$$2(x_0 - x_*)\lambda_1 + B_{\mathcal{A}^C}\lambda_{\mathcal{A}^C} + B_{\mathcal{A}}\lambda_{\mathcal{A}} = 0 \tag{28}$$

where $B_{\mathcal{A}^C}$ is formed from rows whose indices are not in $\mathcal{A}$ and $\lambda_{\mathcal{A}^C}$ is the multiplier vector corresponding to $\mathcal{A}^C$. Note that $\lambda_{\mathcal{A}^C}$ is equal to the sign of the residuals. For $i \in \mathcal{A}^C$, we know that $b_i^T x_*$ is not zero. Therefore, since Δx_* is just a matrix times n, as long as n is small enough, $b_i^T \tilde{x}_*$ will have the same sign as $b_i^T x_*$ for $i \in \mathcal{A}^C$. This establishes equation (27).

Next define $\tilde{\lambda}_{\mathcal{A}} = \lambda_{\mathcal{A}} - 2(B_{\mathcal{A}}^T)^+ n\lambda_1$. Set $\tilde{\lambda}_1 = \lambda_1$ and set $\tilde{\lambda}_{\mathcal{A}^C} = \lambda_{\mathcal{A}^C}$. Substituting these definitions into equation (28) gives

$$2(\tilde{x}_0 - n - \tilde{x}_* + \Delta x_*)\tilde{\lambda}_1 + B_{\mathcal{A}^C}^T\tilde{\lambda}_{\mathcal{A}^C} + B_{\mathcal{A}}^T(\tilde{\lambda}_{\mathcal{A}} + 2(B_{\mathcal{A}}^T)^+ n\tilde{\lambda}_1) = 0.$$

Expanding Δx_* shows that

$$2(\tilde{x}_0 - \tilde{x}_* - B_{\mathcal{A}}^T(B_{\mathcal{A}}^T)^+ n)\tilde{\lambda}_1 + B_{\mathcal{A}^C}^T\tilde{\lambda}_{\mathcal{A}^C} + B_{\mathcal{A}}^T(\tilde{\lambda}_{\mathcal{A}} + 2(B_{\mathcal{A}}^T)^+ n\tilde{\lambda}_1) = 0.$$

The two expressions involving $(B_{\mathcal{A}}^T)^+$ cancel leaving

$$2(\tilde{x}_0 - \tilde{x}_*)\tilde{\lambda}_1 + B_{\mathcal{A}^C}^T\tilde{\lambda}_{\mathcal{A}^C} + B_{\mathcal{A}}^T\tilde{\lambda}_{\mathcal{A}} = 0$$

which is equation (28) rewritten for the perturbed problem. Since (26) and (27) hold, we know that the $\mathcal{A}$ is the same for the perturbed problem.

Set $\tilde{\sigma} = \|\tilde{x}_* - \tilde{x}_0\|_2$. Then the optimality equation $\mathrm{diag}(\tilde{r})(\tilde{g} - \tilde{\lambda}) = 0$ will be satisfied since (27) holds. Since $\tilde{\lambda}_1 = \lambda_1$, the constraint on $\tilde{\lambda}_1$ is satisfied. The last constraint that must be checked is $|\tilde{\lambda}_i| \le 1$ for $i > 1$. This needs to be checked only for $\tilde{\lambda}_{\mathcal{A}}$ because the other $\tilde{\lambda}$ values are fixed at ± 1. In order to satisfy this condition, we need the entries of

$$\lambda_{\mathcal{A}} - 2(B_{\mathcal{A}}^T)^+ n\lambda_1 \tag{29}$$

to lie between 1 and -1. Because we assumed that the entries of $\lambda_{\mathcal{A}}$ are strictly between 1 and -1, we can conclude that for sufficiently small n, this condition will hold which completes the proof. ■

Remark: If there are entries in $\lambda_{\mathcal{A}}$ which are equal to ± 1 then the conclusion can hold if the noise function satisfies an additional technical condition. For example, if every entry in $-2(B_{\mathcal{A}}^T)^+ n\lambda_1$ which corresponds to a λ value of 1 (-1) has sign equal to -1 (1), then the results of the theorem hold.

Corollary 1 *Let x_0 be either a piecewise constant function or an image of nested squares, with R_i the regions on which x_0 is constant. Assume that R_i is an extremal region for every i. Suppose that x_0 is corrupted by noise to give $\tilde{x}_0 = x_0 + n$, and define $\tilde{x}_*$ so that $\tilde{x}_* - x_0 = \tilde{\delta}_i$ on R_i. Suppose n is sufficiently* *small. Then there exist $\tilde{\sigma}_{min}$ and $\tilde{\sigma}_{max}$ so that if $\tilde{\sigma}_{min} < \tilde{\sigma} < \tilde{\sigma}_{max}$ then $\tilde{x}_*$ is the optimal solution to*

$$\min_x \|Bx\|_1 \quad s.t. \quad \|x - \tilde{x}_0\|_2 \leq \tilde{\sigma}$$

where $B = B_{1D}$ or $B = B_{2D}$ as appropriate and $\tilde{x}_*$ depends on $\tilde{\sigma}$. Furthermore the jumps in $\tilde{x}_*$ have the same sign as the jumps in x_0.

Proof: Let $x_*(\sigma) = x_0 + \delta_i(\sigma)$ on R_i. By Theorem 1 or Theorem 2, we know that there is a $\sigma_{\max}$ such that for any σ satisfying $0 < \sigma < \sigma_{\max}$, the $\delta_i(\sigma)$ can be chosen so that $x_*(\sigma)$ is the optimal solution to

$$\min_x \|Bx\|_1 \quad \text{s.t.} \quad \|x - x_0\|_2 \leq \sigma.$$

Because R_i is an extremal region for all i, we know from the proof of Theorems 1 and 2 that the multipliers are equal to ± 1 only at points where jumps occur in the solution, so $\lambda_{\mathcal{A}}$ is strictly between 1 and -1. Furthermore, the value of $\lambda_{\mathcal{A}}$ is independent of the value of σ.

We want to establish that the results of Theorem 3 hold over an interval of σ values. To do this, we must show that there exists a fixed n which is small enough independent of σ over some interval. In the proof of Theorem 3, the perturbation is bounded in two places. The perturbation is required to be small enough so that (27) holds. This condition is independent of σ. We require that n be small enough so that (27) holds. The second bound on the perturbation is equation (29) from the proof of Theorem 3 which requires that $\lambda_{\mathcal{A}} - 2(B_{\mathcal{A}}^T)^+ n\lambda_1$ have entries lying between 1 and -1. Equation (15) from Theorem 1 or (21) from Theorem 2 allows us to rewrite that expression as

$$\lambda_{\mathcal{A}} - \left(\sqrt{\sum_{i\in\mathcal{E}} \frac{\beta_i^2}{\alpha_i}} (B_{\mathcal{A}}^T)^+ \right) \frac{n}{\sigma}. \tag{30}$$

We have shown that $\lambda_{\mathcal{A}}$ does not depend on σ, and we also know that β_i, α_i, and $B_{\mathcal{A}}$ are independent of σ. Therefore, there is a minimal value of σ so that (30) is between 1 and -1 (where n is assumed to be fixed). Define $\sigma_{\min}$ to be this minimal value of σ. It is possible that $\sigma_{\min} > \sigma_{\max}$. The value of $\sigma_{\min}$ can be reduced by making n smaller; we require that n be small enough so that $\sigma_{\min} < \sigma_{\max}$.

We have shown that if $\sigma_{\min} < \sigma < \sigma_{\max}$ then the conclusions of Theorem 3 hold. Theorem 3 provides information about $\tilde{x}_*(\sigma)$ which is the optimal solution to

$$\min_x \|Bx\|_1 \quad \text{s.t.} \quad \|x - \tilde{x}_0\|_2 \leq \tilde{\sigma}(\sigma)$$

where $\tilde{\sigma}(\sigma)$ is specified by the theorem. In this setting, $B_{\mathcal{A}}$ from Theorem 3 is a matrix which takes differences between all adjacent pixels that lie within the same constant regions of x_0; therefore, $B_{\mathcal{A}} x_*(\sigma)$ is equal to zero since all of these differences are zero. Theorem 3 says that $B_{\mathcal{A}}\tilde{x}_*(\sigma) = B_{\mathcal{A}} x_*(\sigma)$ so we can conclude that $\tilde{x}_*(\sigma)$ also has zero differences at the same locations, hence $\tilde{x}_*(\sigma)$ is constant on the same regions as $x_*(\sigma)$. By conclusion (27) from Theorem 3 we see that the jumps in $\tilde{x}_*(\sigma)$ have the same sign as the jumps in x_*, and furthermore, these jumps are still present—they have not been smoothed out.

The value of $\tilde{\sigma}(\sigma)$ is given in the proof of Theorem 3:

$$\tilde{\sigma}(\sigma) = \|\tilde{x}_*(\sigma) - x_0 - n\|_2 .$$

Theorem 3 also says that $\tilde{x}_*(\sigma) = x_*(\sigma) + \Delta x_*$ where Δx_* depends on n but not on x_* or σ. Hence

$$\tilde{\sigma}(\sigma) = \|x_*(\sigma) + \Delta x_* - x_0 - n\|_2 .$$

The function $\tilde{\sigma}$ is a continuous function of $x_*(\sigma)$. By examining the definition of $x_*(\sigma)$ given in Theorems 1 and 2, it is clear that $x_*(\sigma)$ is a continuous function of σ. Therefore the composition $\tilde{\sigma}(\sigma)$ is a continuous function of σ.

For the interval $(\sigma_{\min}, \sigma_{\max})$ we have established that $\tilde{x}_*(\sigma)$ solves the problem for parameter $\tilde{\sigma}$. Since $\tilde{\sigma}$ depends continuously on σ, we can conclude that $\tilde{x}_*(\sigma)$ solves the problem for a connected set of values of $\tilde{\sigma}$.

The final step is to establish that this connected set contains more than one point. This is easy to establish because $x_*(\sigma)$ is a one to one function, hence $\tilde{x}_*(\sigma)$ is also one to one. Therefore if $\sigma_1 \neq \sigma_2$ then $\tilde{x}_*(\sigma_1) \neq \tilde{x}_*(\sigma_2)$. By uniqueness of solutions, it follows that $\tilde{\sigma}(\sigma_1) \neq \tilde{\sigma}(\sigma_2)$, so $\tilde{\sigma}(\sigma)$ is one to one. Hence there is a nonempty connected set of values as required. This set must contain an open interval. Call that interval's lower bound $\tilde{\sigma}_{\min}$ and call the upper bound $\tilde{\sigma}_{\max}$. ■

References

[1] T. Asano and N. Yokoya. Image segmentation schema for low-level computer vision. *Pattern Recognition*, 14:267–273, 1981.

[2] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[3] J. Bryant. On the clustering of multidimensional pictorial data. *Pattern Recognition*, 11:115–125, 1979.

[4] Nicos Christofides. *Graph Theory: An Algorithmic Approach*. Academic Press, London, 1975.

[5] Thomas F. Coleman and Yuying Li. A globally and quadratically convergent affine scaling method for linear ℓ_1 problems. *Mathematical Programming*, 56:189–222, 1992.

[6] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15:11–15, 1972.

[7] K. S. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3–16, 1981.

[8] Maryellen L. Giger, Kyongtae T. Bae, and Heber MacMahon. Computerized detection of pulmonary nodules in computed tomography images. *Investigative Radiology*, 29(4):459–465, April 1994.

[9] Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–32, Jan 1985.

[10] Yuying Li and Fadil Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5(6):987–995, 1996.

[11] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1:169–182, 1972.

[12] Jean-Michael Morel. *Variational Methods in Image Segmentation*. Birkhauser, 1995.

[13] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 85CH2145–1, pages 22–26, 1985.

[14] Nikhil R. Pal and Sankar K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[15] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:288–305, 1993.

[16] David M. Strong and Tony F. Chan. Exact solutions to the total variation regularization problem. Math Department CAM Report 96–41, UCLA, 1996.

[17] Shunsuke Toshioka, Keizo Kanazawa, Noboru Niki, Hitoshi Satoh, Hironobu Ohmatsu, Kenji Eguchi, and Noriyuki Moriyama. Computer aided diagnosis system for lung cancer based on helical CT images. In *Medical Imaging 1997*, volume 3034 of *Proceedings of SPIE*, pages 975–84, 1997.

[18] J. S. Weszka. A survey of threshold selection techniques. *Computer Graphics and Image Processing*, 7:259–265, 1978.

[19] S. W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.