

# A VARIATIONAL APPROACH TO VIDEO SEGMENTATION FOR BOTANICAL DATA

AARON LUTTMAN<sup>†§</sup> AND JOHN BARDSLEY<sup>‡§</sup>

**Abstract.** In order to engage in photosynthesis, plant leaves absorb  $\text{CO}_2$  via the opening of pores in their surfaces called *stomata*. Water evaporates through open stomata, however, which is a detriment to plant function. Thus a plant will seek a stomatal aperture that balances its need for  $\text{CO}_2$  with its aversion to  $\text{H}_2\text{O}$  loss. In order to visualize a particular leaf's stomatal aperture, an experimentalist injects the leaf with dye so that it fluoresces. Regions with a higher relative intensity that do not correspond to veins in the leaf then correspond to areas in which the stomata are closed and the darker non-vein regions to areas in which the stomata are open. A camera is used to collect the emitted light, and a fluorescence pattern is measured. Images are continually recorded as these patterns change with time, resulting in a video sequence. The goal in this paper is to segment one such video sequence into fluorescing and non-fluorescing regions. After preprocessing the video, we take a variational approach to the segmentation problem and solve the associated three-dimensional evolution equation using a semi-implicit numerical scheme. A fixed-point formulation of the method is presented as are results of the segmentation for actual leaf data.

**Key words.** Image Segmentation, Shape Reconstruction, Variational Methods

**1. Introduction.** A plant's leaves have pores on their surfaces called *stomata* that occur at densities of tens to hundreds per  $\text{mm}^2$  [28]. Carbon dioxide ( $\text{CO}_2$ ), which is used in photosynthetic energy storage, is absorbed when the stomata are open, and, in general, the more open the pores are, the more  $\text{CO}_2$  is absorbed. When the stomata are opened, however, water is lost through evaporation, which is detrimental to plant function. Thus, in response to external conditions, the stomatal apertures in a leaf must be regulated in such a way that the plant's need to absorb  $\text{CO}_2$  is balanced with its aversion to losing  $\text{H}_2\text{O}$ . Peak, et. al. [28] make the following statement regarding this process: "A central paradigm of plant biology is that, in the face of spatially heterogeneous and temporally varying environmental conditions, a plant continually adjusts its stomatal aperture so that, over time, it maximizes  $\text{CO}_2$  uptake for a fixed amount of water loss." Moreover, it is thought that this optimization occurs over the entire leaf rather than for each stoma individually. It is not yet understood how plants solve this global optimization problem, but it has been hypothesized (cf. [28]) that the stomatal aperture depends on external environmental conditions as well as on interactions between neighboring stomata. The latter claim is somewhat controversial among plant biologists, and its primary motivation is the observation - made in over 200 plant species - that, even under spatially varying environmental conditions, stomatal apertures are often synchronized into spatially extended patches. This behavior is observed even though patchiness often results in less than optimal local  $\text{CO}_2$  uptake for fixed water loss. At this point it is not known how, or even if, neighboring stomata communicate, but it is believed that a study of stomatal patch dynamics in particular leaves may provide insight into this phenomena.

In order to study stomatal patch dynamics, this phenomenon must be observed in a particular leaf. The data analyzed in this paper was collected by the *Complexity and Stomatal Behavior* research lab at Utah State University [28]. A cocklebur (*Xanthium*

---

<sup>†</sup>Department of Mathematical Sciences, University of Montana; Division of Science and Mathematics, Bethany Lutheran College, Mankato, MN. Email: luttman@blc.edu

<sup>‡</sup>Department of Mathematical Sciences, University of Montana. Email: bardsleyj@mso.umt.edu

<sup>§</sup>This work was partially supported by the NSF under grant DMS-0504325 and by Montana NSF EPSCoR.

*strumarium L.*) leaf is placed in a light-penetrating chamber so that the temperature is controlled and gasses can be independently regulated and measured both on the top and bottom surfaces of the leaf. A dye is injected into the leaf that fluoresces when the stomata close, and stomatal dynamics are initiated by decreasing the concentration of  $\text{H}_2\text{O}$  on the upper surface of the leaf. This causes the stomata to close, since a decrease in environmental humidity causes an increase in  $\text{H}_2\text{O}$  loss through evaporation. This results in increased fluorescence, which is measured by a charge-coupled device (CCD) camera. Each image captured by the camera consists of a  $512 \times 512$  array of pixels. Regions of high relative intensity correspond to areas in which the stomata are closed, whereas regions of low relative intensity correspond to areas in which the stomata are open. The images in the sequence were collected every 40 seconds over a period of several hours. (For more details on the data collection procedure and chlorophyll fluorescence, see [14, 25, 28].) Four sample images from the video sequence analyzed in this paper are shown in Figure 1.1. In frames 150, 275 and 400, the light (closed stomata) and dark (open stomata) patches are evident. Our task in this paper is to provide a binary segmentation of the video into these light and dark regions. From such a segmentation, a study of patch dynamics can then be performed.

The first step in processing the data is denoising each of the images in the video sequence. For this we use the PDE-based denoising method of Rudin-Osher-Fatemi [30], with  $L^1$ -fidelity as proposed by Chan and Esedoglu in [8]. We choose this method because it is effective at denoising an image without degrading its inherent structure. It is necessary to use such a method so that resolution is not lost in the images, particularly at the boundaries of the stomatal patches and veins, which correspond to the darkest regions in the images of Figure 1.1. Additional noise enters the problem in the form of global lighting changes that occur during the data collection experiment, so the data must also be scaled and normalized against such effects.

Having preprocessed the video data, our final task is to extract the stomatal patches from the video sequence by segmenting the video into the fluorescing and non-fluorescing regions. To do this, we take a variational approach inspired by the methods found in [10, 11, 18, 19]. In contrast to these methods, our segmentation is computed in three-dimensions (over the entire video sequence) and segments based on contrast to an *a priori* background model generated from the experiment. We perform the segmentation in three-dimensions, because the algorithm is then able to globally compare the changes between the video sequence and the background model. When each frame is considered individually, no such comparison is possible, and, as a result, differences between the images and the background model tend to be overemphasized in frames with little or no dynamics. Such frames occur in the early stages of the video analyzed here. Other standard segmentation approaches include threshold-based methods [31], edge-based methods [3, 4, 17], region-based methods [16, 27], and hybrid techniques [35, 39] that combine several approaches. The reason that such approaches fail for our problem is that vein and non-vein regions fluoresce differently. In particular, the standard methods tend to either ignore the veins, by incorporating them in a background model which is subtracted off, or over-emphasize the veins by applying the same segmentation standard to the vein pixels as the non-vein pixels. The algorithm used here incorporates both the measured data and a static background model into the segmentation, so that the veins are neither ignored nor over-emphasized.

The video preprocessing is discussed in Section 2. In Section 3, we present the variational segmentation method. The numerical implementation of the method is

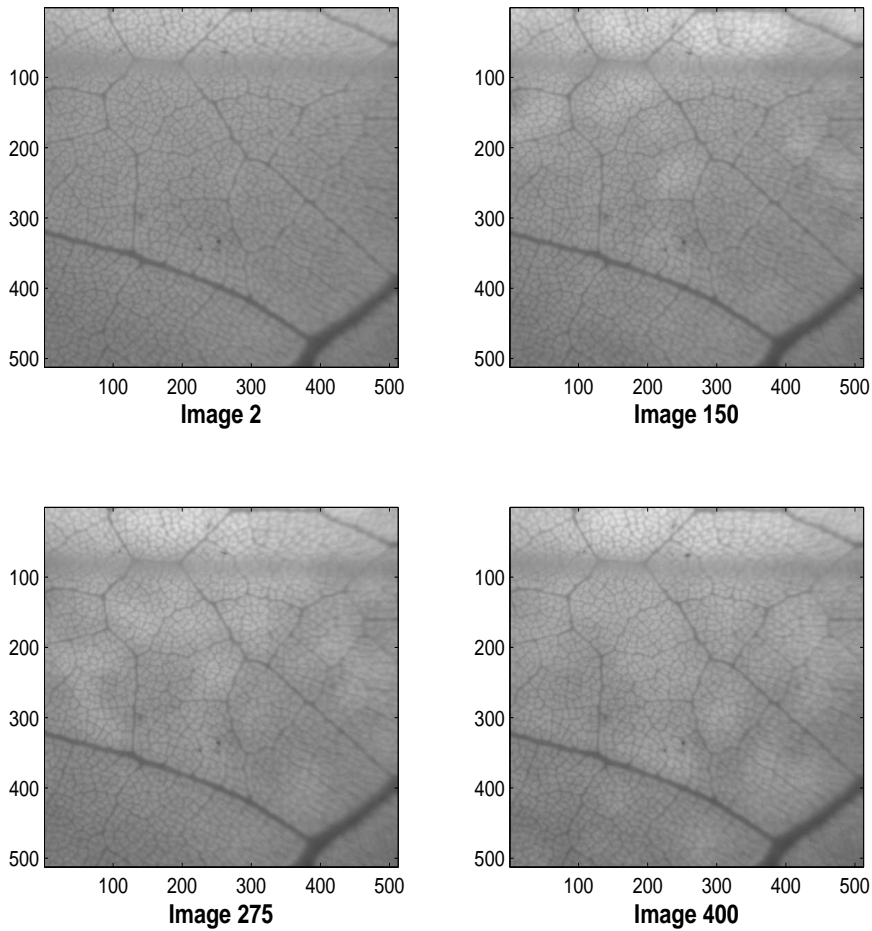


FIG. 1.1. *Four 512x512 Images from the Video Sequence of 436 Frames*

detailed, and we present our results in Section 4, and conclusions are given in Section 5.

**2. Data Preprocessing.** Before segmentation, it is necessary to remove noise from each frame of the video, and two such types of noise will be accounted for. The first is the noise incurred during image formation due to the CCD camera instrumentation. This noise is high frequency and will be removed using the method discussed in Section 2.1. In Section 2.2 we discuss removal of lower frequency noise due to global lighting changes that occurred during the data collection experiment.

**2.1. Denoising.** The images in the video sequence of interest are formed via the counting of photons by a mechanism in each pixel of the camera's CCD array. The imprecision of this counting process together with stochastic instrumental errors results in collected images that are noisy. (For more details on CCD camera noise

statistics see [33].) Instrumentation noise is high frequency and can be removed using a wide array of denoising methods. Due to the fact that the images in the video have edges at the vein boundaries, the method chosen must preserve edges. Several methods of this type have recently been developed [6, 12, 29], and one of the most popular choices is that of Rudin, Osher and Fatemi (ROF) [30], which seeks to minimize the sum of the total variation of the digital signal and an  $L^2$  data-fidelity term. The popularity of ROF is due to the fact that the approach is effective, and computationally efficient algorithms exist for its solution [7, 9, 36]. It has been noticed, however, that if an  $L^1$ -fidelity is used instead, as is done in [8], less degradation occurs in the structure of the denoised images (see [2] and the references therein). Moreover, the  $L^1$  denoising technique is completely contrast invariant. Since our images are highly structured at multiple contrast scales, we take this approach, minimizing the total variation regularized  $L^1$ -fidelity functional using the gradient descent method of [8]. Although we are able to obtain satisfactory results in a computationally feasible manner using this algorithm, it was pointed out to us by the referees that the numerical approach of [8] is typically very slow to converge, and that a faster method is presented [2].

Let  $I_0(\vec{x})$ , where  $\vec{x} = (x, y)$ , be the input image corresponding to a single frame of the video sequence. Then the denoising corresponds to computing a minimum of the energy functional

$$E(I) = \int_{\Omega} |\nabla I| \, d\vec{x} + \lambda \int_{\Omega} |I_0 - I| \, d\vec{x}, \quad (2.1)$$

where  $\lambda > 0$  is the regularization parameter and  $\Omega \subset \mathbb{R}^2$  is the image domain. The first term on the right-hand side of (2.1) is the *total variation* of  $I$  which acts as a local smoothing operator, and the second term is the fidelity term which ensures that the resulting  $I$  approximates the input image  $I_0$  in  $L^1$ . The minimization cannot be performed directly due to numerical instability, so a differentiable approximation of the functional, given by

$$E(I) = \int_{\Omega} \sqrt{(\nabla I)^2 + \delta} \, d\vec{x} + \lambda \int_{\Omega} \sqrt{(I_0 - I)^2 + \epsilon} \, d\vec{x}, \quad (2.2)$$

with parameters  $\delta$  and  $\epsilon$  small and positive, is minimized instead (see [11]). The minimization procedure is performed using the gradient descent method for (2.2), which involves evolving the image  $I(\vec{x}; t)$  according to

$$I_t = \operatorname{div} \left( \frac{\nabla I}{|\nabla I| + \delta} \right) + \lambda \frac{I_0 - I}{\left( (I_0 - I)^2 + \epsilon \right)^{1/2}}, \quad I_0(\vec{x}) = I(\vec{x}, 0), \quad (2.3)$$

where  $t$  is an artificial time parameter. This is done for each frame in the video sequence.

Total-variation regularization is used for image denoising when it is assumed that the image has jump discontinuities. The total-variation penalization strengthens jump discontinuities, which, for our data, means that the veins are enhanced. Enhancing the veins is good, because that will improve our background model for vein pixels and also make the edges along the vein boundaries sharper. Jump discontinuities in our data along the time dimension are experimental artifacts caused by global lighting changes between frames (recall that the frames are generated every 40 seconds). These artifacts must be removed, so applying the denoising algorithm to the data in 3D

would not be appropriate, as this would enhance these experimental artifacts. This is the motivation behind denoising each image in the video sequence individually. Our method for removing these global lighting artifacts is discussed in the next section.

**2.2. Median-Change Normalization.** After denoising, the video data must be normalized against global lighting effects. The leaf experiments run over several hours, and there are large-scale, global lighting changes that must be accounted for. We make the following assumption, which is reasonable for our data: the number of pixels that have increasing intensity due to fluorescence dynamics at any given time is less than half of the total number of pixels, and likewise less than half of all pixels demonstrate a decrease in intensity due to dynamics. Given that this assumption is correct, the median intensity changes from one frame to the next are primarily due to global effects. Thus we normalize each frame in the video sequence by subtracting off the median intensity change. If  $I(\vec{x}, n)$  represents the  $n^{\text{th}}$  image in the video sequence, then we replace  $I(\vec{x}, n)$  by  $I^*(\vec{x}, n)$  where

$$I^*(\vec{x}, n) = I(\vec{x}, n) - \underset{\vec{x}}{\text{median}}(I(\vec{x}, n) - I^*(\vec{x}, n - 1)),$$

with  $I^*(\vec{x}, 1) = I(\vec{x}, 1)$ . Note that this is not equivalent to simply normalizing each image to have the same median value. Lastly the video sequence is linearly scaled so that the intensity values in the sequence take up the full range  $[0, 255]$ .

The preprocessed images corresponding to the original images in Figure 1.1 are given in Figure 2.1. In these images, the fluorescing stomatal patches are much clearer. Furthermore, it is evident that the patches on the leaf change in time, as a large portion of the image in frame 400 is quite bright whereas essentially all of frame 2 is much darker. The goal is to extract the bright regions from each frame in order to determine the evolution of the fluorescence patterns over time. Our method for this segmentation step is presented in the next section.

**3. Variational Segmentation of the Leaf Data.** Let  $\vec{x} = (x, y, z)$  be the coordinates of the video sequence, with  $x$  and  $y$  denoting pixel coordinates within each frame and  $z$  corresponding to frame number. We define  $I(\vec{x})$  to be the preprocessed video sequence, and let  $D(\vec{x}) = |I(\vec{x}) - B(x, y)|$ , where  $B$  is a static background model generated using a sequence of frames from the beginning of the video. The value of  $B(x, y)$  at pixel  $(x, y)$  is taken to be the mean intensity of  $I(x, y, z)$  for values of  $z$  corresponding to these early frames (the first 40 frames are used for the data analyzed here). The early frames in this video sequence contain no noticeable dynamics, so the background model contains no bright regions. Thus any bright regions that occur in later frames are in contrast to the background model, except at the veins, which do not fluoresce and remain dark even when they are within a bright region.

We define  $\Omega \subset \mathbb{R}^3$  to be the (open) domain of  $D$  and characterize all possible segmentations of the video sequence by piecewise-constant functions  $u$  on  $\Omega$  that take on two distinct values,  $c_1$  and  $c_2$ , such that the sets  $\Omega_1 = \{\vec{x} \in \Omega \mid u(\vec{x}) = c_1\}$  and  $\Omega_2 = \{\vec{x} \in \Omega \mid u(\vec{x}) = c_2\}$  have piecewise-smooth boundaries of finite surface area. From any such  $u$ , a natural segmentation of the video results; namely,  $\Omega_1$  corresponds to the bright regions and  $\Omega_2$  to the dark regions of the image, or vice versa.

**3.1. The Segmentation Functional.** We assume that the optimal segmentation of the video is given by the function  $u$ , together with values  $c_1$  and  $c_2$ , that

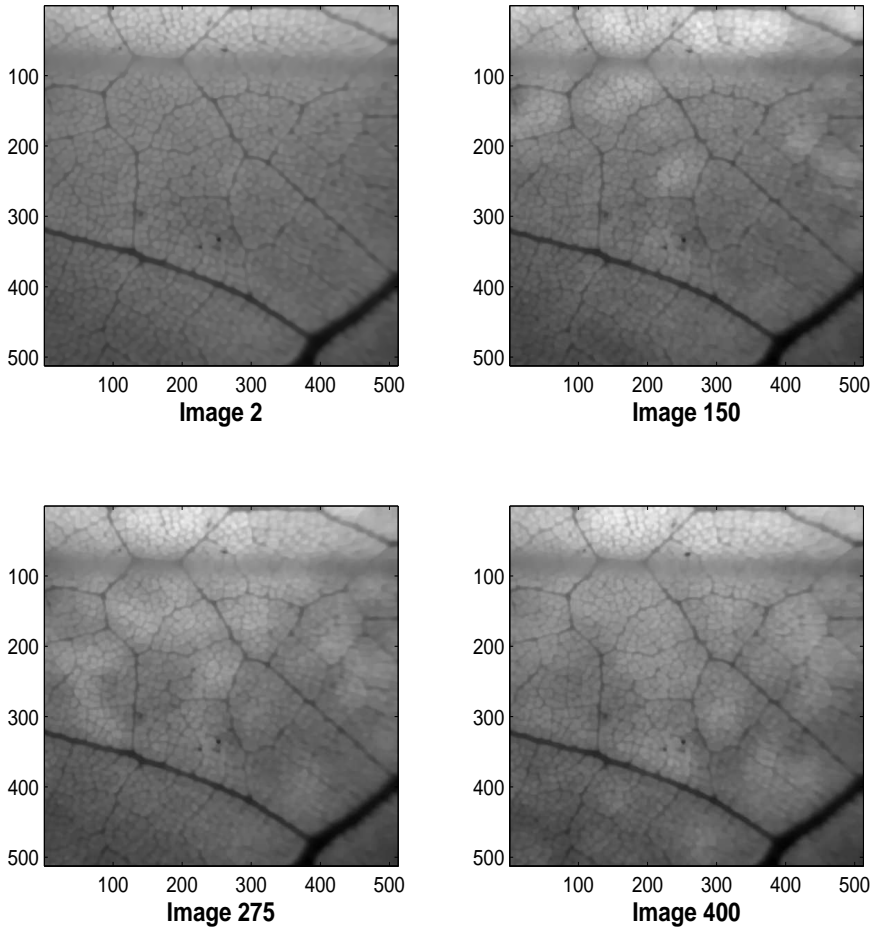


FIG. 2.1. *Images 2, 150, 275, and 400 After Preprocessing*

minimize the squared  $L^2$  norm of the difference between  $D$  and  $u$ , i.e.

$$E(u, c_1, c_2) = \|D(\vec{x}) - u(\vec{x})\|_{L^2(\Omega)}^2 = \int_{\Omega_1} (D(\vec{x}) - c_1)^2 d\vec{x} + \int_{\Omega_2} (D(\vec{x}) - c_2)^2 d\vec{x}. \quad (3.1)$$

Following Kimmel in [18, 19], we reduce  $E$  by computing the optimal values of  $c_1$  and  $c_2$  for fixed  $\Omega_1$  and  $\Omega_2$ . Assuming that  $u$  and  $c_2$  are fixed, the minimizer of  $E(u, c_1, c_2)$  occurs where its derivative with respect to  $c_1$  is zero. Since

$$\begin{aligned} E(u, c_1, c_2) &= \int_{\Omega_1} (D(\vec{x}) - c_1)^2 d\vec{x} + \int_{\Omega_2} (D(\vec{x}) - c_2)^2 d\vec{x}, \\ &= \int_{\Omega_1} D(\vec{x}) d\vec{x} - 2c_1 \int_{\Omega_1} D(\vec{x}) d\vec{x} + c_1^2 \int_{\Omega_1} d\vec{x} + \int_{\Omega_2} (D(\vec{x}) - c_2)^2 d\vec{x}, \end{aligned}$$

and  $u$  and  $c_2$  are assumed fixed, it is readily seen that the derivative of  $E$  with respect to  $c_1$  is zero when

$$c_1 = \int_{\Omega_1} D(\vec{x}) d\vec{x} \Big/ \int_{\Omega_1} d\vec{x}, \quad (3.2)$$

which is the mean value of  $D$  on  $\Omega_1$ . Note that  $E$  is bounded below with respect to  $c_1$  but not from above. Thus, since (3.2) corresponds to the only local extremum of  $E$ , it must be a global minimum with respect to  $c_1$  for fixed  $c_2$  and  $u$ . A similar argument shows that, for fixed  $c_1$  and  $u$ , the minimizing value of  $c_2$  is the mean value of  $D$  on  $\Omega_2$ ,

$$c_2 = \int_{\Omega_2} D(\vec{x}) d\vec{x} \Big/ \int_{\Omega_2} d\vec{x}. \quad (3.3)$$

In the sequel, we will use  $c_1$  to denote (3.2), and  $c_2$  to denote (3.3). The resulting functional  $E$  then depends explicitly only on  $u$ .

Computing a minimizer of  $E$  analytically on the set of piecewise-constant functions  $u$  taking on exactly two values is not possible, and hence, a numerical approach must be taken. This can be done efficiently if we recast the minimization problem as an evolution equation.

### 3.2. Formulating the Segmentation Problem as an Evolution Equation.

For each function  $u$  of the type discussed above, there exists a Lipschitz continuous level set function  $\Phi: \Omega \rightarrow \mathbb{R}$  satisfying

$$\Omega_1 = \{\vec{x} \in \Omega \mid u(\vec{x}) = c_1\} = \{\vec{x} \in \Omega \mid \Phi(\vec{x}) < 0\}, \quad (3.4)$$

$$\Omega_2 = \{\vec{x} \in \Omega \mid u(\vec{x}) = c_2\} = \{\vec{x} \in \Omega \mid \Phi(\vec{x}) > 0\}, \quad (3.5)$$

where set equality is always assumed to be up to a set of Lebesgue measure zero. Such a function  $\Phi$  is called a level set representation for  $u$ , and, though level set representations of  $u$  are not unique, each particular  $\Phi$  yields a unique segmentation of the video sequence via (3.4) and (3.5). Then (3.1) can be rewritten as

$$E(\Phi) = \int_{\Omega} |D - c_1|^2 (1 - H(\Phi)) + |D - c_2|^2 H(\Phi) d\vec{x}, \quad (3.6)$$

where  $H$  is the Heaviside function, i.e.  $H(\Phi)$  is 1 for  $\Phi > 0$  and is 0 for  $\Phi \leq 0$ . This is the model proposed in [11], and it is based on the variational segmentation algorithm first developed by Mumford and Shah [26].

In order to obtain efficient numerical methods for minimizing (3.6), we seek to recast this problem as an evolution equation of the form

$$\frac{\partial \Phi}{\partial t} = F(\Phi), \quad (3.7)$$

with initial condition  $\Phi(\vec{x}; 0) = \Phi_0(\vec{x})$ . The function  $F$  dictates the deformation of  $\Phi$  in time and is chosen so that the steady-state solution of (3.7) corresponds to a solution of the Euler-Lagrange equation

$$\frac{\delta E}{\delta \Phi} = 0, \quad (3.8)$$

where  $\delta/\delta\Phi$  denotes the variational derivative. Recalling that the distributional derivative of  $H(\Phi)$  is the Dirac delta distribution  $\delta(\Phi)$ , a straightforward calculation yields

$$\frac{\delta E(\Phi)}{\delta\Phi} = -2 \left( (c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right) \right) \delta(\Phi). \quad (3.9)$$

The standard approach taken in the literature is to approximately solve (3.8) by numerically solving the gradient descent evolution equation

$$\frac{\partial\Phi}{\partial t} = (c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right) \delta(\Phi). \quad (3.10)$$

However, we note that (3.8)-(3.9) is satisfied if

$$(c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right) = 0, \quad (3.11)$$

which leads, instead, to the simplified evolution equation

$$\frac{\partial\Phi}{\partial t} = (c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right). \quad (3.12)$$

To see that (3.12) is a minimization scheme for (3.6), we note that  $E(\Phi(\vec{x}; t))$  is a non-increasing function in  $t$  provided the variational derivative of  $E$  with respect to  $t$ , given by

$$\frac{\delta E(\Phi(\vec{x}; t))}{\delta t} = \frac{\delta E(\Phi(\vec{x}; t))}{\delta\Phi} \frac{\partial\Phi(\vec{x}; t)}{\partial t},$$

is non-positive. A straightforward calculation shows that this is satisfied by both (3.10) and (3.12).

We note that in [11] and related works, approximate solutions of (3.10) are sought, requiring an approximation of the  $\delta(\Phi)$  term. In [40], for example, a computationally feasible problem is obtained by replacing  $\delta(\Phi)$  with  $|\nabla\Phi|$ ; whereas in [1, 11, 20],  $\delta(\Phi)$  is approximated by a piecewise smooth function. Our own numerical experiments suggest, however, that although these approaches can be effective, adequate segmentations and dramatically less computationally intensive algorithms result if (3.12) is used instead. We also note the fact that with the ‘‘approximation’’  $\delta(\Phi) = 1$ , (3.10) becomes (3.12).

**3.3. Regularizing the Segmentation Algorithm.** There is no guarantee that a solution  $\Phi$  of (3.12) will yield a segmentation (3.4)-(3.5) in which the boundary of  $\Omega_1$  is piecewise-smooth and of finite surface area. To ensure that our solutions have this property, a regularization functional is added to the segmentation functional (3.6). Following Kimmel [18], we begin with the geodesic active contour [5] functional given by

$$E_{\text{reg}}(\Phi) = \oint_{\partial\Omega_1} g \, dS,$$

where  $\partial\Omega_1$  denotes the boundary of  $\Omega_1$  and  $g$  is an edge-indicator function for each image in the video sequence. This corresponds to the regularization used in [11] if  $g \equiv 1$ . We use, instead, the standard edge indicator function,

$$g(x, y) = \frac{1}{1 + |\nabla I(x, y)|^2}, \quad (3.13)$$

evaluated in each image individually. Note that edges are two-dimensional features, a fact that requires  $g$  be evaluated in 2D rather than 3D. Moreover, this particular regularization allows us to incorporate edge information directly into the segmentation functional. Using  $g = 1$ , as is done in [11], does not allow edges of different strengths to be weighted differently. All edges are either enhanced or flattened according to the value of the regularization parameter. Using the edge-indicator function in (3.13), however, allows the segmentation functional to treat edges of different strengths differently, according to the weighting factor determined by  $g$ . Thus some edges will be enhanced greatly, some a little bit, some not at all, and some will be flattened. Since determining the impact of the leaf veins on stomatal dynamics is of particular interest, and edges in the data correspond to vein boundaries, this yields a measurement, based on vein size, of the effect of the veins on the segmentation. Moreover, note that the fidelity functional acts on the background-subtracted data but that  $g$  acts on the preprocessed data directly. Thus the regularization allows us to include this vein information directly into the segmentation algorithm, where other segmentation methods that act on background-subtracted data do not.

In order to incorporate the regularization functional into the level set PDE (3.12), it is necessary to compute its variational derivative with respect to  $\Phi$ , which is (see [5])

$$\frac{\delta E_{\text{reg}}(\Phi)}{\delta \Phi} = -\text{div} \left( g \frac{\nabla \Phi}{|\nabla \Phi|} \right) |\nabla \Phi|. \quad (3.14)$$

In order to optimize computational efficiency (recall that we must numerically solve the evolution equation in three-dimensions), we seek an approximation of (3.14) that does not require an evaluation of  $|\nabla \Phi|$ . Such an approximation can be obtained by appealing to the equality

$$\text{div} \left( g \frac{\nabla \Phi}{|\nabla \Phi|} \right) |\nabla \Phi| = \text{div}(g \nabla \Phi) - \frac{g}{|\nabla \Phi|} \nabla(|\nabla \Phi|).$$

In particular, rather than using the full geodesic active contour model for regularization, we use only the diffusion term  $\text{div}(g \nabla \Phi)$ . This then yields the regularized analogue to (3.11), given by the nonlinear equation

$$\alpha \text{div}(g \nabla \Phi) + \beta(c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right) = 0 \quad (3.15)$$

and corresponding gradient descent evolution equation

$$\frac{\partial \Phi}{\partial t} = \alpha \text{div}(g \nabla \Phi) + \beta(c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right), \quad (3.16)$$

where  $\alpha$  and  $\beta$  are weighting factors. Note that (3.16) has the form of (3.7), as desired.

**4. Numerical Methods and Results.** The most direct method of solving equation (3.16) is to use an explicit time discretization. This is not a practical approach for our problem, however, since the corresponding stability criterion is computationally prohibitive. Thus, rather than discretizing (3.16) explicitly, we follow Weickert, et. al. [37, 38] and use the *semi-implicit* time discretization

$$\Phi^{n+1} - \alpha \Delta t \text{div}(g \nabla \Phi^{n+1}) = \Phi^n + \Delta t \beta (c_2^n - c_1^n) \left( D - \frac{1}{2}(c_1^n + c_2^n) \right) \quad (4.1)$$

where  $c_1^n$  and  $c_2^n$  represent the values of  $c_1$  and  $c_2$  at the  $n^{\text{th}}$  time step. Recall that  $c_1$  and  $c_2$  depend on  $\Phi$ . This numerical scheme is called semi-implicit, or linear-implicit, because the regularization term is evaluated at the  $n+1$  time step, whereas the fidelity term is evaluated at the  $n^{\text{th}}$  time step. Scheme (4.1) can be rewritten as

$$(\mathcal{I} - \alpha\Delta t(A_x + A_y + A_z))\Phi^{n+1} = \Phi^n + \Delta t\beta(c_2^n - c_1^n)\left(D - \frac{1}{2}(c_1^n + c_2^n)\right), \quad (4.2)$$

where  $A_s = \frac{\partial}{\partial s}g\frac{\partial}{\partial s}$  for  $s \in \{x, y, z\}$  and  $\mathcal{I}$  is the identity operator. It is shown in [38] that this discretization leads to a convergent, unconditionally stable numerical scheme.

The unconditional stability of (4.2) results in a relaxed restriction on the size of the time step, which is good for computational efficiency. The drawback is that a large linear system must be solved at every iteration. There are various methods for solving large linear systems of this type efficiently such as multigrid, multiplicative operator splitting (e.g., ADI or a Locally One Dimensional scheme such as in [18, 19]), and Additive Operator Splitting (AOS) [37, 38]. We use AOS, where the linear system (4.2) is approximately solved at each iteration using

$$\begin{aligned} & (\mathcal{I} - \alpha\Delta t(A_x + A_y + A_z))^{-1} \\ & \approx \frac{1}{3}(\mathcal{I} - 3\alpha\Delta tA_x)^{-1} + \frac{1}{3}(\mathcal{I} - 3\alpha\Delta tA_y)^{-1} + \frac{1}{3}(\mathcal{I} - 3\alpha\Delta tA_z)^{-1}. \end{aligned} \quad (4.3)$$

This operator splitting yields an  $\mathcal{O}(\Delta t^2)$  approximation to the semi-implicit scheme as can be seen from the Taylor series expansions

$$(\mathcal{I} - \alpha\Delta t(A_x + A_y + A_z))^{-1}v = v + \alpha\Delta t(A_x + A_y + A_z)v + \mathcal{O}(\Delta t^2),$$

and

$$\begin{aligned} & \frac{1}{3}(\mathcal{I} - 3\alpha\Delta tA_x)^{-1}v + \frac{1}{3}(\mathcal{I} - 3\alpha\Delta tA_y)^{-1}v + \frac{1}{3}(\mathcal{I} - 3\alpha\Delta tA_z)^{-1}v \\ & = \frac{1}{3}(v + 3\alpha\Delta tA_xv) + \frac{1}{3}(v + 3\alpha\Delta tA_yv) + \frac{1}{3}(v + 3\alpha\Delta tA_zv) + \mathcal{O}(\Delta t^2) \\ & = v + \alpha\Delta t(A_x + A_y + A_z)v + \mathcal{O}(\Delta t^2). \end{aligned}$$

Thus, since (4.2) is already an  $\mathcal{O}(\Delta t^2)$  scheme, the global error does not change when AOS is used in its place.

Applying the AOS approximation to (4.2) yields

$$\Phi^{n+1} = \frac{1}{3} \sum_{s \in \{x, y, z\}} (\mathcal{I} - 3\alpha\Delta tA_s)^{-1} \left( \Phi^n + \Delta t\beta(c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right) \right). \quad (4.4)$$

Convergence and unconditional stability results for this scheme follow directly from the results found in [21, 22], which are obtained using operator semi-group theory. A straightforward proof of these facts is presented in [23], however, where the Lax-Equivalence theorem is used together with the fact that the eigenvalues of  $(\mathcal{I} - 3\alpha\Delta tA_s)^{-1}$  are between 0 and 1 for  $s \in \{x, y, z\}$ .

In order to use (4.4), we discretize the  $A_s$  operators using a standard finite difference approximation (cf. [24], Chapter 6) with spatial grid size  $h = 1$  (the natural

pixel-frame discretization). The  $A_x$  approximation is given by

$$(A_x)_{i,j} = \begin{cases} (g_{k,j,l} + g_{k,i,l})/2, & j \in \{i+1, i-1\}, \\ - \sum_{n \in \{i-1, i+1\}} (g_{k,n,l} + g_{k,i,l})/2, & j = i, \\ 0, & \text{otherwise,} \end{cases}$$

where  $k$  and  $l$  denote the row and frame number respectively.

Similarly, the  $A_y$  approximation has the form

$$(A_y)_{i,j} = \begin{cases} (g_{j,k,l} + g_{i,k,l})/2, & j \in \{i+1, i-1\}, \\ - \sum_{n \in \{i-1, i+1\}} (g_{n,k,l} + g_{i,k,l})/2, & j = i, \\ 0, & \text{otherwise,} \end{cases}$$

where  $k$  and  $l$  denote the column and frame number respectively.

Finally, the  $A_z$  approximation is given by

$$(A_z)_{i,j} = \begin{cases} (g_{k,l,j} + g_{k,l,i})/2, & j \in \{i+1, i-1\}, \\ - \sum_{n \in \{i-1, i+1\}} (g_{k,l,n} + g_{k,l,i})/2, & j = i, \\ 0, & \text{otherwise,} \end{cases}$$

where  $k$  and  $l$  denote the row and column number respectively.

The  $A_x$  matrices are generated for every row in each frame; the  $A_y$  matrices are generated for every column in each frame; and the  $A_z$  matrices are generated for each pixel time series. The associated linear systems are small since each row and column has only 512 pixels, and the time series of each pixel has length 436. Each of these linear systems can be solved independently, and therefore in parallel, using appropriate boundary conditions. We use antireflective boundary conditions [32], which guarantee a  $C^1$  extension of the signal at the boundary. Furthermore, since  $A_x$ ,  $A_y$  and  $A_z$  are tridiagonal and strictly diagonally dominant, the linear systems involving each can be solved efficiently using the well-known Thomas algorithm - a special form of Gaussian elimination for tridiagonal systems. Thus, each step of (4.4) can be computed in a computationally efficient manner.

Finally, to further increase computational efficiency, the numerical scheme is implemented at multiple scales. We use a power-of-two pyramid scheme for computing the solution. In this approach, a solution is computed using a subset of the data that is downsampled by a power of 2 in each dimension. This solution is then upsampled to a larger subset of the data and used as an initial condition for the initial-value problem at that scale. The initial condition for the initial-value problem using the full data set is therefore a linearly interpolated version of the computed solution one scale down. In this way, an initial condition  $\Phi(\vec{x}, 0)$  that well-approximates the steady-state solution of (3.16) is obtained in a computationally efficient manner.

**4.1. A Fixed-Point Formulation of the Segmentation Algorithm.** We now make note of the fact that iteration (4.2) can also be formulated as a fixed point iteration for solving the nonlinear equation (3.15). Specifically, by multiplying (3.15) by  $\Delta t$  and then adding  $\Phi$  to both sides of the equation, we see that (3.15) is equivalent to

$$(\mathcal{I} - \Delta t \alpha (\operatorname{div}(g\nabla))) \Phi = \Phi + \Delta t \beta (c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right).$$

If we evaluate the right-hand side of this equation at  $\Phi^n$ , and solve for  $\Phi$  to obtain  $\Phi^{n+1}$ , the resulting fixed-point iteration is given by (4.2).

The fixed point iteration (4.2) can also be expressed in quasi-Newton form. To see this, we note that if we define

$$F(\Phi) = \Delta t \alpha \operatorname{div}(g\nabla\Phi) + \Delta t \beta (c_2 - c_1) \left( D - \frac{1}{2}(c_1 + c_2) \right), \quad (4.5)$$

equation (4.2) can be written

$$\Phi^{n+1} = \Phi^n - MF(\Phi^n), \quad (4.6)$$

where  $M = (\Delta t \alpha (\operatorname{div}(g\nabla)) - \mathcal{I})^{-1}$ . In our numerical implementation, however, (4.4) is used, which is equivalent to (4.6) with

$$M = \frac{1}{3} \sum_{s \in \{x,y,z\}} (\mathcal{I} - 3\alpha \Delta t A_s^n)^{-1} - \mathcal{I}. \quad (4.7)$$

Equation (4.6) can be viewed as a quasi-Newton method in which the operator  $M$  in (4.7) serves as an approximation of the inverse of the Jacobian of  $F$  at  $\Phi$  in (4.5).

Formulation (4.6)-(4.7) of (4.2) allows us to make some observations regarding the convergence properties of this algorithm. First, we recall from above that (4.4) is convergent and unconditionally stable. Time-step size restrictions are therefore greatly relaxed when compared to the corresponding fully explicit formulation, which is given by (4.6) with  $M = \mathcal{I}$ . This fact, together with the quasi-Newton formulation (4.6)-(4.7) suggest that we can expect (4.4) to converge more rapidly than will the fully explicit method. Our numerical experiments support this observation, though a more thorough convergence analysis is saved for a future work.

Before continuing, we make mention of a few other fast methods for minimizing the Chan-Vese segmentation functional: these include the hybrid k-means level set algorithm of [15], which reduces the segmentation problem to the solution of an ODE in order to speed up convergence; the fast algorithm of [34], which does not require the evaluation of the gradient of the segmentation functional (3.6); and the two-stage algorithm of [13], which requires the solution of a related PDE that allows for fast solutions.

**4.2. Results.** As was noted above, the algorithm is implemented in three-dimensions. Because of this both computational and memory issues arise. In particular, although the data  $I$  and background model  $B$  can be stored as images using relatively small amounts of memory, and  $D$  can be computed directly from  $I$  and  $B$ , the level set function must be stored using floating point values at each node in the data array. For a video sequence of 436 images of  $512 \times 512$  pixels, this requires almost a gigabyte of memory when floating point values are 64-bit. In addition, the

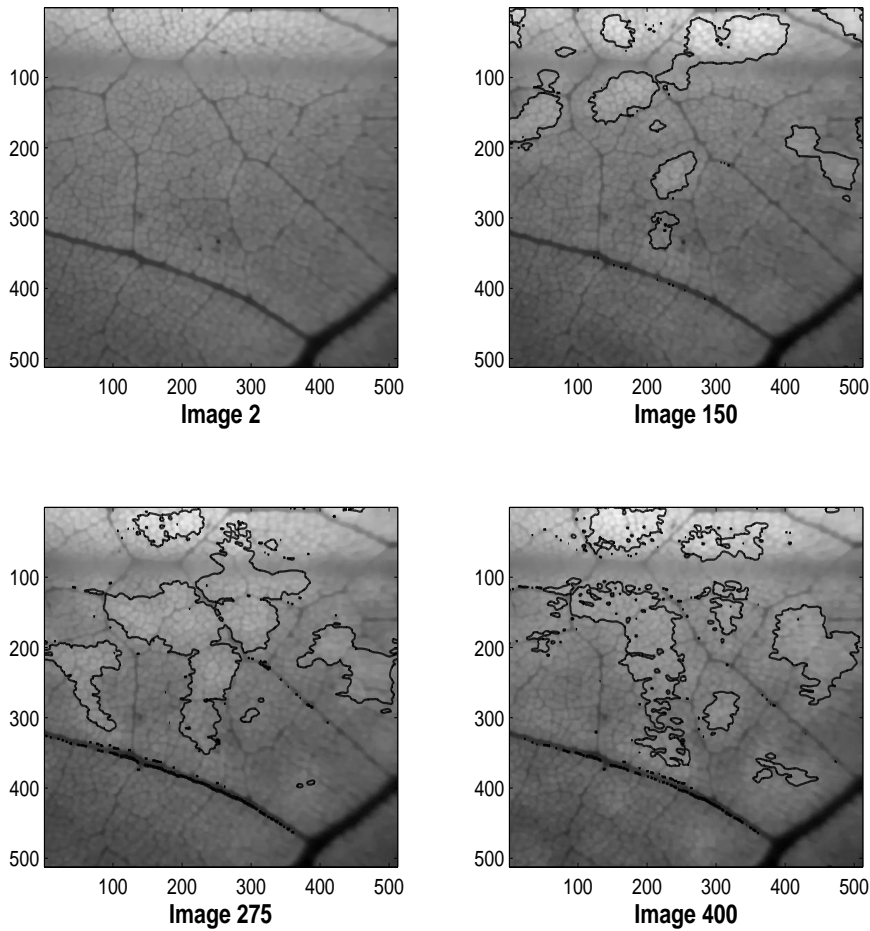


FIG. 4.1. *Segmented Images 2, 150, 275, and 400*

computation is intensive due, primarily, to the large number of linear system back-solves required at each time iteration (708608 for this particular video sequence at the highest scale).

Our algorithm is implemented on an SGI Origin 2000 with 32 processors and 32 GB of shared memory and is fully parallelized using OpenMP. Though none of the individual processors on this machine is fast by current standards (300 Mhz), the large memory capacity and parallel capabilities make the machine suitable for this application. Though each iteration at the highest scale is quite slow, few iterations are required at the highest scale, since the approximate solution computed at the lower scales is already quite good.

Since the numerical scheme is unconditionally stable with respect to  $\Delta t$ , it is possible to choose  $\Delta t$  to be quite large. For the experiments shown in this work,  $\Delta t = 0.5$ , though it is also possible to start the algorithm with a larger value of

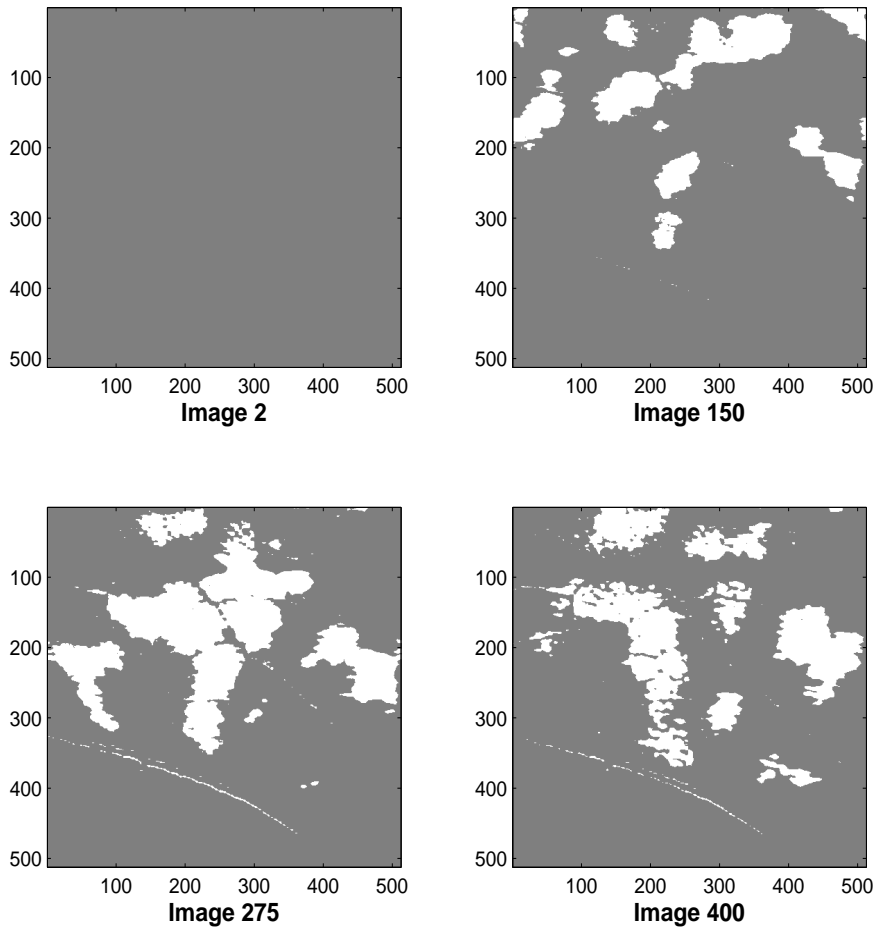


FIG. 4.2. *Binary Segmentation for Images 2, 150, 275, and 400*

$\Delta t$ , decreasing it incrementally as the solution stabilizes. The ratio of the weighting factors  $\alpha$  and  $\beta$  in (3.16) determines the amount of regularization used in the numerical implementation, and for the examples shown here,  $\alpha = 1$  and  $\beta = 2$ .

The last important detail of the numerical implementation is the initial surface from which we evolve our solution. For the examples shown here, the initial surface is a sphere contained completely inside the computational domain, and the initial level set representation of the surface,  $\Phi(\vec{x}; 0)$ , is defined to be the signed distance between  $\vec{x}$  and the sphere boundary. The distance is taken to be negative inside the sphere and positive outside the sphere.

The segmented images can be seen in Figure 4.1<sup>1</sup>. Note that there is no segmen-

<sup>1</sup>The full videos of the original, preprocessed, and segmented data can be seen at <http://hans.blc.edu/~luttman/>.

tation in image 2. This is due to the fact that there are no dynamics in the early images of the video, and it is correctly captured because the algorithm is working on the full three-dimensional video sequence. When a two-dimensional version of the segmentation algorithm is applied to each individual frame, meaningless segmentation artifacts are obtained in the early frames. This difference is due to the fact that in three-dimensions the algorithm is able to globally compare the changes between the video sequence and the background model, whereas, when each frame is considered individually, no such comparison is possible, resulting in differences between the images and the background model being overemphasized in frames with little-to-no dynamics.

The main goal in this project is to provide a binary segmentation of the video into light and dark regions, since from such a segmentation a study of two-dimensional patch dynamics can more easily be done. This is easily obtained from the results given in Figure 4.1, and can be seen in Figure 4.2. The regions of fluorescence are in white and the non-fluorescent regions are in gray.

**5. Conclusions.** The formation and dynamics of stomatal patches in plant leaves is a phenomenon that is not fully understood. In order to study this process, plant biologists collect video in which dynamically changing light and dark stomatal patches are evident. An in-depth study of the dynamics of these patches is desirable and requires a segmentation of such video into fluorescing (light) and non-fluorescing (dark) regions.

In this paper, we have presented an algorithm that provides such a segmentation for a video sequence with example frames given in Figure 1.1. The algorithm consists of several stages. First, due to experimental inconsistencies, the video in our example requires preprocessing. This is accomplished by first using a denoising algorithm for the removal of high frequency (instrumental) noise. The video is then normalized against global lighting changes, which removes low frequency (experiment specific) noise. Images from the preprocessed video are given in Figure 2.1.

The three-dimensional video segmentation is formulated as a variational inverse problem. Since the problem is ill-posed, a regularization term is added so that solutions of the corresponding minimization problem provide a sufficiently smooth segmentation of the video. The desired minimizer of the segmentation functional is computed via a time-continuous gradient descent method in the form of a three-dimensional evolution equation that is solved efficiently using a semi-implicit time discretization with additive operator splitting. A fixed-point formulation of the numerical scheme is also presented. Frames 2, 150, 275 and 400 of the segmented video are given in Figures 4.1 and 4.2.

**6. Acknowledgements.** The authors would like to thank Prof. Emily Stone of the University of Montana for introducing us to this problem and for many helpful discussions regarding the leaf data and also Profs. David Peak and Keith Mott of Utah State University for providing the leaf data and for helpful discussions regarding the current theories of stomatal patchiness. The authors would also like to thank the referees for their helpful comments and suggestions. Lastly, we acknowledge the support of the NSF under both DMS-0504325 and Montana EPSCoR.

#### REFERENCES

- [1] E. ANGELINI, R. OTSUKA, S. HOMMA, AND A. LAINE, *Comparison of ventricular geometry for two real-time 3d ultrasound machines with three-dimensional level set*, in Proceedings of

- the IEEE International Symposium on Biomedical Imaging (ISBI), vol. 1, 2004, pp. 1323–1326.
- [2] J. F. AUJOL, G. GILBOA, T. CHAN, S. OSHER, *Structure-texture image decomposition - modeling, algorithms, and parameter selection*, International Journal of Computer Vision (IJCV), Vol. 67, No. 1, pp. 111-136, 2006.
  - [3] A. BLAKE, *Introduction to active contours and visual dynamics*, in Web Reference, 1999.
  - [4] J. CANNY, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (1986).
  - [5] V. CASELLES, R. KIMMEL, AND G. SAPIRO, *On geodesic active contours*, International Journal of Computer Vision, 22 (1997), pp. 61–79.
  - [6] M. CECCARELLI, V. D. SIMONE, AND A. MURLI, *Well-posed anisotropic diffusion for image denoising*, Vision, Image and Signal Processing, IEE Proceedings-, 149 (2002), pp. 244–252.
  - [7] A. CHAMBOLLE AND J. DARBONE, *Algorithms for Total Variation Minimization*, Mathematical Models for Multi-Channel Image Processing, Beijing, 2006.7.7-8.
  - [8] T. F. CHAN AND S. ESEDOGLU, *Aspects of total variation regularized  $L^1$  function approximation*, SIAM Journal on Applied Mathematics, 65 (2005), pp. 1817–1837.
  - [9] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for TV-based image restoration*, SIAM Journal on Scientific Computing, 20 (1999), pp. 1964-1977.
  - [10] T. F. CHAN AND J. J. SHEN, *Image Processing and Analysis. Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, 2005.
  - [11] T. F. CHAN AND L. A. VESE, *Active contours without edges*, IEEE Transactions on Image Processing, 10 (2001), pp. 266–277.
  - [12] M. DESBRUN, M. MEYER, P. SCHRÖDER, AND A. H. BARR, *Anisotropic feature-preserving denoising of height fields and bivariate data*, in Graphics Interface, May 2000, pp. 145–152.
  - [13] S. ESEDOGLU AND Y.-H. TSAI., *Threshold Dynamics for the Piecewise Constant Mumford-Shah Functional*, Journal of Computational Physics, 211:1 (2006), pp. 367-384.
  - [14] B. GENTY AND S. MEYER, *Quantitative mapping of leaf photosynthesis using chlorophyll fluorescence imaging*, Australian Journal of Plant Physiology, 22 (1994), pp. 277–284.
  - [15] F. GIBOU AND R. FEDKIW, *A Fast Hybrid k-Means Level Set Algorithm for Segmentation*, 4th Annual Hawaii International Conference on Statistics and Mathematics, pp. 281-291, 2005.
  - [16] H. JIN, A. J. YEZZI, AND S. SOATTO, *Region-based segmentation on evolving surfaces with application to 3d reconstruction of shape and piecewise constant radiance*, in Proceedings of the European Conference on Computer Vision, 2004.
  - [17] M. KASS, A. WITKIN, AND D. TERZOPOULOS, *Snakes: Active contour models*, 1 (1988), pp. 321–331.
  - [18] R. KIMMEL, *Fast edge integration*, in Geometric Level Set Methods, S. Osher and N. Paragios, eds., Springer-Verlag, 2003, pp. 59–77.
  - [19] ———, *Numerical Geometry of Images: Theory, Algorithms, and Applications*, Springer, 2004.
  - [20] C. LI, C. XU, C. GUI, AND M. D. FOX, *Level set evolution without re-initialization: A new variational formulation*, in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 430–436.
  - [21] T. LÜ, P. NEITTAANMÄKI, AND X-C. TAI, *A parallel splitting up method and its application to navier-stokes equations*, Applied Mathematics Letters, 4 (1991), pp. 25–29.
  - [22] ———, *A parallel splitting up method for partial differential equations and its application to navier-stokes equations*, RAIRO Mathematical Modeling and Numerical Analysis, 26 (1992), pp. 673–708.
  - [23] A. LUTTMAN, *A Three-Dimensional Variational Approach to Video Segmentation*, PhD Thesis, University of Montana, 2006.
  - [24] K. W. MORTON AND D. F. MAYERS, *Numerical Solutions of Partial Differential Equations*, Cambridge University Press, 1994.
  - [25] K. A. MOTT, Z. G. CARDON, AND J. A. BERRY, *Asymmetric patchy stomatal closure for the two surfaces of Xanthium strumarium L. leaves at low humidity*, Plant, Cell, and Environment, 16 (1993), pp. 25–34.
  - [26] D. MUMFORD AND J. SHAH, *Optimal approximation by piecewise smooth functions and associated variational problems*, Communications in Pure and Applied Mathematics, 42 (1989), pp. 577–685.
  - [27] N. PARAGIOS AND R. DERICHE, *Coupled geodesic active regions for image segmentation: A level set approach*, in Proceedings of the European Conference on Computer Vision, vol. 2, 2000.
  - [28] D. PEAK, J. WEST, S. MESSINGER, AND K. MOTT, *Evidence for Complex, Collective Dynamics*

- and Emergent, Distributed Computation in Plants*, vol. 101, 2004.
- [29] P. PERONA AND J. MALIK, *Scale space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (1990), pp. 629–639.
  - [30] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.
  - [31] P. K. SAHOO, S. SOLTANI, AND A. K. C. WONG, *A survey of thresholding techniques*, Computer Vision, Graphics, and Image Processing, 41 (1988), pp. 233–260.
  - [32] S. SERRA-CAPIZZANO, *A note on antireflective boundary conditions and fast deblurring models*, SIAM Journal on Scientific Computing, 25 (2003), pp. 1307–1325.
  - [33] D. L. SNYDER, A. M. HAMMOND, AND R. L. WHITE, *Image recovery from data acquired with a charged-coupled-device camera*, Journal of the Optical Society of America A, 10 (1993), pp. 1014–1023.
  - [34] B. SONG AND T. F. CHAN, *A Fast Algorithm for Level Set Based Optimization*, UCLA CAM Report 02-68, December 2002.
  - [35] B. SUMENGEN, B. S. MANJUNATH, AND C. KENNEY, *Image segmentation using multi-region stability and edge strength*, in IEEE International Conference on Image Processing (ICIP), vol. 3, Sep 2003, pp. 429–432.
  - [36] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM Journal on Scientific Computing, 17 (1996), pp. 227–238.
  - [37] J. WEICKERT AND G. KÜHNE, *Fast methods for implicit active contour models*, in Geometric Level Set Methods, S. Osher and N. Paragios, eds., Springer-Verlag, 2003, pp. 43–57.
  - [38] J. WEICKERT, B. M. TER HAAR ROMENY, AND M. A. VIERGEVER, *Efficient and reliable schemes for nonlinear diffusion filtering*, IEEE Transactions on Image Processing, 7 (1998).
  - [39] Q. WU AND Y. YU, *Two-level image segmentation based on region and edge integration*, in Proceedings of the VIIth Digital Image Computing Conference: Techniques and Applications, 2003.
  - [40] H. ZHAO, T. F. CHAN, B. MERRIMAN, AND S. OSHER, *A variational level set approach to multiphase motion*, Journal of Computational Physics, 127 (1996), pp. 179–195.