A VARIATIONAL APPROACH TO VIDEO SEGMENTATION FOR BOTANICAL DATA

AARON LUTTMAN[†§] AND JOHN BARDSLEY[‡§]

Abstract. In order to engage in photosynthesis, plant leaves absorb 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 CO_2 with its aversion to H_2O loss. In order to visualize a particular leaf's stomatal aperture, an experimentalist injects the leaf with dye so that it fluoresces. The regions with a higher relative intensity then correspond to areas in which the stomata are closed and the darker regions where 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. Our task 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. The associated three-dimensional evolution equation is solved using a semi-implicit numerical scheme. Results of the segmentation for actual leaf data are presented.

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 mm^2 [20]. Carbon dioxide (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 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 CO_2 is balanced with its aversion to losing H_2O. Peak, et. al. [20] 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 CO_2 uptake for a fixed amount of water loss." Moreover, it is thought that this 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. [20]) 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 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, we need to be able to observe this phenomena in a particular leaf. The data analyzed in this paper comes from a cocklebur plant (*Xanthium strumarium L.*) and was collected by the *Complexity and Stomatal Behavior* research lab at Utah State University [20]. In order to visualize the stomata opening and closing, a chlorophyll dye is injected into the leaf. The leaf then fluoresces,

[†]Department of Mathematical Sciences, University of Montana, Email: luttman@mso.umt.edu

[‡]Department of Mathematical Sciences, University of Montana, Email: bardsleyj@mso.umt.edu

[§]This work was partially supported by the NSF under grant DMS-0504325 and by Montana NSF EPSCoR.

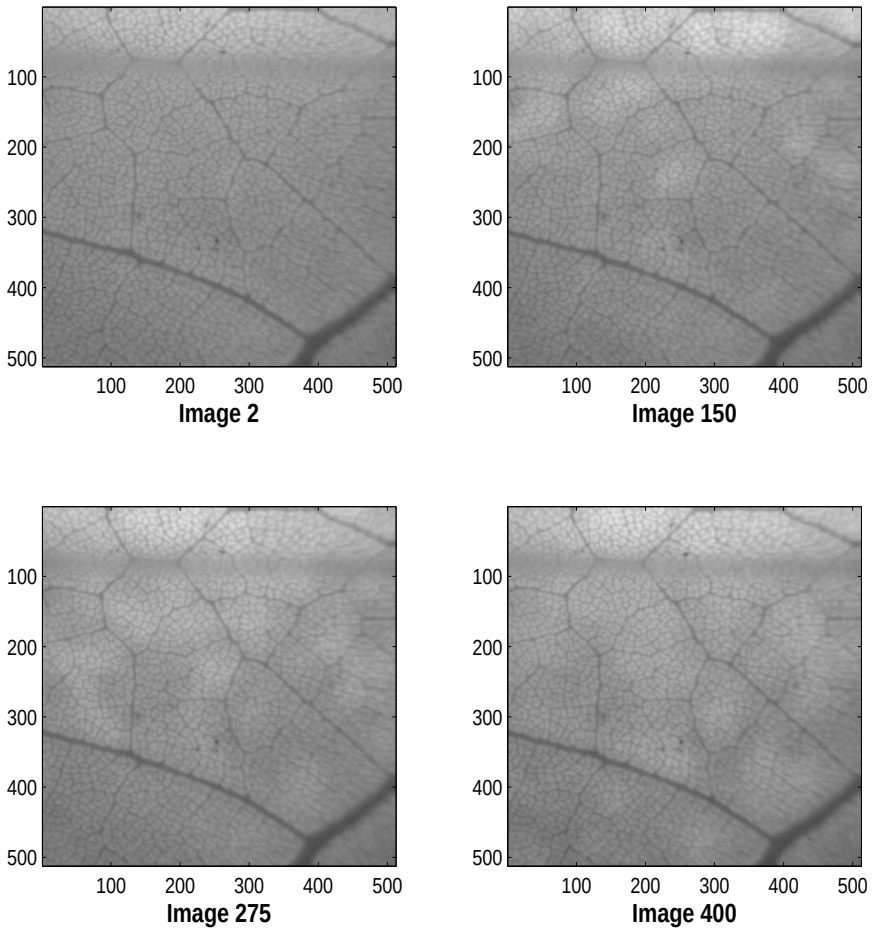

FIG. 1.1. *Four 512x512 Images from the Video Sequence of 436 Frames*

and an image is collected by a charge-coupled device (CCD) camera. Each image consists of a 512×512 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 [10, 17, 20].) Four sample images from the video sequence that we will analyze 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 more easily be done.

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 [22],

with L^1-fidelity as proposed by Chan and Esedoglu in [6]. We choose this method, because it is computationally efficient and seeks to denoise without excessive smoothing, such as would occur with a Gaussian convolution. 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 occurred 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 - and the primary thrust of this paper - 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 [7, 8, 13, 14]. In contrast to these methods, our segmentation is done 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-dimensional 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, for example, in the early stages of the video. Other standard segmentation approaches include threshold-based methods [23], edge-based methods [2, 3, 12], region-based methods [11, 19], and hybrid techniques [26, 31] that combine several approaches. The reason that standard approaches fail for our problem is that a good background model is very difficult to obtain due to the fact that the vein and non-vein regions fluoresce differently. In particular, the standard methods tend to be overly sensitive to the veins. This is not the case for our approach, which segments based on average information within each region.

The paper is organized as follows. The video preprocessing step is discussed in Section 2. In Section 3, we present the variational segmentation method. The numerical implementation of the method is detailed and we present our results in Section 4. Conclusions are given in Section 5.

2. Data Preprocessing. Before segmentation, it is necessary to remove noise from each frame of the video, and, in the preprocessing step, 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 instrumental noise results in collected images that are noisy. (For more details on CCD camera noise statistics see [25].) This 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 [5, 9, 21]. We will use the method of Rudin-Osher-Fatemi [22] with L^1-fidelity as proposed by Chan and Esedoglu in [6]. This method retains the sharp edges at the vein-boundaries and is computationally efficient for our problem. A short description of the method follows.

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

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

where $\Omega \subset \mathbb{R}^2$ is the image domain. The first term on the right-hand side of (2.1) is the smoothing term, and the second term is the fidelity term which assures 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 regularized version of the functional, given by

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

with regularization parameters δ and ϵ, is minimized instead (see [8]). The minimization procedure is performed via a gradient descent for (2.2). One such algorithm is given by evolving the image $I(\vec{x}; t)$ according to

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

where λ is a weighting factor, t is an artificial time parameter, and the input image I_0 is the initial condition $I(\vec{x}; 0)$. This is done for each frame in the video sequence.

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 example: 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, changes in median intensity will be 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^{th} image in the video sequence, then we replace $I(\vec{x}, n)$ by $I^*(\vec{x}, n)$ where

$$I^*(\vec{x}, n) = I(\vec{x}, n) - \operatorname*{median}_{\vec{x}} \left(I(\vec{x}, n) - I^*(\vec{x}, n-1)\right),$$

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 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.

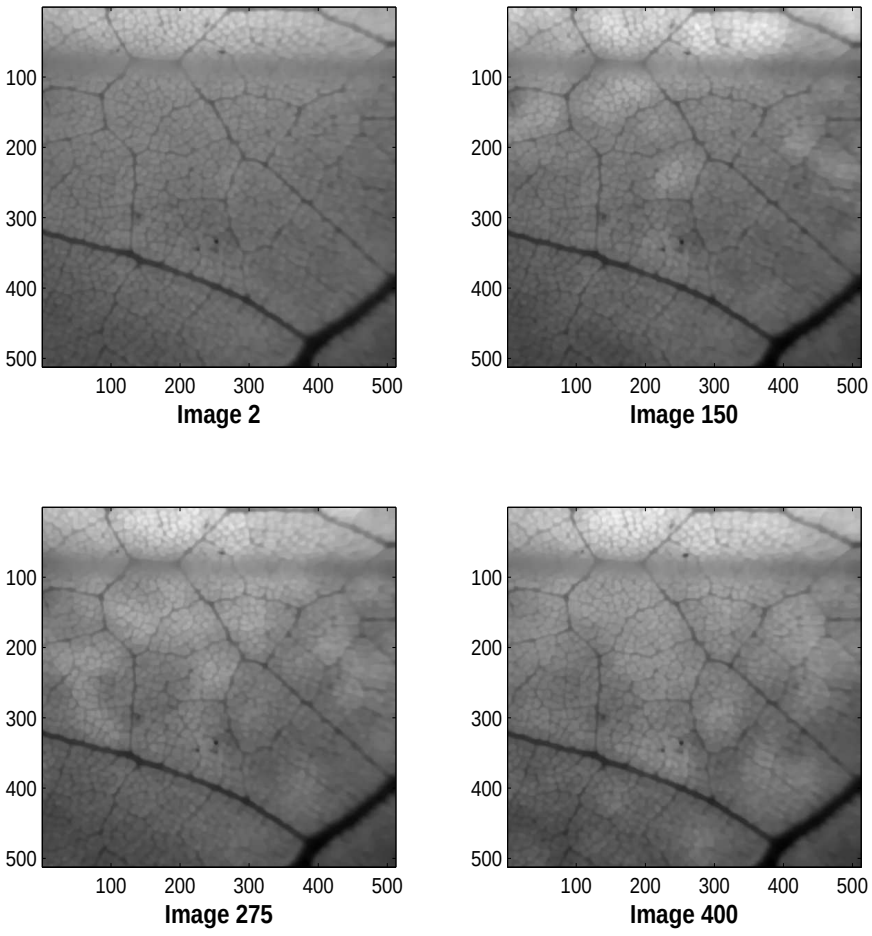

FIG. 2.1. *Images 2, 150, 275, and 400 After Preprocessing*

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. Because the early frames contain no noticeable dynamics, 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. We characterize all possible segmentations of the video sequence by piecewise-constant functions u on Ω 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, Ω_1 will correspond to the bright regions and Ω_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 minimize the functional

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

which is called the *minimum variance* functional by Kimmel in [13,14]. Following [13], we reduce E_{MV} by computing the optimal values of c_1 and c_2 for fixed Ω_1 and Ω_2. Assuming that u and c_2 are fixed, the minimizer of $E_{\mathrm{MV}}(u, c_1, c_2)$ occurs where its derivative with respect to c_1 is zero. Since

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

and u and c_2 are assumed fixed, it is readily seen that the derivative of E_{MV} 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 Ω_1. Note that E_{MV} is bounded below with respect to c_1 but not from above. Thus, since (3.2) corresponds to the only local extremum of E_{MV}, 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 Ω_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_{MV} then depends explicitly only on u.

Computing a minimizer of E_{MV} 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 $\Phi \colon \Omega \to \mathbb{R}$ 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 Φ is called a level set representation for u, and, though level set

representations of u are not unique, each particular Φ yields a unique segmentation of the video sequence via (3.4) and (3.5). Then (3.1) can be rewritten as

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

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 [8], and it is based on the variational segmentation algorithm first developed by Mumford and Shah [18].

For reasons of computational efficiency, we now formulate the problem of minimizing (3.6) as an evolution equation of the form

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

with initial condition $\Phi(\vec{x}; 0) = \Phi_0(\vec{x})$. The function F dictates the deformation of Φ in time and is chosen, at least in theory, so that the steady-state solution of (3.7) corresponds to a local minimizer of $E_{\mathrm{MV}}(\Phi)$.

It remains for us to express the problem of minimizing (3.6) as a problem of the form (3.7). This can be done in a straightforward manner. We require that $E_{\mathrm{MV}}(\Phi(\vec{x}; t))$ be a non-increasing function in t, which can be ensured if the variational derivative of E_{MV} with respect to t, given by

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

is non-positive. This is guaranteed when

$$\frac{\partial \Phi(\vec{x}; t)}{\partial t} = -\frac{\delta E_{\mathrm{MV}}(\Phi(\vec{x}; t))}{\delta \Phi}. \tag{3.8}$$

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

$$\frac{\delta E_{\mathrm{MV}}(\Phi)}{\delta \Phi} = -2 \left((c_2 - c_1) \left(D - \frac{1}{2}(c_1 + c_2) \right) \right) \delta(\Phi). \tag{3.9}$$

Any numerical method for solving (3.8)-(3.9) must have some means of dealing with the $\delta(\Phi)$ term on the right-hand side. In [32], $\delta(\Phi)$ is replaced by $|\nabla \Phi|$. This is not a computationally attractive approach for our problem. In [1, 8, 15], $\delta(\Phi)$ is approximated by a piecewise smooth function. Our own numerical experiments suggest that, although this approach is effective, adequate segmentations and dramatically less computationally intensive algorithms result if $\delta(\Phi)$ is replaced by a positive constant. Note that this allows for all of the level sets of Φ to evolve with equal weighting, rather than localizing the evolution at or near the zero level set. If $1/2$ is used, (3.8)-(3.9) can be written

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

3.3. Regularizing the Segmentation Algorithm. There is no guarantee that a solution Φ of (3.10) will yield a segmentation (3.4)-(3.5) in which the boundary of Ω_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 [13], we begin with the geodesic active contour [4] functional given by

$$E_{\mathrm{GAC}}(\Phi) = \oint_{\partial\Omega_1} g \, dS,$$

where $\partial\Omega_1$ denotes the boundary of Ω_1 and g is an edge-indicator function for each image in the video sequence. This corresponds to the regularization used in [8] if $g \equiv 1$. We use, instead, the standard edge indicator function,

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

evaluated in each image individually.

In order to incorporate the regularization functional into the level set PDE (3.10), it is necessary to compute its variational derivative with respect to Φ, which is (see [4])

$$\frac{\delta E_{\mathrm{GAC}}(\Phi)}{\delta\Phi} = -\mathrm{div}\left(g\frac{\nabla\Phi}{|\nabla\Phi|}\right)|\nabla\Phi|. \tag{3.11}$$

In order to optimize computational efficiency (recall that we must numerically solving our evolution equation in three-dimensions), we seek an approximation of (3.11) without the $|\nabla\Phi|$ terms. Such an approximation can be obtained by appealing to the equality

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

In particular, rather than using the full geodesic active contour model for regularization, we use only the first term on the right-hand side, $\mathrm{div}(g\nabla\Phi)$. This, finally, yields a regularized version of (3.10) given by

$$\frac{\partial\Phi}{\partial t} = \alpha\mathrm{div}(g\nabla\Phi) + \beta(c_2-c_1)\left(D - \frac{1}{2}(c_1+c_2)\right), \tag{3.12}$$

where α and β are weighting factors. Note that (3.12) has the form of (3.7), as desired.

4. Numerical Methods and Results. The most direct method of solving equation (3.12) is to use an explicit time discretization. This is not a practical approach for our problem, however, since the corresponding stability criterion is too restrictive. Thus, rather than discretizing (3.12) explicitly, we follow Weickert, et. al. [29,30] and use the *semi-implicit* time discretization

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

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

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

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 [30] 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 efficiently such as multigrid, multiplicative operator splitting (e.g., ADI or a Locally One Dimensional scheme such as in [13, 14]), and Additive Operator Splitting (AOS) [29,30]. We will use AOS, where the linear system (4.2) is approximately solved at each iteration using

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

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

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

and

$$\begin{aligned}&\frac{1}{3}(\mathcal{I} - 3\alpha\Delta t A_x)^{-1} v + \frac{1}{3}\left(\mathcal{I} - 3\alpha\Delta t A_y\right)^{-1} v + \frac{1}{3}\left(\mathcal{I} - 3\alpha\Delta t A_z\right)^{-1} v \\ &\quad = \frac{1}{3}\left(v + 3\alpha\Delta t A_x v\right) + \frac{1}{3}\left(v + 3\alpha\Delta t A_y v\right) + \frac{1}{3}\left(v + 3\alpha\Delta t A_z v\right) + \mathcal{O}(\Delta t^2) \\ &\quad = v + \alpha\Delta t(A_x + A_y + A_z)v + \mathcal{O}(\Delta t^2).\end{aligned}$$

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

Using the AOS approximation in (4.2) yields

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

Convergence and stability results for this scheme follow directly form the results found in [27, 28] given the fact that D is bounded on Ω.

In order to use (4.4), we must discretize the A_s operators. For this, we use a standard finite difference approximation (cf. [16], Chapter 6), with spatial grid size $h = 1$ (the natural pixel-frame discretization). The A_x approximation is given by

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

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

Similarly, the A_y approximation has the form

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

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

Finally, the A_z approximation is given by

$$(A_z)_{i,j} = \begin{cases} (g_{k,l,j} + g_{k,l,i})\left(\Phi^n_{k,l,j} - \Phi^n_{k,l,i}\right)/\,2, & j \in \{i+1, i-1\}, \\ -\sum\limits_{n \in \{i-1, i+1\}} (g_{k,l,n} + g_{k,l,i})\left(\Phi^n_{k,l,n} - \Phi^n_{k,l,i}\right)/\,2, & j = i, \\ 0, & \text{otherwise}, \end{cases}$$

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. All of these systems can be solved independently, and therefore in parallel, using appropriate boundary conditions. We use antireflective boundary conditions [24], which guarantee a C^1 extension of the signal at the boundary. Furthermore, since A_x, A_y and A_z are tridiagonal, 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 numerical scheme (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 (3.12) is obtained in a computationally efficient manner.

4.1. 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 512×512 pixels, this requires almost a gigabyte of memory when floating point values are 64-bit. In addition, the 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).

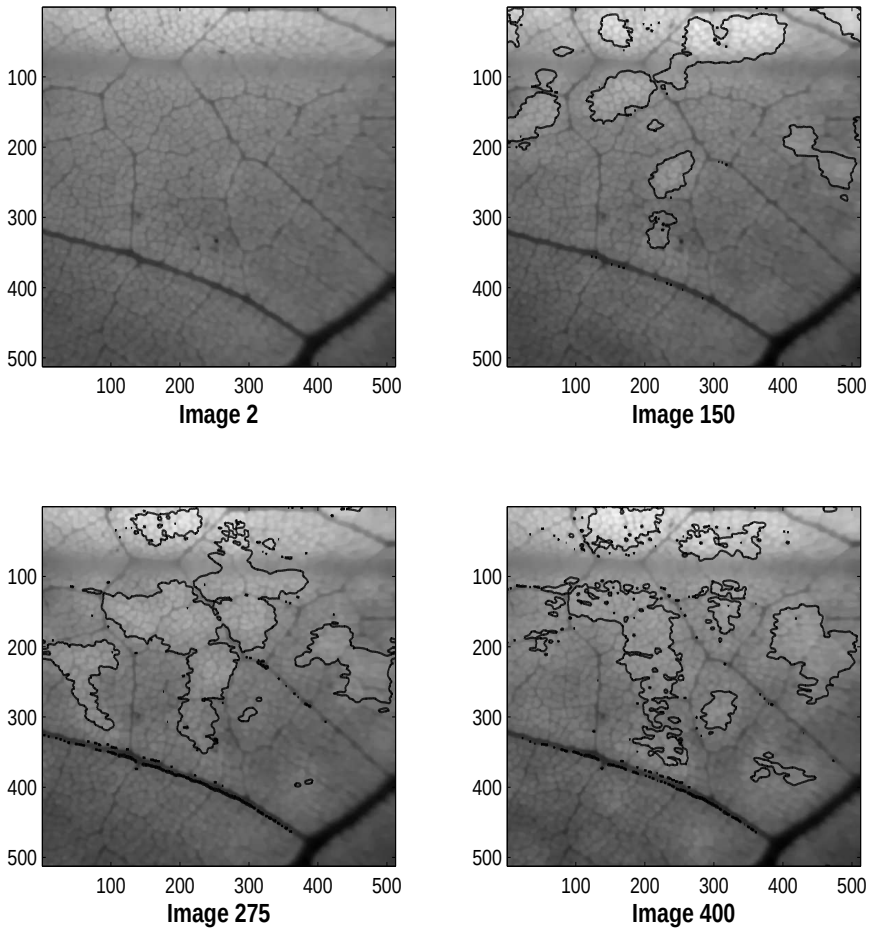

FIG. 4.1. *Segmented Images 2, 150, 275, and 400*

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 solution computed at the lower scales is already quite good.

Since the numerical scheme is unconditionally stable with respect to Δt, it is possible to choose Δt 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 Δt, decreasing it incrementally as the solution stabilizes. The ratio of the weighting factors α and β in (3.12) determines the amount of regularization used in the numerical implementation, and for the examples shown here, $\alpha = 1$ and $\beta = 2$.

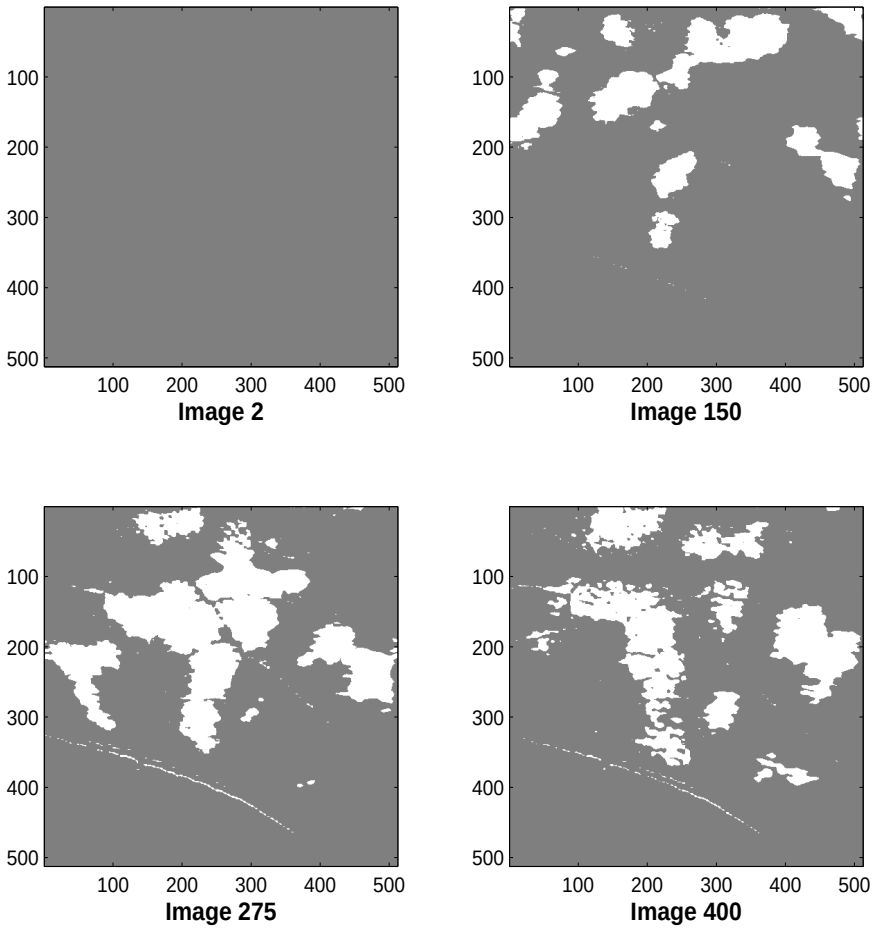

FIG. 4.2. *Binary Segmentation for Images 2, 150, 275, and 400*

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. Also, we mention that since (3.1) does have not a unique minimizer, different initial surfaces will lead to steady-state solutions of (3.12) corresponding to different local minima of E_{MV}, but the level set representation of the initial surface does not affect the evolution.

The segmented images can be seen in Figure 4.1[1]. Note that there is no segmentation in image 2. This is due to the fact that there are no dynamics in the early images

[1]The full videos of segmented data can be seen at `http://www.math.umt.edu/luttman/`.

of the video, and it is correctly captured because the algorithm is working on the full three-dimensional video sequence. We initially applied a two-dimensional version of the segmentation algorithm to each individual frame, and obtained meaningless segmentation artifacts in the early frames. This was 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. 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-to-no dynamics.

Recall that our 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 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. 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. 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] A. Blake, *Introduction to active contours and visual dynamics*, in Web Reference, 1999.

[3] J. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (1986).

[4] V. Caselles, R. Kimmel, and G. Sapiro, *On geodesic active contours*, International Journal of Computer Vision, 22 (1997), pp. 61–79.
[5] 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.
[6] 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.
[7] T. F. Chan and J. J. Shen, *Image Processing and Analysis. Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, 2005.
[8] T. F. Chan and L. A. Vese, *Active contours without edges*, IEEE Transactions on Image Processing, 10 (2001), pp. 266–277.
[9] 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.
[10] B. Genty and S. Meyer, *Quantitative mapping of leaf photosynthesis using chlorophyll fluorescence imaging*, Australian Journal of Plant Physiology, 22 (1994), pp. 277–284.
[11] 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.
[12] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, 1 (1988), pp. 321–331.
[13] R. Kimmel, *Fast edge integration*, in Geometric Level Set Methods, S. Osher and N. Paragios, eds., Springer-Verlag, 2003, pp. 59–77.
[14] ———, *Numerical Geometry of Images: Theory, Algorithms, and Applications*, Springer, 2004.
[15] 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.
[16] K. W. Morton and D. F. Mayers, *Numerical Solutions of Partial Differential Equations*, Cambridge University Press, 1994.
[17] 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.
[18] 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.
[19] 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.
[20] D. Peak, J. West, S. Messinger, and K. Mott, *Evidence for Complex, Collective Dynamics and Emergent, Distributed Computation in Plants*, vol. 101, 2004.
[21] 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.
[22] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.
[23] 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.
[24] S. Serra-Capizzano, *A note on antireflective boundary conditions and fast deblurring models*, SIAM Journal on Scientific Computing, 25 (2003), pp. 1307–1325.
[25] 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.
[26] 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.
[27] 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.
[28] ———, *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.
[29] 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.
[30] 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).

[31] 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.

[32] H. Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion*, Journal of Computational Physics, 127 (1996), pp. 179–195.