Energy Minimization for Cirrus and Cumulus Cloud Separation in Atmospheric Images

Charles Marshak¹, Igor Yanovsky^{2, 3}, and Luminita Vese¹

¹University of California, Los Angeles, Department of Mathematics, CA 90095, USA ²Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA ³Joint Institute for Regional Earth System Science and Engineering, University of California, Los Angeles, CA 90095, USA

November 28, 2017

Abstract

Multi-angle Imaging Spectro-Radiometer (MISR) instrument provides the multi-angle images of aerosols and clouds. There are a multitude of challenges for stereo imaging of clouds and aerosols including the high variation of radiative properties of aerosols and clouds spanning large altitudes within a two-dimensional image. In this work, we adapt a mathematical model to process these images in order to separate two specific types of clouds frequently appearing in MISR images. Specifically, we separate cirrus and cumulus clouds in the two-dimensional MISR single-channel images. We characterize these two cloud types according to their spatial variations and optical brightness within a given two-dimensional single-channel image. Cirrus clouds appear smooth and optically thin, while cumulus clouds present high optical oscillations and appear brighter. We adapt the additive piecewise-smooth (APS) model of Le and Vese for this cloud separation task. Our methodology uses a single energy minimization function to perform this cloud separation. We compare our results to the previous joint work of the second author on cloud separation.

1 Introduction and Related Work

The radiative forcing of aerosols and clouds in the Earth's atmosphere are key ingredients of climate models [13, 2]. The Multi-angle Imaging Spectro-Radiometer's (MISR) image data provides multi-angle images of clouds to help validate and understand such radiative processes within the atmosphere [6, 19, 17, 1, 5]. The diverse radiative properties of aerosols and clouds that span large altitudes poses a challenge for stereo imaging using the raw MISR images [6, 19]. In this work, we concentrate on identifying and separating cirrus and cumulus clouds within an image to help overcome this obstacle. This image processing task was first identified and studied in [17, 18]. In this work, we adapt a single energy functional from [10] to perform this cloud separation task. The framework we adapt here decomposes an image into a piecewise constant part and a smooth background. This, in turn, provides a way to track the different cloud fronts and assist stereo imaging within the atmosphere.

There are two primary obstacles for cloud separation. The first is the superposition of two clouds within an image. Specifically, the translucent cirrus clouds sit above cumulus clouds in the atmosphere and therefore share many of the same pixels within an image. The second obstacle is the occurrence of localized cumulus clouds that share the optical traits of cirrus clouds. We highlight how this model handles these obstacles and compare our output to that of [17].

In the second author's joint work [17], Yanovsky and Davis combined scale separation, segmentation, and disocclusion for the separation of cirrus and cumulus clouds within a MISR image. In this work, we adapt an energy minimization scheme to perform this cloud separation. Specifically, we use the additive piecewise-smooth

(APS) model from [10]. The APS model is a modification of the seminal work of Chan-Vese [4]. In implementing our algorithms, we carefully follow the literature [8, 4] and discuss the numerical methods employed. We note this approach differs from [17] in that our model decomposes an image into simpler parts, namely piecewise constant segments and a smooth background. We also indicate relevant initializations to obtain the best results for MISR data.

2 The Model

In this section, we discuss the mathematical model that we adapted for cloud separation. We consider singlechannel images. We assume that our image u_0 has the following decomposition:

$$u_0 = u + v,$$

where u is a highly oscillatory function and v is smooth. We have followed the convention that $u_0, u, v : \Omega \to \mathbb{R}$ are continuous functions on a bounded domain. For our application, v will represent an image with only cirrus while $u_0 - v$ will be the image with only cumulus. Moreover, u is the piecewise constant representation of the cumulus clouds segmented in the image. In [10], this image decomposition is called the *additive piecewise-smooth* (APS) model.

From these assumptions about the APS model, we adapt the energy proposed in [10] for the layer separation problem:

$$E_{+}^{CV}(\varphi, v, c_{1}, c_{2}) = \underbrace{\mu \int_{\Omega} \delta(\varphi(x)) |\nabla\varphi(x)| dx}_{(a)} + \underbrace{\nu \int_{\Omega} H(\varphi(x)) dx}_{(b)}$$

$$+ \underbrace{\lambda_{1} \int_{\Omega} (u_{0} - c_{1} - v)^{2} H(\varphi(x)) dx}_{+ \underbrace{\lambda_{2} \int_{\Omega} (u_{0} - c_{2} - v)^{2} (1 - H(\varphi(x)) dx}_{(c')} \\ + \underbrace{\gamma_{1} \int_{\Omega} |\nabla v|^{2} dx + \gamma_{2} \int_{\Omega} |D^{2}v|^{2} dx}_{(d)}.$$

$$(1)$$

This is a variation of the original Chan-Vese model [4] for image segmentation. Here, $\varphi : \Omega \to \mathbb{R}$ is a level-set function whose interface $\{x \in \Omega \mid \varphi(x) = 0\}$ will enclose the thick cumulus layer [12, 11]. The function H is the heavy-side distribution on \mathbb{R} and δ the delta distribution on \mathbb{R} centered at 0, such that $H' = \delta$ [16]. Also, we have used the notation $|D^2v|$ for the Frobenius norm of the Hessian matrix:

$$|D^2v| := \sqrt{v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2}.$$

We now review the geometric interpretations of each term in the energy functional (1). The first term (a) is an approximation of the length of the interface of φ . The second term (b) is the area of the image with $\varphi > 0$. We will ignore this term, setting $\nu = 0$, but include it for historical completeness [4]. The third term (c) and fourth term (c') are the so-called fidelity terms that ensure that the piecewise constants approximate the image reasonably well once v has been removed. The last two terms are the smooth regularizers of the v.

We can also write the energy in (1) in terms of familiar norms from functional analysis:

$$E_{+}^{CV}(\varphi, v, c_{1}, c_{2}) = \mu ||\nabla H(\varphi)||_{L^{2}}^{2} + \lambda_{1} ||u_{0} - c_{1} - v||_{L^{2}, \varphi < 0}^{2} + \lambda_{2} ||u_{0} - c_{1} - v||_{L^{2}, \varphi > 0}^{2} + \gamma_{1} ||v||_{H^{1}}^{2} + \gamma_{2} ||v||_{H^{2}}^{2}.$$

Note that we ignored the area term stemming from (b) in (1). To minimize this energy, we proceed by coordinate descent. Here, we will provide a high-level discussion of this descent and in the Section 3, we will elaborate on the numerical updates.

For fixed level set function φ and smooth inhomogeneity v, we determine c_1 and c_2 via the optimality conditions:

$$c_1 = \frac{\int_{\Omega} (u_0(x) - v(x)) H(\varphi(x)) dx}{\int_{\Omega} H(\varphi(x)) dx}$$
(2)

$$c_{2} = \frac{\int_{\Omega} (u_{0}(x) - v(x))(1 - H(\varphi(x)))dx}{\int_{\Omega} (1 - H(\varphi(x)))dx}.$$
(3)

Indeed, c_1 and c_2 are updated to be the average pixel intensities inside and outside of the zero level set of φ , respectively, after the smooth inhomogeneity v has been removed.

The gradient descent for φ follows nearly the same form as in the Chan-Vese model [4]:

$$\begin{cases} \frac{d\varphi_t}{dt} = \delta(\varphi_t(x)) \left(\mu \operatorname{div} \left(\frac{\nabla \varphi_t}{|\nabla \varphi_t|} \right) - \nu - \lambda_1 (u_0 - v - c_1)^2 + \lambda_2 (u_0 - v - c_2)^2 \right) & x \in \Omega\\ \frac{\delta(\varphi_t(x))}{|\nabla \varphi_t(x)|} \frac{\partial \varphi_t}{\partial \mathbf{n}} = 0 & x \in \partial\Omega. \end{cases}$$
(4)

This requires a semi-implicit scheme and is thoroughly outlined in [8]. We will discuss our implementation in the Section 3 as well.

A key assumption of the APS model is that indeed the length can be approximated as:

$$\int_\Omega \delta(\varphi(x)) |\nabla \varphi(x)| dx \approx \mathrm{Length}(\{x \in \Omega \mid \varphi(x) = 0\}).$$

Indeed, when φ is a signed distance function from the zero level-set φ , the above results in equality. Given an arbitrary level-set function φ that is not necessarily a signed distance function, we can obtain a distance function ψ using the reinitialization procedure determined as in [15, 14] with the same zero-level set. Often such reinitialization is not required, but in the event a level set can become increasingly "flat" [4] around the interface, a reinitialization ensures numerical stability. We employ the fast marching method of Sethian [14] for reinitialization using the **python** package found at [7].

The gradient descent of v is determined similarly:

$$\frac{dv}{dt} = 2\lambda_1(u_0 - c_1 - v)H(\varphi) + 2\lambda_2(u_0 - c_2 - v)(1 - H(\varphi)) + 2\gamma_1\Delta v - 2\gamma_2\Delta^2 v \quad x \in \Omega,$$
(5)

with first, second and third order partial derivatives along the boundary are zero. The boundary conditions are imposed by artificially copying the edge pixels so that the finite difference schemes at the boundary produce zero derivative. We will use a semi-implicit scheme to update v within Ω using standard 5-point and 13-point stencils for Δ and Δ^2 , respectively [3]. We provide pseudocode of the proposed method in Algorithm 1. In the next section we will precisely describe the numerical methods used in our experiments.

3 Numerical Methods

In this section, we discuss implementation details for the APS model.

3.1 Setup

We will view images as real valued matrices. In particular, our original image u_0 will be seen as a matrix in $\mathbb{R}^{M \times N}$ and its domain Ω as a grid corresponding to the indices of this matrix.

Algorithm 1 APS Segmentation

1: procedure APS SEGMENTATION $(u_0, \varphi^0, v^0, \varepsilon, N, n_r)$ $\varphi^k \leftarrow \varphi^0 \ v^k \leftarrow v^0.$ 2: for $k = 1, \ldots, N$ do 3: Update c_1 according to (2). 4: Update c_2 according to (3). 5:Update φ^k using a semi-implicit timestep derived using (4) from φ^{k-1} . 6: Update v^k using a semi-implicit timestep derived using (5) from v^{k-1} . 7: if $||\varphi^k - \varphi^{k-1}||/|\Omega| < \varepsilon$ then 8: break 9: if $k \mod n_r \equiv 0$ then 10: Replace φ^k with signed distance function using [7]. 11: $\varphi^{k-1} \leftarrow \varphi^k, v^{k-1} \leftarrow v^k$ 12:

We store φ as a matrix in $\mathbb{R}^{(M+2)\times(N+2)}$. We will label pixels φ_{ij} as boundary pixels whenever i = 0, i = N+1, j = 0, or j = M+1. These added boundary pixels are to enforce the Neumann boundary conditions discussed in the previous section. This is accomplished by running the descent on the interior matrix (matrix of non-boundary pixels) of size $M \times N$ and then updating boundary pixels whenever an adjacent interior pixel has been updated. The copying of boundary pixels is not needed and could be enforced virtually.

Similarly, we store v as a matrix in $\mathbb{R}^{(M+6)\times(N+6)}$ to ensure the appropriate boundary conditions. In this case, boundary pixels occur whenever $0 \le i \le 2$, $N \le i \le N+2$, $0 \le j \le 2$, or $M \le j \le M+2$. Because these matrices are of different sizes, the same index does not refer to the same pixel in each image. As such, we note in Table 1 how the different indices can be transformed so they refer to the same pixel.

	u_0	φ	v
Relative Indices from u_0	(i,j)	(i+1,j+1)	(i+3, j+3)
Size	$M \times N$	$(M+2) \times (N+2)$	$(M+6) \times (N+6).$

Table 1: Above are the variables u_0, φ, v when viewed as matrices. Our matrices relative indices from u_0 indicate where pixels (i, j) in the original image (u_0) occur in v and φ .

3.2 The φ -update

We will now describe the method for the φ -update in the APS model. Before describing the method, we must discuss the regularized distributions H and φ . For our algorithms, we use the regularizers of [4] and refer the reader to this work for the experimental benefits of this choice. In particular, we define

$$H_{\varepsilon}(x) := \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{x}{\varepsilon}\right) \right)$$
(6)

$$\delta_{\varepsilon}(x) := H'_{\varepsilon}(x) = \frac{\varepsilon}{\pi^2(\varepsilon^2 + x^2)} \tag{7}$$

In all of our experiments, we set $\varepsilon = 1$ as discussed in [4].

We discretize the PDE in (4) as discussed in [8]. We will assume that the relative indices for u_0, φ and v are

identical to ease the exposition, that is to say v_{ij} and φ_{ij} refer to the same pixel. The φ update in this form is:

$$\frac{\varphi_{ij}^{k+1} - \varphi_{ij}^{k}}{dt} = \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) \left[\mu \operatorname{div}_{x}^{+} \left(\frac{\nabla_{x}^{+} \varphi_{ij}^{k}}{\sqrt{\eta^{2} + (\nabla_{x}^{+} \varphi_{ij})^{2} + (\nabla_{y}^{0} \varphi_{ij})^{2}}} \right) + \mu \operatorname{div}_{y}^{+} \left(\frac{\nabla_{y}^{+} \varphi_{ij}^{k}}{\sqrt{\eta^{2} + (\nabla_{x}^{0} \varphi_{ij})^{2} + (\nabla_{y}^{+} \varphi_{ij})^{2}}} \right) \right] + \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) \left[-\nu - \lambda_{1}((u_{0})_{ij} - v_{ij} - c_{1})^{2} + \lambda_{2}((u_{0})_{ij} - v_{ij} - c_{2})^{2} \right].$$
(8)

We have ignored the timestep index atop v_{ij} , c_1 and c_2 as these are constant during this update. Above, ∇^+ and ∇^0 denote the forward and center difference operators, respectively. To properly localize the discretization of div, we note that div $^+ = (\nabla^+)^T$, which is a backward difference. Note that the norm of the gradient $|\nabla \varphi|$ has been regularized to prevent division by zero with $\eta = 10^{-9}$. Moreover, in computation of the norm of the gradient, each term has been discretized differently prior to the backward difference div⁺ for proper localization.

To write out our finite differences, we provide the following definitions:

$$A_{ij} := \frac{\mu}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{ij})^2 + (\nabla_y^0 \varphi_{ij})^2}}}$$
(9)
$$= \frac{\mu}{\sqrt{\eta^2 + (\varphi_{i+1,j}^k - \varphi_{ij}^k)^2 + \frac{1}{2} (\varphi_{i,j+1}^k - \varphi_{i,j-1}^{k+1})^2}},$$

$$B_{ij} := \frac{\mu}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{ij})^2 + (\nabla_y^+ \varphi_{ij})^2}}}$$
(10)
$$= \frac{\mu}{\sqrt{\eta^2 + \frac{1}{2} (\varphi_{i+1,j}^k - \varphi_{i-1,j}^{k+1})^2 + (\varphi_{i,j+1}^k - \varphi_{ij}^k)^2}}.$$

We will assume that we traverse through the matrix from i = 0, ..., N - 1 and j = 0, ..., M - 1 and so indices that have already been updated, will be employed for the subsequent update and to indicate this the appropriate superscript is used. Note that these updates can all be done in place without any extra memory storage. Using the matrices defined above, we can finally write an explicit discretization:

$$\frac{\varphi_{ij}^{k+1} - \varphi_{ij}^{k}}{dt} = \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) \left(A_{ij}(\varphi_{i+1,j}^{k} - \varphi_{ij}^{k+1}) - A_{i-1,j}(\varphi_{ij}^{k+1} - \varphi_{i-1,j}^{k+1})\right) \\
+ \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) \left(B_{ij}(\varphi_{i,j+1}^{k} - \varphi_{ij}^{k+1}) - B_{i,j-1}(\varphi_{ij}^{k+1} - \varphi_{i,j-1}^{k+1})\right) \\
+ \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) \left(-\nu - \lambda_{1}((u_{0})_{ij} - v_{ij} - c_{1})^{2} + \lambda_{2}((u_{0})_{ij} - v_{ij} - c_{2})^{2}\right).$$
(11)

Using the above, and solving for φ_{ij}^{k+1} , we find:

$$\varphi_{ij}^{k+1} = dt \cdot \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) \left[A_{ij}\varphi_{i+1,j}^{k} + A_{i-1,j}\varphi_{i-1,j}^{k+1} + B_{ij}\varphi_{i,j+1}^{k} + B_{i,j-1}\varphi_{i,j-1}^{k+1} -\nu - \lambda_{1}((u_{0})_{ij} - v_{ij} - c_{1})^{2} + \lambda_{2}((u_{0})_{ij} - v_{ij} - c_{2})^{2} \right] / \left[1 + dt \cdot \delta_{\varepsilon} \left(\varphi_{ij}^{k}\right) (A_{ij} + A_{i-1,j} + B_{ij} + B_{i,j-1}) \right].$$
(12)

In current implementation, the boundary conditions are handled as follows. Recall φ is stored as a matrix in $\mathbb{R}^{(M+2)\times(N+2)}$. We only update the interior and when a pixel is adjacent to the boundary, we update boundary

pixels by copying the adjacent interior pixels appropriately. This is a *semi-implicit boundary scheme* because the current boundary iterates are used for the boundary that is subsequently updated. It is well known that boundary conditions are a major source of instability in numerical schemes on bounded regions and we found these updates were crucial for convergence in our experiments. In the next section, we experimented with a fully implicit v-update scheme and found that the resulting method was unstable for many applications.

3.3 The *v*-update

In this section we discuss the v update in the APS model. Let's recall two standard finite difference schemes for the Laplacian Δ and biharmonic operator Δ^2 that occur in (5):

$$P_{\Delta} := \nabla_x^- \nabla_x^+ + \nabla_y^- \nabla_y^+ \tag{13}$$

$$P_{\Delta^2} := P_{\Delta} \circ P_{\Delta}$$

$$- \nabla^- \nabla^+ \nabla^- \nabla^+ + \nabla^- \nabla^+ \nabla^- \nabla^+ + \nabla^- \nabla^+ \nabla^- \nabla^+ + \nabla^- \nabla^+ \nabla^- \nabla^+$$
(14)

$$= \nabla_x^- \nabla_x^+ \nabla_x^- \nabla_x^+ + \nabla_y^- \nabla_y^+ \nabla_x^- \nabla_x^+ + \nabla_x^- \nabla_x^+ \nabla_y^- \nabla_y^+ + \nabla_y^- \nabla_y^+ \nabla_y^- \nabla_y^+$$

where the composition above is defined in terms of difference operators. On the interior of the image we write our finite differences explicitly as follows:

$$P_{\Delta}(v_{ij}) = 4v_{ij}^{k+1} - v_{i-1,j}^{k+1} - v_{i+1,j}^{k} - v_{i,j-1}^{k+1} - v_{i,j+1}^{k}$$
(15)

$$P_{\Delta^2}(v_{ij}) = 20v_{ij}^{k+1} - 8v_{i-1,j}^{k+1} - 8v_{i+1,j}^k - 8v_{i,j-1}^{k+1} - 8v_{i,j+1}^k$$

$$(16)$$

$$+ 2v_{i-1,j-1}^{k+1} + 2v_{i+1,j+1}^{k} + 2v_{i+1,j-1}^{k} + 2v_{i-1,j+1}^{k} + v_{i,j+2}^{k} + v_{i+2,j}^{k} + v_{i,j-2}^{k+1} + v_{i-2,j}^{k+1}.$$

We have used the semi-implicit scheme very similarly to that in Section 3.2 for the update of φ . Along the boundary pixels, our updates may become numerically unstable.

We mention that there are many ways to update v along the boundary. The Nuemann boundary conditions dictate that pixels along the boundary have the same value. During the v-update, we may elect to replace all the pixels that must be identical with the newest iterate and then create a scheme that is solved implicitly. This implicit approach is pursued in [9] when a heat-equation dictates a particular minimization. As an example, for the Laplacian stencil in the lower left corner i = 0, j = 0, the implicit method is:

$$P_{\Delta}^{\rm im}(v_{0,0}) = 2v_{0,0}^{k+1} - v_{1,0}^k - v_{0,1}^k.$$

We found the above stencil to be unstable in our numerical experiments on MISR data. However, this fully implicit update worked well for the examples found in the original synthetic examples presented in [10]. In Figure 1c, we provide a matrix visualization of the implicit stencil in the lower left corner (that is when i = 0, j = 0) in which all the blue shaded pixels have the same value.

Instead, along the boundary, we use an older value of the pixel for all but the center pixel. For example, for the Laplacian stencil in the lower left corner i = 0, j = 0, we use:

$$P_{\Delta}(v_{0,0}) = 4v_{0,0}^{k+1} - 2v_{0,0}^k - v_{1,0}^k - v_{0,1}^k.$$

In Figure 1a and 1b, we provide a matrix visualization of these stencils.

We can now write our discretization of (5) as follows:

$$\frac{v_{ij}^{k+1} - v_{ij}^{k}}{dt} = 2\lambda_1((u_0)_{ij} - c_1 - v_{ij}^{k+1})H(\varphi_{ij}) + 2\lambda_2((u_0)_{ij} - c_2 - v_{ij}^{k+1})(1 - H(\varphi_{ij})) + 2\gamma_1 P_{\Delta}(v_{ij}) - 2\gamma_2 P_{\Delta^2}(v_{ij}).$$
(17)



Figure 1: Above, Figure 1a and 1b show the stencils for the two partial differential operators used in the v update. Assuming we move from left to right and bottom to top during our update of the v matrix, the gray highlighted portion of the matrix denotes those pixels that will have been updated within the current time step. Figure 1a and 1b provide interior stencils. The dark blue pixel in the center of each matrix denotes the pixel that we are currently updating. The lighter gray pixel are the newest pixels updates that are used in the current stencil. Figure 1c denotes an implicit scheme for left-corner boundary, in which all the dark blue pixels have identical values. We found this fully implicit update to be unstable.

Above, we have removed the time dependence of c_1, c_2 , and φ_{ij} as they are held constant while v_{ij}^k is updated. Deriving the precise stability conditions for this numerical method is beyond the scope of this work. Employing (15), (16), and (17), we can produce the following update for v_{ij}^{k+1} :

$$v_{ij}^{k+1} = dt \left[2\lambda_1((u_0)_{ij} - c_1)H(\varphi_{ij}) + 2\lambda_2((u_0)_{ij} - c_2)(1 - H(\varphi_{ij})) - v_{i-1,j}^{k+1} - v_{i,j-1}^k - v_{i,j-1}^{k} - v_{i,j+1}^k - v_{i,j-1}^{k} - v_{i,j-1}^k - v_{i,j+1}^k \right] - 8v_{i-1,j}^{k+1} - 8v_{i,j-1}^{k} - 8v_{i,j+1}^{k+1} + 2v_{i-1,j-1}^{k+1} + 2v_{i+1,j+1}^k + 2v_{i+1,j-1}^k + 2v_{i-1,j+1}^k + v_{i,j+2}^k + v_{i+2,j}^{k+1} + v_{i,j-2}^{k+1} + v_{i-2,j}^k \right] / \left[1 + dt(2\lambda_1 H(\varphi_{ij}) + 2\lambda_2(1 - H(\varphi_{ij})) - 8\gamma_1 + 40\gamma_2) \right].$$

$$(18)$$

3.4 Initial Conditions

Since the energy is non-convex, the minimum and resulting cloud separation is highly dependent on how we initialize v and φ . We found that initializing φ in a checkerboard like fashion was very effective. As in [8], we initialized

$$\varphi(x) = \sin\left(\frac{\pi}{5}x_1\right)\sin\left(\frac{\pi}{5}x_2\right).$$

Since the cumulus often occurred throughout an entire image it was important for φ to be equally distributed.

The initialization of v was less straightforward. Finding a suitable v greatly impacted our results, particularly as it was the smooth representation of cirrus. In Figure 2, we display several choices for initial v. One candidate for the initialization of v is the original image itself. Another possibility is thresholding bright regions of the images as in Figure 2b. Specifically, we employed the following:

$$v(x) = \begin{cases} u_0(x) & \text{if } u_0(x) (19)$$

where $p \in (0, 1)$. In Figure 2b, we selected p = .45. Yet another possible initialization for v was to remove brightest regions entirely. We can set:

$$v(x) = \begin{cases} u_0(x) & \text{if } u_0(x) (20)$$

where $p \in (0, 1)$. In Figure 2b, we selected p = .45. In 2d, we have convolved our image with a Gaussian $v(x) = (G_{\sigma} * u_0)(x)$, where $\sigma = 10$. In Figure 2e, we combined thresholding as in (19) and the Gaussian blur. In

Figure 2f, we combined thresholding and a low pass filter. We describe the low pass filter now. If $u_0 \in \mathbb{R}^{M \times N}$ is the original image, let $F(u_0)$ be the Fourier transform of u_0 and:

$$\begin{cases}
K = \min(M, N) \\
r_p = p \cdot K \\
c = \left(\frac{M}{2}, \frac{N}{2}\right)
\end{cases}$$
(21)

where $p \in (0, 1)$. We set F(v) to 0 that are outside of the circle with radius r_p in Ω . Explicitly, we have:

$$U_p(x) = \begin{cases} F(u_0)(x) & \text{if } ||x - c||_2 \le r_p \\ 0 & \text{otherwise.} \end{cases}$$
(22)

Our initialization for v can be written as $v = F^{-1}(U_p)$. In Figure 2f, we apply a low pass filter using (22) with p = .1 after we first threshold an image with p = .45 using (19).

4 Results

In this section, we present the results of the APS model. This model had different output than the methodology of [17]. Namely, in our methodology, we find a smooth C^2 representation (v) of cirrus within an image. The cumulus is represented as the removal of this smooth part from the image $(u_0 - v)$. We also have a simple, piecewise constant representation (u) of the cumulus clouds in the image. However, the energy does not accurately capture the textures of cirrus and cumulus parts and this can be problematic for the decomposition.

In Figures 3 and 4, we display the best results of our applications of the APS model after we tuned the parameters of the model. We make some important observations. Firstly, in both images we see the overall texture of the cirrus is lost as in the bottom right of Figure 3a. In the original work of [17], the texture of cirrus is more accurately captured after the decomposition of the image into cirrus and cumulus. Secondly, for the two examples provided, we removed regions with high pixel value (bright pixels) using (19) for the initialization of v. Determining a threshold, independent of the maximum brightness, would be required when running this across MISR imagery. Thirdly, we notice that strongly localized cumulus in an image will usually appear in v, as cirrus. In Figure 4a, there are localized cumulus clouds both on the right and center of the image; however, these two regions are classified as cirrus with the APS model as seen in Figure 4f. In fact, the v-updates usually appeared as blurring in our numerical experiments. Lastly, we notice that the piecewise constant model for cumulus intensities. Certainly, our method detects many cumulus clouds within an image. However, certain small clouds are removed and lost from our decomposition as is shown in Figure 3f. The APS model [10] assumes that each segmented class, that is the class of pixels with $\varphi > 0$ and those with $\varphi < 0$, have roughly similar intensity. Frequently, cumulus clouds may have varying intensity throughout a single image.

5 Conclusion

We adapted the APS model [10] for the two layer cirrus and cumulus separation problem originally investigated in [17]. We provided a detail explanation of the numerical setup and demonstrated the decomposition of atmospheric images into piecewise constant segments and a smooth part. We provided a detailed discussion of the parameter space of this non-convex problem and our preprocessing steps. We used this methodology to separate cirrus and cumulus clouds within MISR images. The APS model had some significant differences from methodology of [17]. Most importantly, the cirrus clouds were modeled as smooth functions and much of the cirrus texture was lost. Moreover, the model frequently modeled localized cumulus as a smooth background, even after removing bright regions of the images were removed during our preprocessing. However, with these limitations in mind, this method provides a single energy minimization to decompose an image. In addition, this framework proved effective in decomposing an image into meaningful (though less detailed) piecewise constant and smooth pieces to understand the important cloud features within MISR images of the atmosphere.





(e) First, thresholding as in 2b and then a Gaussian (f) First, thresholding as in Figure 2b and then a low Blur with $\sigma = 10$. pass filter.

Figure 2: Choices for initial v.



(a) Original MISR Image

(b) φ^0







(f) Modelled Cumulus + Modelled Cirrus

Figure 3: $\mu = 127$, $\gamma_1 = 1$, $\gamma_2 = 84375$. Function v is initialized using a threshold and then a low pass filter, as in Figure 2f. The image was thresholded using (19) with p = .3 and a low pass filter was applied as constructed in Section 3.4 with p = .1.



(a) Scaled Superposition of Figure 4b and 4c from [17].

(b) Original Cirrus



(c) Original Cumulus



(d) φ^0







(g) Final $u_0 - v$; Modelled Cumulus

(h) Modelled Cumulus + Modelled Cirrus

Figure 4: $\mu = 127$, $\gamma_1 = 1$, $\gamma_2 = 135886$. Function v is initialized using a threshold and then a low pass filter, as in Figure 2f. The image was thresholded using (19) with p = .3 and a low pass filter was applied as constructed in Section 3.4 with p = .1. 11

6 Acknowledgements

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Igor Yanovsky and Luminita Vese are thankful for the National Science Foundation Grant DMS 1217239. We are grateful for the numerous helpful discussions with Triet Le to effectively adapt the original APS model for the current application. © 2017. All rights reserved.

References

- Amit Aides, Yoav Y. Schechner, Vadim Holodovsky, Michael J. Garay, and Anthony B. Davis. Multi Sky-View 3D Aerosol Distribution Recovery. Optics Express, 21(22):25820–25833, 2013.
- [2] Sandrine Bony and Jean-Louis Dufresne. Marine Boundary Layer Clouds at the Heart of Tropical Cloud Feedback Uncertainties in Climate Models. *Geophysical Research Letters*, 32(20), 2005.
- [3] Dietrich Braess. Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics. Cambridge University Press, 2007.
- [4] Tony F. Chan, B. Yezrielev Sandberg, and Luminita A. Vese. Active Contours without Edges for Vectorvalued Images. *Journal of Visual Communication and Image Representation*, 11(2):130–141, 2000.
- [5] Anthony B. Davis and Alexander Marshak. Solar Radiation Transport in the Cloudy Atmosphere: a 3D Perspective on Observations and Climate Impacts. *Reports on Progress in Physics*, 73(2):026801, 2010.
- [6] David J. Diner, Jewel C. Beckert, Terrence H. Reilly, Carol J. Bruegge, James E. Conel, Ralph A. Kahn, John V. Martonchik, Thomas P. Ackerman, Roger Davies, and Siegfried A.W. Gerstl. Multi-angle Imaging Spectro-Radiometer (MISR) Instrument Description and Experiment Overview. *IEEE Transactions on Geoscience and Remote Sensing*, 36(4):1072–1087, 1998.
- [7] Jason Furtney. Scikit-FMM. https://pypi.python.org/pypi/scikit-fmm.
- [8] Pascal Getreuer. Chan-Vese Segmentation. Image Processing On Line, 2:214–224, 2012.
- [9] Tom Goldstein and Stanley Osher. The Split-Bregman Method for ℓ_1 -regularized Problems. SIAM journal on imaging sciences, 2(2):323–343, 2009.
- [10] Triet M. Le and Luminita A. Vese. Additive & Mutiplicative Piecewise-smooth Segmentation Models in a Functional Minimization Approach. *Contemporary Mathematics*, 445:207–224, 2007.
- [11] Stanley Osher and Ron Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Applied Mathematical Sciences. Springer-Verlag, New York, 2003.
- [12] Stanley Osher and James Sethian. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. JCP, 79:12–49, 1988.
- [13] V. Ramanathan, P.J. Crutzen, J.T. Kiehl, and D. Rosenfeld. Aerosols, Climate, and the Hydrological Cycle. Science, 294(5549):2119–2124, 2001.
- [14] James Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. Proceedings of the National Academy of Sciences, 93(4):1591–1595, 1996.
- [15] Mark Sussman, Peter Smereka, and Stanley Osher. A Level Set Approach for Computing solutions to Incompressible Two-phase Flow. Journal of Computational Physics, 114(1):146–159, 1994.

- [16] Terence Tao. Epsilon of Room, volume 1. American Mathematical Soc., 2010.
- [17] Igor Yanovsky and Anthony B. Davis. Separation of a Cirrus Layer and Broken Cumulus Clouds in Multispectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2275–2285, May 2015.
- [18] Igor Yanovsky, Anthony B. Davis, and Veljko M. Jovanovic. Separation of Cloud Layers in Multispectral Imager Data. In *IEEE Geoscience and Remote Sensing Symposium*, pages 1627–1630, July 2014.
- [19] Jia Zong, Roger Davies, Jan-Peter Muller, and David J. Diner. Photogrammetric Retrieval of Cloud Advection and Top Height from the Multi-angle Imaging Spectro-Radiometer (MISR). *Photogrammetric Engineering and Remote Sensing*, 68(8):821–830, 2002.