Automatic Characterization of Cross-sectional Coated Particle Nuclear Fuel using Greedy Coupled Bayesian Snakes

Jeffery R. Price, Deniz Aykac, John D. Hunn, Andrew K. Kercher

Oak Ridge National Laboratory, Oak Ridge, TN USA

ABSTRACT

We describe new image analysis developments in support of the U.S. Department of Energy's (DOE) Advanced Gas Reactor (AGR) Fuel Development and Qualification Program. We previously reported a non-iterative, Bayesian approach for locating the boundaries of different particle layers in cross-sectional imagery. That method, however, had to be initialized by manual preprocessing where a user must select two points in each image, one indicating the particle center and the other indicating the first layer interface. Here, we describe a technique designed to eliminate the manual preprocessing and provide full automation. With a low resolution image, we use "EdgeFlow" to approximate the layer boundaries with circular templates. Multiple snakes are initialized to these circles and deformed using a greedy Bayesian strategy that incorporates coupling terms as well as a priori information on the layer thicknesses and relative contrast. We show results indicating the effectiveness of the proposed method.

Keywords: Image segmentation, EdgeFlow, snakes, multiple active contours, coated-particle nuclear fuel, TRISO fuel, image-based metrology

1. INTRODUCTION

The U.S. Department of Energy's (DOE) Advanced Gas Reactor (AGR) Fuel Development and Qualification Program was established to support near-term deployment of high-temperature, gas-cooled reactor technology. This reactor technology employs silicon carbide-based coated particle nuclear fuel, commonly known as "TRISO" fuel. As a part of this program, Oak Ridge National Laboratory (ORNL) has established capabilities to thoroughly characterize such particles. Over the last several years, we have developed metrology techniques employing optical microscopy, digital imaging, and image processing to characterize the shape and coating thicknesses of the particles.[1,2] In this work, we are concerned particularly with cross-sectional images such as the fully-coated examples shown in Fig. 1. Referring to Fig. 1, the particle layers from the inside out are (1) the fuel *kernel*, (2) a porous carbon *buffer*, (3) an inner layer of pyrocarbon referred to as *IPyC*, (4) a layer of silcon carbide referred to as *SiC*, and finally (5) an outer layer of pyrocarbon referred to as *OPyC*. To acquire images such as these, we first embed the particles in a mount of acrylic epoxy and polish them to their approximate mid-plane. The mounts are next placed in an optical microscope and illuminated with a combination of reflected and back lighting; images are acquired with a high-resolution digital camera attached to the microscope.

In our previous work, we developed a semi-automatic system to locate the different layer boundaries in these images. In the first step of this previous system, the user is presented with a cross-sectional image and must indicate, via point-and-click, the approximate kernel center and one point on the kernel-buffer interface. These points are used to set *a priori* information for a Bayesian-motivated boundary finding method. After this initial user interaction, all subsequent processing is performed automatically. The ultimate goal of this work is to eliminate the user interaction so that the entire process can be fully automated. Towards that goal, we adopt an approach that employs multiple active contours or snakes.[3] Our approach is motivated in part by earlier work of Abe and Matsuzawa,[7] except that we incorporate *a priori* shape information and do not use sub-region competition. We also adopt and expand upon some characteristics from the work of Gunn and Nixon[6] with regard to shape and smoothness.

The remainder of this paper is organized as follows. In Section 2, we present a method for intializing the contours to circles using an efficient implementation of the EdgeFlow[4] technique followed by likelihood-based

Correspondence: J. Price, pricejr@ornl.gov, 865-574-5743

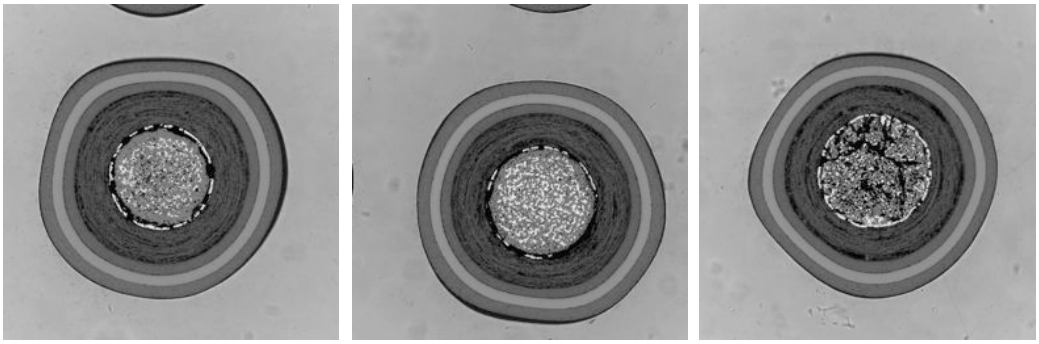

Figure 1. Three representative particle images. Each image represents an area of 1.09 mm × 1.09 mm. Note the inhomogeneous kernel region and the unpredictable nature of the kernel-to-buffer transition.

template detection. We then describe the multiple active contour model in Section 3. The contours, one for each layer interface in the particle, evolve together under a set of criteria capturing *a priori* shape information, smoothness, and intensity relationships. These criteria are expressed in a probabilistic fashion and the contours are deformed iteratively via a greedy, maximum likelihood search. In Section 4, we demonstrate results of the proposed technique and then close in Section 5.

2. INITIALIZING WITH EFFICIENT EDGEFLOW

"EdgeFlow" is a boundary detection algorithm proposed by Ma and Manjunath.[4] Though it can be used to detect intensity and/or texture boundaries, we use it only for intensity (grayscale) boundaries herein. In this section, we first briefly review the EdgeFlow construction for intensity edges (see the original reference[4] for more detail). We then make some notes on efficient implementation using Fourier properties and steerability[5] principles and subsequently describe how we use the EdgeFlow result to initialize our multi-contour model.

2.1. Intensity EdgeFlow

Let the two-vector $\mathbf{x} = [x_1\, x_2]^T$ represent two-dimensional (2D) spatial coordinates in the image plane, referenced to the image center, and select a suitable finite set of angles equally spaced over $[0, 2\pi)$ – we use the 12 angles $\theta \in \{0, \pi/12, \pi/6, ..., 11\pi/12\}$. Define the 2D Gaussian filter by

$$g(\mathbf{x}) = \exp\Big(\frac{-\|\mathbf{x}\|^2}{2s^2}\Big) \tag{1}$$

with directional derivatives in the cardinal directions $g_1(\mathbf{x}) = \frac{\partial}{\partial x_1} g(\mathbf{x})$ and $g_2(\mathbf{x}) = \frac{\partial}{\partial x_2} g(\mathbf{x})$ and where s is a user-defined scale parameter. The directional derivative in an arbitrary direction θ is then given by

$$g_\theta(\mathbf{x}) = \frac{\partial}{\partial \mathbf{n}_\theta} g(\mathbf{x}), \tag{2}$$

where $\mathbf{n}_\theta$ is the normal vector pointing in direction θ. We also define the EdgeFlow "difference-of-offset-Gaussians" filter, $\gamma_\theta(\mathbf{x})$, as

$$\gamma_\theta(\mathbf{x}) = g(\mathbf{x}) - g(\mathbf{x} + \mathbf{d}_\theta) \tag{3}$$

where the offset vector $\mathbf{d}_\theta$ is given by $\mathbf{d}_\theta = d[\cos(\theta)\ \sin(\theta)]^T$ and d is scale parameter set to $4s$.

Letting the image of interest be represented by $u(\mathbf{x})$ and using the notation just described, the EdgeFlow method defines a directional *edge energy* term, $\epsilon_\theta(\mathbf{x})$, as

$$\epsilon_\theta(\mathbf{x}) = |u(\mathbf{x}) * g_\theta(\mathbf{x})|, \tag{4}$$

using the directional derivatives of the Gaussian from Eq. (2). EdgeFlow also defines a directional *edge probability* term, $p_\theta(\mathbf{x})$, as

$$p_\theta(\mathbf{x}) = \frac{|u(\mathbf{x}) * \gamma_\theta(\mathbf{x})|}{|u(\mathbf{x}) * \gamma_\theta(\mathbf{x})| + |u(\mathbf{x}) * \gamma_{\theta+\pi}(\mathbf{x})|}, \tag{5}$$

using the difference-of-offset-Gaussian filters from Eq. (3). Intuitively, a large $p_\theta(\mathbf{x})$ value implies that there is a significant change in grayscale value along θ and, therefore, that there is a good probability of finding a boundary in that direction. With these definitions of edge energy and edge probability, the EdgeFlow vector field, $\mathbf{f}(\mathbf{x})$, is then given by

$$\mathbf{f}(\mathbf{x}) = \sum_{\theta=\hat{\theta}(\mathbf{x})-\pi/2}^{\hat{\theta}(\mathbf{x})+\pi/2} \epsilon_\theta(\mathbf{x}) \exp(j\theta) \tag{6}$$

where $\hat{\theta}(\mathbf{x})$ is defined to be the direction that maximizes the half-plane edge probabilities, i.e.,

$$\hat{\theta}(\mathbf{x}) = \arg\max_\theta \sum_{\theta'=\theta-\pi/2}^{\theta+\pi/2} p_{\theta'}(\mathbf{x}). \tag{7}$$

2.2. Efficient EdgeFlow

The first derivative of the Gaussian is steerable,[5] hence its derivative in any direction θ (with normal vector $\mathbf{n}_\theta$) can be written as a linear combination of the two cardinal directional derivatives:

$$g_\theta(\mathbf{x}) = \frac{\partial}{\partial \mathbf{n}_\theta} g(\mathbf{x}) = \cos(\theta) g_1(\mathbf{x}) + \sin(\theta) g_2(\mathbf{x}). \tag{8}$$

Due to the linearity of convolution, we can then express edge energy from Eq. (4) as

$$\epsilon_\theta(\mathbf{x}) = |\cos(\theta)\big(u(\mathbf{x}) * g_1(\mathbf{x})\big) + \sin(\theta)\big(u(\mathbf{x}) * g_2(\mathbf{x})\big)|. \tag{9}$$

Hence, to compute the edge energy at any angle θ we need only the convolutions of the image with the cardinal direction derivatives. Letting $U(\boldsymbol{\omega})$, $G_1(\mathbf{w})$, and $G_2(\boldsymbol{\omega})$ represent the Fourier transforms of $u(\mathbf{x})$, $g_1(\mathbf{x})$, and $g_2(\mathbf{x})$, respectively, and defining $\mathcal{F}^{-1}$ to be the inverse Fourier transform operator, we can rewrite Eq. (9) as

$$\epsilon_\theta(\mathbf{x}) = |\mathcal{F}^{-1}\{\cos(\theta)U(\boldsymbol{\omega})G_1(\boldsymbol{\omega}) + \sin(\theta)U(\boldsymbol{\omega})G_2(\boldsymbol{\omega})\}| \tag{10}$$

Similarly, letting $\Gamma_\theta(\boldsymbol{\omega})$ and $G(\boldsymbol{\omega})$ represent the Fourier transforms of $\gamma_\theta(\mathbf{x})$ and $g(\mathbf{x})$, respectively, we can rewrite Eq. (3) in the Fourier domain as

$$\Gamma_\theta(\boldsymbol{\omega}) = G(\boldsymbol{\omega})\big(1 - \exp(-j\boldsymbol{\omega}^T \mathbf{d}_\theta)\big) = G(\boldsymbol{\omega}) A_\theta(\boldsymbol{\omega}) \tag{11}$$

where

$$A_\theta(\boldsymbol{\omega}) = (1 - \exp(-j\boldsymbol{\omega}^T \mathbf{d}_\theta)) \tag{12}$$

This implies that Eq. (5) can be expressed as

$$p_\theta(\mathbf{x}) = \frac{|\mathcal{F}^{-1}\{U(\boldsymbol{\omega})\Gamma(\boldsymbol{\omega})A_\theta(\boldsymbol{\omega})\}|}{|\mathcal{F}^{-1}\{U(\boldsymbol{\omega})\Gamma(\boldsymbol{\omega})A_\theta(\boldsymbol{\omega})\}| + |\mathcal{F}^{-1}\{U(\boldsymbol{\omega})\Gamma(\boldsymbol{\omega})A_{\theta+\pi}(\boldsymbol{\omega})\}|}, \tag{13}$$

Using the above relationships, we can efficiently implement the EdgeFlow technique using fast Fourier transforms (FFTs) as follows. We first compute $U(\boldsymbol{\omega})$, $G(\mathbf{w})$, $G_1(\mathbf{w})$, and $G_2(\mathbf{w})$. Then, for each angle θ to be evaluated, we compute $A_\theta(\mathbf{w})$ from Eq. (12) and the edge energy and edge probability using Eq. (10) and Eq. (13), respectively. We then compute the EdgeFlow vector field, $\mathbf{f}(\mathbf{x})$, via Eq. (6). For our purposes, we are only interested in the magnitude $|\mathbf{f}(\mathbf{x})|$, which we normalize to $[0, 1]$. The EdgeFlow magnitudes for the example images from Fig. 1 are shown in Fig. 2.

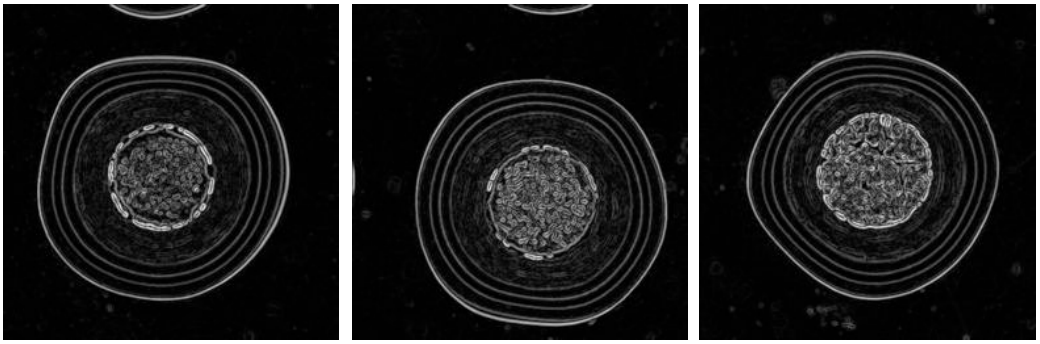

Figure 2. EdgeFlow magnitudes – $|\mathbf{f}(\mathbf{x})|$ from Eq. (6) – corresponding to example particle images of Fig. 1.

2.3. Template Detection for Initialization

Though the layer boundaries are approximately circular, there is often enough variation from perfect circularity to limit the efficacy of the well-known Hough transform approach. Instead, we adopt a simple and effective template-based strategy as follows. We first note that we have *a priori* information about the contours from the fuel particle design specifications and historical analysis. For a given batch of particle images to analyze, we know the approximate mean thickness and standard deviations of the thickness for each layer.

Based upon the *a priori* information, we define a template image, $t_c(\mathbf{x})$, for each layer boundary $c = 1, \ldots, C$ as

$$t_c(\mathbf{x}) = \exp\left(\frac{-(\|\mathbf{x}\| - r_c)^2}{2\sigma_c^2}\right) \tag{14}$$

where r_c is the radius based upon design specifications and σ_c is the standard deviation of that radius based upon historical anaylsis. Each such template, $t_c(\mathbf{x})$, is a probabilistic representation of the layer boundary; it is a Gaussian-profile circle around the image center with radius r_c and with width determined by σ_c. The likelihood of finding an approximately circular boundary c centered on $\mathbf{x}$ can then be approximated as $t_c(\mathbf{x}) * |\mathbf{f}(\mathbf{x})|$. With this in mind, we compute the net log likelihood, $\mathcal{T}(\mathbf{x})$, using all of the boundary templates, using

$$\mathcal{T}(\mathbf{x}) = \sum_{c=1}^{C} \log\big(t_c(\mathbf{x}) * |\mathbf{f}(\mathbf{x})|\big). \tag{15}$$

We then select the maximizing coordinate as the initial center point, $\mathbf{o}_{\text{init}}$, of our multiple active contours:

$$\mathbf{o}_{\text{init}} = \arg\max_{\mathbf{x}} \mathcal{T}(\mathbf{x}). \tag{16}$$

In Fig. 3, we show contour plots of the net likelihoods from Eq. (15) for the example particles from Fig. 1; the maximizing coordinate defined by Eq. (16) is also indicated in the figure. We initialize our contours as N points on C circles centered on $\mathbf{o}_{\text{init}}$. Examples of such initializations for the particles of Fig. 1 are shown later in Fig. 7, where $N = 48$ and $C = 5$.

3. BOUNDARY FINDING WITH GREEDY, COUPLED BAYESIAN SNAKES

In this section, we describe our multi-contour model and develop the criteria under which the contours evolve. These criteria are expressed as likelihoods based upon normal (Gaussian) parameterization. We make the (unrealistic though effective) assumption that the criteria are independent from one another; this implies that total log likelihoods can be computed by summing the individual log likelihoods. The goal of the multi-contour model is, through iterative refinement, to find the maximum likelihood (ML) solution. We have broken the criteria down into intensity-based likelihoods, described in Section 3.1, and shape-based likelihoods, described in Section 3.2. We conclude this section with some notes on the iteration and convergence criteria in Section 3.3.

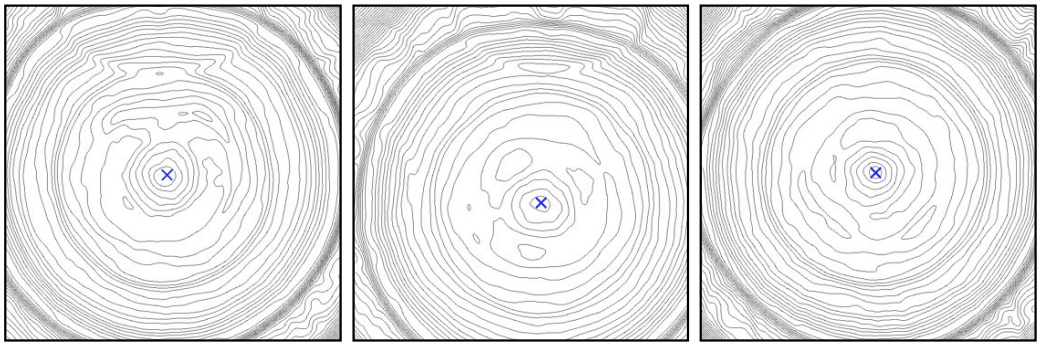

Figure 3. Results of likelihood-based template detection results, described in Section 2.3, for the example particles displayed in Fig. 1. The "×" symbol indicates the coordinate selected as the center point for each of the initial circular contours.

To ensure notational clarity henceforth, note that we use C as the number of contours, each of which is represented by N discrete points. Also, we let $\mathbf{x}_c^p$ represent the image plane location of point p on contour c, where $p \in \{1, \ldots, N\}$ and $c \in \{0, \ldots, C\}$, with $c = 0$ corresponding to the particle center point. All mathematical operations with the point index p are performed modulo N. We will use $\mathbf{y}_c^p$ to represent a specific location for which we are evaluating the likelihood that the location is point p on contour c.

3.1. Intensity Criteria

Considering the particle images, there are a few properties regarding the intensities that we would like to capture with the final boundary locations. In expressing the associated likelihood approximations, we adopt a measure of statistical similarity (or difference) between image regions that is illustrated in Fig. 4 and can be described as follows. Suppose we wish to compare two image regions $\mathcal{R}_1$ and $\mathcal{R}_2$. We compute the means and standard deviations of the pixel intensities in these two regions and consider the associated normal distributions $\mathcal{N}_1$ and $\mathcal{N}_2$. The similarity between regions $\mathcal{R}_1$ and $\mathcal{R}_2$ is denoted by $\mathcal{S}(\mathcal{R}_1, \mathcal{R}_2)$ and is given by the shaded area in Fig. 4 which is the Bayes error rate when discriminating between two normal distributions. This similarity metric takes values on $[0, 1]$. Alternatively, the quantity $1 - \mathcal{S}(\mathcal{R}_1, \mathcal{R}_2)$ is a measure of the difference between the regions and also takes values on $[0, 1]$.

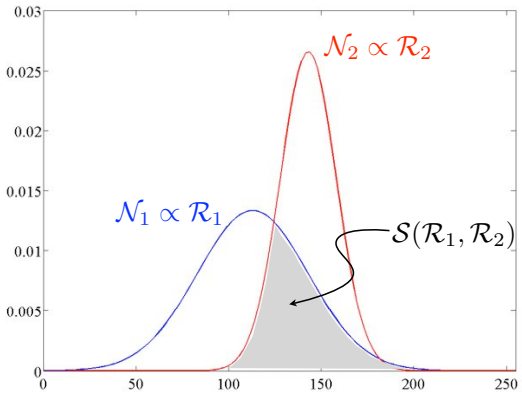

Figure 4. For all of the grayscale criteria of Section 3.1, we compare two image regions by parameterizing their intensity values with normal (Gaussian) distributions and computing the area under the intersection (shaded region).

In defininining the following intensity criteria, we refer to Fig. 5. We first note that a boundary should separate regions of different intensities, i.e., $\mathbf{y}_c^p$ should be on an edge. If $\mathbf{y}_c^p$ is indeed a boundary point, the pixels in I_c^p and O_c^p should therefore be statistically different. This leads us to express the *edge likelihood*, $L_1(\mathbf{y}_c^p)$, as

$$L_1(\mathbf{y}_c^p) = \log\big(1 - \mathcal{S}(I_c^p, O_c^p)\big). \tag{17}$$

If the two regions I_c^p and O_c^p are stastically different, then L_1 will approach $\log(1) = 0$. Regarding the edge likelihoods, we note that known properties of the intensity transitions are used to rule out nonsensical edges. For example, referring back to Fig. 1, the *IPyC* to *SiC* transition is always from a dark-to-bright intensity change (from the center outward). If we are evaluating points for this boundary, we set the likelihood to (near) zero for any edge transitions that do not match this fact.

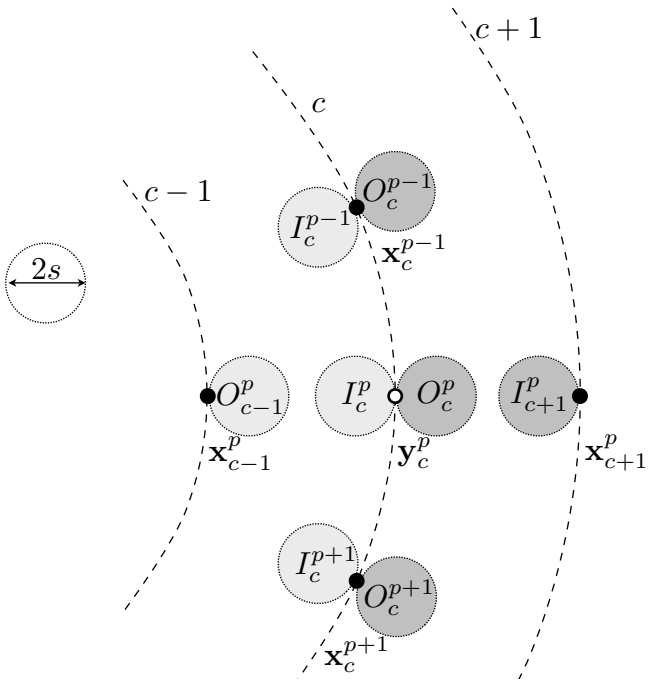

Figure 5. Regions used when computing intensity-based likelihoods while evaluating $\mathbf{y}_c^p$ for selection as point p on contour c as described in Section 3.1. Each of the indicated regions is a disk of diameter $2s$. I and O are used to represent "inner" and "outer" respectively.

If $\mathbf{y}_c^p$ is a boundary point, we expect the intensities for regions around the neighbors of $\mathbf{y}_c^p$ on contour c, namely $\mathbf{x}_c^{p-1}$ and $\mathbf{x}_c^{p+1}$, to be homogeneous with corresponding regions around $\mathbf{y}_c^p$. Referring to Fig. 5, this would imply the regions I_c^{p-1}, I_c^p, and I_c^{p+1} are similar to one another and that the regions O_c^{p-1}, O_c^p, and O_c^{p+1} are similar to one another. This leads us to express the *neighbor homogeneity likelihood*, $L_2(\mathbf{y}_c^p)$, as

$$L_2(\mathbf{y}_c^p) = \log\big(\mathcal{S}(I_c^p, I_c^{p-1})\big) + \log\big(\mathcal{S}(I_c^p, I_c^{p+1})\big) + \log\big(\mathcal{S}(O_c^p, O_c^{p-1})\big) + \log\big(\mathcal{S}(O_c^p, O_c^{p+1})\big). \tag{18}$$

We note, however, that due to the unpredictability of the kernel-to-buffer transition, the first two terms in Eq. (18) are not used for $c = 1$.

The region between contour c and its inner neighbor contour $c-1$ should be homogeneous, as should the region between contour c and its outer neighbor contour $c+1$. In other words, if $\mathbf{y}_c^p$ is a good boundary point, then O_{c-1}^p and I_c^p should be similar, as should O_c^p and I_{c+1}^p. These yields us to propose the *layer homogeneity likelihood*, $L_3(\mathbf{y}_c^p)$, given by

$$L_3(\mathbf{x}_c^p) = \log\big(\mathcal{S}(I_c^p, O_{c-1}^p)\big) + \log\big(\mathcal{S}(O_c^p, I_{c+1}^p)\big) \tag{19}$$

The first term, however, is excluded from the computation for $c = 1$ due to the inhomogeneous nature of the kernel.

3.2. Shape Criteria

We define several shape-based criteria to ensure relatively smooth contours and incorporate *a priori* information on the particle and layer dimensions into the evolution. In constructing these criteria, we refer to Fig. 6, where

all vectors are referenced to the point $\mathbf{o}$ which is defined by a circle fit[8] to the $c = 1$ contour. From the design specifications, we know the mean thickness, μ_c, and approximate standard deviation, σ_c for each layer (contour) c. In Fig. 6, we note that the layer thickness assuming $\mathbf{y}_c^p$ as a boundary point is shown as Δ and given by

$$\Delta = \|\mathbf{y}_c^p - \mathbf{x}_{c-1}^p\|. \tag{20}$$

We hence define the *layer thickness likelihood*, $L_4(\mathbf{y}_c^p)$, as

$$\begin{aligned} L_4(\mathbf{y}_c^p) &= \log\left[\frac{1}{\sigma_c\sqrt{2\pi}}\exp\left(\frac{-(\Delta-\mu_c)^2}{2\sigma_c^2}\right)\right] \\ &= \log\big[\mathcal{N}(\Delta\,;\,\mu_c,\sigma_c)\big] \end{aligned} \tag{21}$$

where the quantity $\|\mathbf{y}_c^p - \mathbf{x}_{c-1}^p\|$ (Δ in the figure) is just the distance between $\mathbf{y}_c^p$, the point under consideration, and its neighbor on the inner contour (or the origin in case $c = 1$). Note that we will henceforth use the notation $\mathcal{N}(x\,;\,\mu,\sigma)$ to represent the normal distribution with parameters μ and σ evaluated at x.

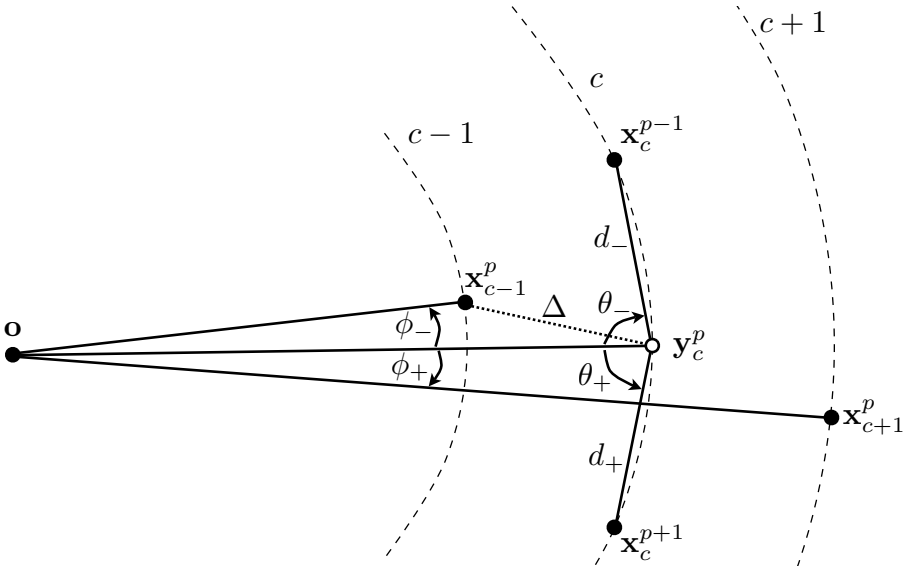

Figure 6. Quantities used when computing shape-based likelihoods while evaluating $\mathbf{y}_c^p$ for selection as point p on contour c as described in Section 3.2. The particle center, $\mathbf{o}$, serves as the coordinate system origin and is computed by fitting a circle[8] to the innermost (kernel) contour.

We next consider the angles θ_- and θ_+ defined by

$$\theta_- = \measuredangle(-\mathbf{y}_c^p,\, \mathbf{x}_c^{p-1} - \mathbf{y}_c^p) \tag{22}$$

$$\theta_+ = \measuredangle(-\mathbf{y}_c^p,\, \mathbf{x}_c^{p+1} - \mathbf{y}_c^p) \tag{23}$$

For a circular contour, these angles would each equal $\mu_\theta = (N-2)\pi/(2N)$.[6] We hence define the *neighbor angles likelihood*, $L_5(\mathbf{y}_c^p)$, to be

$$L_5(\mathbf{y}_c^p) = \log\big[\mathcal{N}(\theta_-\,;\,\mu_\theta,\sigma_\theta)\big] + \log\big[\mathcal{N}(\theta_+\,;\,\mu_\theta,\sigma_\theta)\big]. \tag{24}$$

We also desire our points to be approximately equally spaced around the closed contour. We express this using the distances to the neighbors defined by

$$d_- = \|\mathbf{y}_c^p - \mathbf{x}_c^{p-1}\| \tag{25}$$

$$d_+ = \|\mathbf{y}_c^p - \mathbf{x}_c^{p+1}\|, \tag{26}$$

which should be approximately equal. This leads us to define the *neighbor distances likelihood*, $L_6(\mathbf{y}_c^p)$, as

$$L_6(\mathbf{y}_c^p) = \log\big[\mathcal{N}(d_+ - d_- \,;\, 0, \sigma_d)\big]. \tag{27}$$

Finally, we expect our contours to be approximately circular and cocentric; hence we desire corresponding points on different contours to lie approximately on the same radial arcs. This can be quantified by considering the angles ϕ_- and ϕ_+ defined by

$$\phi_- = \measuredangle(\mathbf{y}_c^p,\, \mathbf{x}_{c-1}^p) \tag{28}$$

$$\phi_+ = \measuredangle(\mathbf{y}_c^p,\, \mathbf{x}_{c+1}^p), \tag{29}$$

which we expect to be close to zero. We hence define the *layer angles likelihood*, $L_7(\mathbf{y}_c^p)$, by

$$L_7(\mathbf{y}_c^p) = \log\big[\mathcal{N}(\phi_- \,;\, 0, \sigma_\phi)\big] + \log\big[\mathcal{N}(\phi_+ \,;\, 0, \sigma_\phi)\big]. \tag{30}$$

3.3. Iteration and Convergence

Given the likelihood terms described above, we define the net likelihood for $\mathbf{y}_c^p$ to be

$$\bar{L}(\mathbf{y}_\mathbf{c}^\mathbf{p}) = \sum_{\{l\}} L_l(\mathbf{y}_c^p), \tag{31}$$

Initially, we use all of the criteria defined above, implying $\{l\} = \{1, \ldots, 7\}$. After initial convergence, however, we switch to a subset of the criteria (for contours $c > 1$), using only $\{l\} = \{1, 6\}$ corresponding to the edge and neighbor distance likelihoods defined in Eq. (17) and Eq. (27), respectively. This approach allows us to get close to the true boundaries initially, and then adapt to any significant variations from circularity. However, due to the unpredictable nature of the kernel-to-buffer transition ($c = 1$), we continue to use the complete set of criteria for $c = 1$ even after initial convergence.

Our ultimate goal is to find the set of all contour points, $\{\mathbf{x}_c^p\}$ that maximizes the total likelihood, i.e., we seek to solve

$$\{\mathbf{x}_c^p\} = \arg\max_{\{\mathbf{y}_c^p\}} \sum_{c=1}^{C} \sum_{p=1}^{N} \bar{L}(\mathbf{y}_c^p). \tag{32}$$

We adopt a simple greedy approach for this problem, using the convergence criterion defined below. Beginning with the innermost contour, $c = 1$, we examine each point of each contour assuming all other points on all other contours are fixed. For the contour point under consideration, we evaluate the net likelihoods in a region surrounding the current contour point and select the maximum likelihood point in this region as the new contour point. We then proceed to the next point in that contour, and then to next contour, and so on until all points on all contours have been evaluated, constituting a single iteration. We repeat this process until initial convergence, at which point we then reduce our likelihood computations to a more local subset (as discussed above) and also reduce the size of our search region around each point. We then restart the iterative process and continue until final convergence.

We determine convergence as follows. Assume that, at iteration k, the likelihood (or probability) for some point n in our complete set of NC contour points (N points on each of C contours) is given by P_n^k. Then the total log likelihood at iteration k can be expressed

$$\mathcal{L}(k) = \sum_{n=1}^{NC} \log(\mathrm{P}_n^k). \tag{33}$$

We would like to terminate the iteration when the total likelihood from one iteration to the next does not grow appreciably. Suppose for example, we wish to stop iterating when the likelihood for every point increases on average by less than a factor of $(1 + \delta)$. An increase by this factor can be expressed as follows:

$$\begin{aligned} \mathcal{L}(k+1) &= \sum_{n=1}^{NC} \log\big[(1+\delta)\mathrm{P}_n(k)\big] = NC\log(1+\delta) + \sum_{n=1}^{NC} \log\big[\mathrm{P}_n(k)\big] \\ &= NC\log(1+\delta) + \mathcal{L}(k) \end{aligned} \tag{34}$$

This leads us to define the termination criteria as:

$$\mathcal{L}(k+1) - \mathcal{L}(k) < NC \log(1+\delta). \tag{35}$$

where $\delta \ll 1$ is a user-defined parameter.

4. RESULTS

Here we show experimental results using the particle images shown earlier in Fig. 1. These images represent an area of 1.09 mm × 1.09 mm; they are 512 × 512 pixels, obtained by reducing the square center of original 3900 × 3090 pixel images. We use $C = 5$ contours, each with $N = 48$ points. For EdgeFlow computation, we set $s = 1.5$ and hence $d = 6$. From design specifications, we know that the mean layer thicknesses and approximate standard deviations (in pixels) are $\mu_c = \{84, 50, 18, 16, 18\}$ and $\sigma_c = \{5, 6, 2, 2, 2\}$, respectively; these parameters are used in computing the layer thickness likelihood term defined in Eq. (21). Recalling that the intensity criteria defined in Section 3.1 are dependent upon a scale parameter s, we set s to be (the ceiling of) 0.125 times the minimum mean layer thickness; in this case, $s = 2$. For the neighbor angle likelihood from Eq. (24), recall that $\mu_\theta = (N-2)\pi/(2N)$; we use $\sigma_\theta = 0.2\mu_\theta$. For the neighbor distance likelihood from Eq. (27), we use $\sigma_d = 0.125 \times 2r_c \sin(\pi/N)$, which is 0.125 times the distance between adjacent, equally spaced points around a circle of radius r_c, where r_c is the expected radius of contour c computed from the mean layer thicknesses. For the layer angle likelihood term from Eq. (30), we use $\sigma_\phi = \pi/64$. Search regions around each contour point are set to an annular wedge that extends radially inward and outward by 1.5 times the minimum mean layer thickness (24 pixels) until initial convergence and then 0.25 times the minimum mean layer thickness (4 pixels) until final convergence. For the convergence criterion from Eq. (35), we use $\delta = 0.001$. Using these parameters and the initializations of Fig. 7, the final results are shown in Fig. 8. From left-to-right in Fig. 8, the numbers of iterations until final convergence were 14, 12, and 11. In Fig. 9, we show a more detailed view of the original image, the initialization, and the final results for the left image of the previous figures.

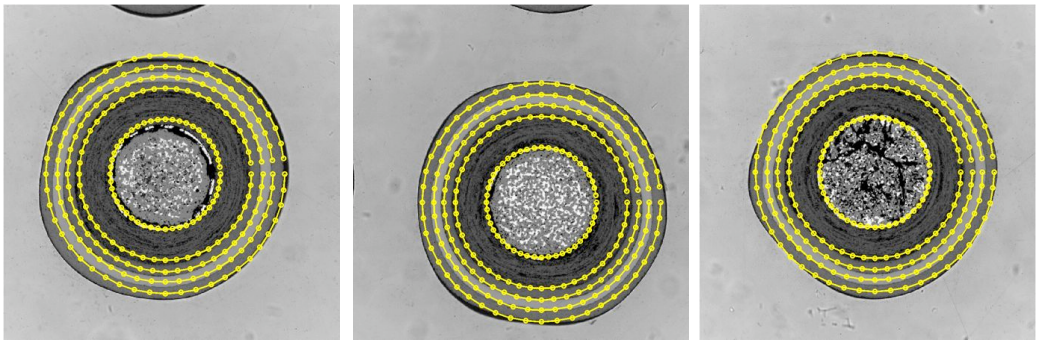

Figure 7. Initial contours for the images from Fig. 1 determined using the approach described in Section 2.3.

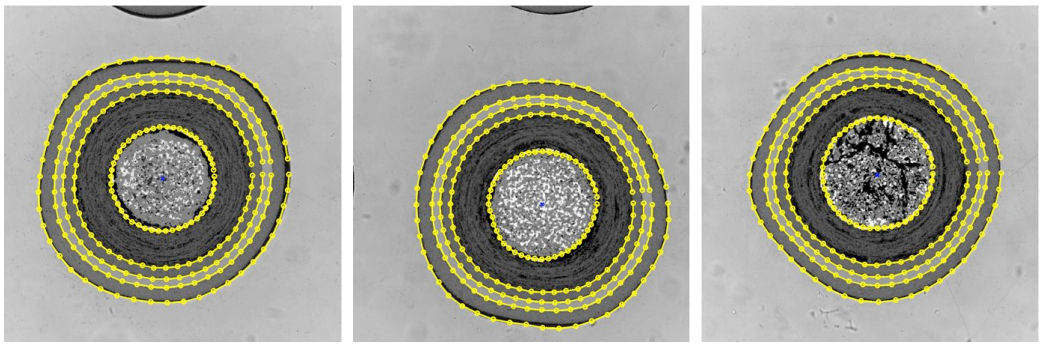

Figure 8. Final results after application of active contour model described in Section 3 using initializations from Fig. 7.

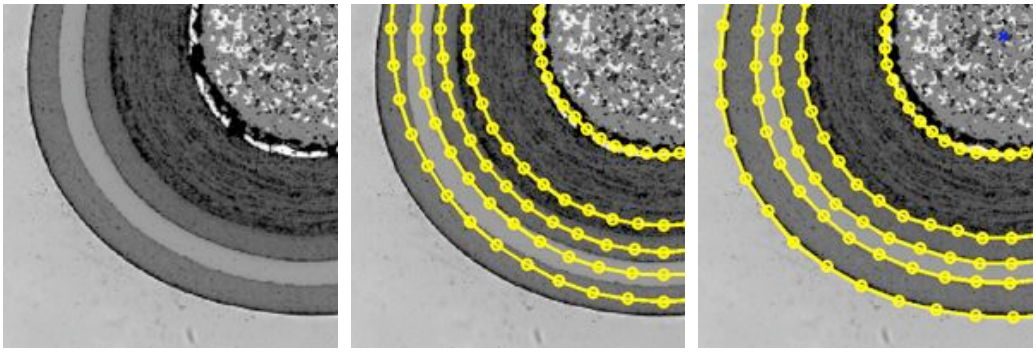

Figure 9. Detailed view of the original image (left), initialization (middle), and final results (right) for the left image of the previous figures.

5. CONCLUSIONS

In this paper, we presented a multiple active contour model for locating boundaries in cross-sectional images of coated particle nuclear fuel. For initialization, we employed intensity EdgeFlow as a robust edge detector followed by template-based estimation of the particle center. We described how EdgeFlow can be implemented efficiently using steerability and Fourier properties. We presented a Bayesian motivated strategy for a multiple snake model. Greedy optimization was employed to evolve the contours towards the maximum likelihood solution of the intensity and shape criteria, including *a priori* information from design specifications and historical analysis. Experimental results were shown that demonstrated the performance of the proposed techniques.

ACKNOWLEDGMENTS

Prepared by Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

1. J. R. Price and J. D. Hunn, "Optical inspection of coated particle nuclear fuel," in *Machine Vision Applications in Industrial Inspection XII*, **5303**, pp. 137–149, 2004.
2. J. Price, D. Aykac, J. Hunn, A. Kercher, and R. Morris, "New developments in image-based characterization of coated-particle nuclear fuel," in *Machine Vision Applications in Industrial Inspection XIV*, **6070**, pp. 153–162, 2006.
3. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision* **1**(4), pp. 321–331, 1987.
4. W.-Y. Ma and B. Manjunath, "EdgeFlow: A technique for boundary detection and image segmentation," *IEEE Transactions on Image Processing* **9**(8), pp. 1375–1388, 2000.
5. W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(9), pp. 891–906, 1991.
6. S. Gunn and M. Nixon, "A robust snake implementation; a dual active contour model," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(1), pp. 63–68, 1997.
7. T. Abe and Y. Matsuzawa, "A region extraction method using multiple active contour models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, **1**, pp. 64–69, 2000.
8. I. Kasa, "A circle fitting procedure and its error analysis," *IEEE Transactions on Instrumentation and Measurement* **25**, pp. 8–14, March 1976.