Seed-Point Detection of Clumped Convex Objects by Short-Range Attractive Long-Range Repulsive Particle Clustering

James Kapaldo, Xu Han, and Domingo Mery, *Member, IEEE*

Abstract—Locating the center of convex objects is important in both image processing and unsupervised machine learning/data clustering fields. The automated analysis of biological images uses both of these fields for locating cell nuclei and for discovering new biological effects or cell phenotypes. In this work, we develop a novel clustering method for locating the centers of overlapping convex objects by modeling particles that interact by a short-range attractive and long-range repulsive potential and are confined to a potential well created from the data. We apply this method to locating the centers of clumped nuclei in cultured cells, where we show that it results in a significant improvement over existing methods (8.2% in F_1 score); and we apply it to unsupervised learning on a difficult data set that has rare classes without local density maxima, and show it is able to well locate cluster centers when other clustering techniques fail.

Index Terms—seed-point detection; data mining; nuclei de-clumping; short-range attractive long-range repulsive

1 Introduction

Locating object and distribution centers is a fundamental step of both image processing and unsupervised machine learning or exploratory data mining. The automated analysis and classification of cells from microscopy images is an important field that includes both of these disciplines [1], [2], [3], [4], [5], [6]. In this field, the first step is often locating nuclei centers using image processing techniques, which allows for seed-point based segmentation algorithms [7], [8], [9], [10] to determine the nuclei boundaries. Using the segmentation results, features of each nucleus or cell are measured and used to classify the cells into different groups (e.g. cancer/non-cancer, different phenotypes/cell-phase [5], [11], [12], [13], [14], [15]). When searching for new groups (new phenotypes, new effects, ...) or when it not possible for an expert to create a labeled data set, unsupervised machine learning, where groups are determined by looking for similarities in the data, must be used.

Locating nuclei centers and using unsupervised learning to locate cluster centers are, abstractly, very similar: they both try to locate the centers of partially-overlapping convex objects or distributions. However, the techniques for solving each of these problems are quite different and are not ideal for several reasons. The current methods for locating nuclei centers (radial voting [10], [16], [17], [18], gradient convergence and sliding band filters [19], [20], Laplacian of Gaussian filters [21], [22], [23], [24]) try to compute a surface, named the *voting landscape*, that has local extrema at the nuclei centers. The fundamental problem with these methods is that the extrema of the voting landscapes are not at the nuclei centers; further, there can be extra extrema, where false nuclei are detected, or missing extrema, where true nuclei are not detected. Since the segmentation methods will return one region per seed-point, it is critical that each nuclei have only one seed-point as close as possible to the nuclei center [10], [22]. For unsupervised learning, there are many well known clustering techniques (K-means/fuzzy C-means [25], [26], [27], expectation maximization/mixture of Gaussians [28], hierarchical clustering [29], mean-shift [30]), but a general drawback of most of these methods is that they produce cluster centers that are biased towards high density regions in the data, and many of them require the number of clusters (which is normally not known beforehand) as an input. Mean-shift clustering can be better in this regard; however, each cluster must have a local maximum in the point density.

In this work, we develop a new clustering method (*SALR clustering*) that can be used for locating both nuclei centers and the cluster centers in scatter point data, while at the same time addressing the above issues. The premise of our method, which is graphically described in Figure 1 and which takes cues from the fundamental physics of modeling classical Wigner crystals [31] and modeling the formation of clusters [32], [33], is that we can find the centers of overlapping convex regions by modeling the dynamics of particles trapped in an appropriate confining well.

Consider a set of repulsive particles confined to a region; these particles will try to move as far apart from each other as possible. Figure 1a shows this for 10 particles confined to a parabolic potential well. If the region the particles are confined to has several local minima, then the particles will preferentially locate as close as possible to these minima to lower the total energy. Assuming our goal is to have particles near each local minima and nowhere else, we

- *J. Kapaldo and X. Han are with the Department of Physics, University of Notre Dame, Notre Dame, IN, 46545. (e-mail: jkapaldo@nd.edu, xhan1@nd.edu)*
- *D. Mery is with the Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile. (email: dmery@ing.puc.cl)*

Manuscript submitted Aug. 26, 2017.

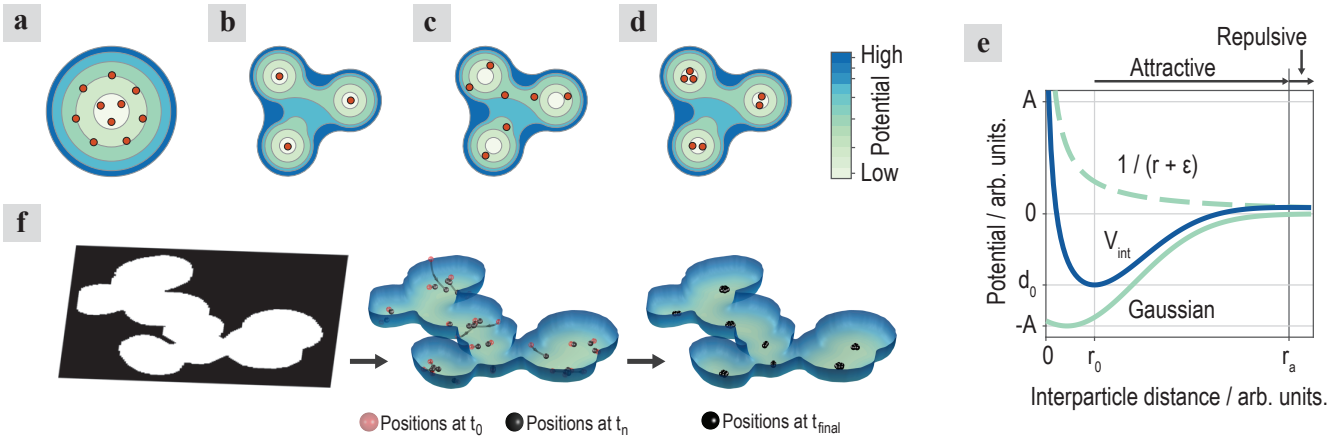


Fig. 1. Premise of SALR clustering. **a–d**, Intuition: schematic of the lowest energy configuration for **a**, 10 repulsive particles in a parabolic well (this is one of the two isomeric ground states [31]); **b**, three repulsive particles with three potential minima; **c**, seven repulsive particles with three potential minima; and **d**, seven SALR interacting particles with three potential minima. Contour levels represent potential energy. **e**, SALR interaction potential formed with a Gaussian attractive term and a $1/r$ repulsive term. The distance r_a, named the attractive extent, is the distance where particles switch from being attractive to repulsive. **f**, Starting from a region of overlapping convex objects, a confining potential is created (third dimension corresponds to potential value); particles are randomly placed and the dynamics modeled; at the end the particles will form clusters located approximately at the center of each convex object.

must have the same number of particles as the number of potential minima, since two particles cannot be near each other, see Figure 1b,c. This is a problem as the number of potential minima is normally not known before hand, but it can be solved by modifying how the particles interact with each other such that particles near each other are attracted to each other instead of repulsed—this short-range attractive long-range repulsive (SALR) particle interaction potential can be seen in Figure 1e. Now, we can use (many) more particles than we expect there to be potential minima, and we can have a cluster of particles located at each potential minima, see Figure 1d.

Using this premise, we create a confining potential from the region of overlapping convex objects that has a higher energy at the region's edges and a lower energy (valley) near the object centers, we randomly place many SALR particles in this region, and then we model the particle dynamics while slowly decreasing their speed; the final position of each cluster of particles will approximately give the center of each locally convex region, see Figure 1f and Supplemental Video 1.

We validate SALR clustering by applying it to the problem of locating nuclei centers, and show that it can significantly improve the location performance compared with previous methods and discuss how/why it is able to improve the performance. We also demonstrate the application of SALR clustering to unsupervised machine learning, and show that it can determine the correct number and position of clusters and better locate rare clusters that do not have local density maxima.

2 Modeling Particle Dynamics

The equations governing the particle dynamics are derived from the Hamiltonian of a set of N interacting charged particles,

$$\mathcal{H} = \sum_i \Big(\frac{1}{2m_i} \mathbf{p}_i^2 + V(\mathbf{r}_i) \Big) + \sum_{\substack{i \\ j \neq i}} \frac{k q_i q_j}{2} V_{\text{int}}\big(|\mathbf{r}_i - \mathbf{r}_j|\big), \quad (1)$$

where m, $\mathbf{p}$, $\mathbf{r}$, and q represent the particle's mass, momentum, position, and charge, respectfully, k is a coupling constant, and $|\mathbf{r}_i - \mathbf{r}_j|$ is the distance between $\mathbf{r}_i$ and $\mathbf{r}_j$. $V_{\text{int}}(\cdot)$ is the SALR particle interaction potential created with a Gaussian attractive term and a $1/r$ repulsive term

$$V_{\text{int}}(r) = \frac{1}{r + \epsilon} - A \exp\left(-\frac{(r - \mu)^2}{2\sigma^2} \right), \quad (2)$$

where r is the distance between two of the particles and ϵ is a small constant value ($\epsilon = 0.2$) to prevent the divergence at $r = 0$. An example of this potential is shown in Figure 1e. The values A, μ, and σ in (2) are set by solving a least-squares problem so that the depth d_0, the location of the potential minimum r_0, and the distance at which the potential goes from being attractive to repulsive, referred to as the attractive extent, r_a may be directly specified, see Figure 1e. For example, the correspondence between two sets of these parameters $(d_0, r_0, r_a) \rightarrow (A, \mu, \sigma)$ are $(-1, 2, 10) \rightarrow (1.58, 0.87, 2.82)$ and $(-1, 2, 15) \rightarrow (1.84, -1.32, 4.84)$.

The particles' equations of motion can be found from the Hamiltonian (1) to be

$$\frac{d\mathbf{p}_i}{dt} = -\boldsymbol{\nabla}_i V(\mathbf{r}_i) - \sum_{j \neq i} k q_i q_j \boldsymbol{\nabla}_i V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) \quad (3)$$

$$\frac{d\mathbf{r}_i}{dt} = \frac{1}{m_i} \mathbf{p}_i, \quad (4)$$

where $\boldsymbol{\nabla}_i$ is the gradient operator acting on vector $\mathbf{r}_i$.

The system described so far conserves energy, which means that the particles/clusters will never stop moving. To find the lowest (or a low lying) energy state of the

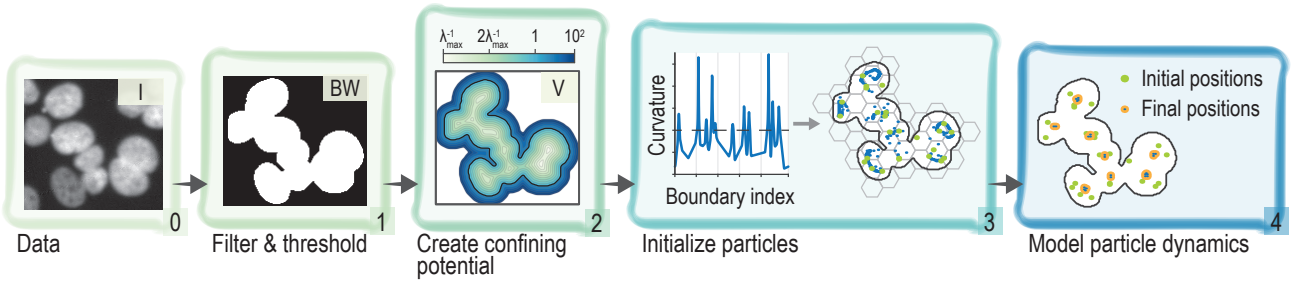


Fig. 2. Locating nuclei centers method. 1) create a binary mask by filtering and thresholding the image. 2) create a confining potential (a 3D version of this potential is shown in Figure 1f). 3) initialize particle positions: create possible set of points (blue dots) based on the boundary curvature and choose one point per grid element of an overlaid grid (green points). 4) model the particle dynamics.

system, we slowly decrease the temperature of the system by damping the particles:

$$\frac{d\mathbf{p}_i}{dt} = -\boldsymbol{\nabla}_i V(\mathbf{r}_i) - \frac{\alpha(t)}{m_i}\mathbf{p}_i - \sum_{j\neq i} kq_iq_j\boldsymbol{\nabla}_i V_{\text{int}}(|\mathbf{r}_i-\mathbf{r}_j|), \quad (5)$$

where $\alpha(t)$ is a time dependent damping constant. $\alpha(t)$ is not required to be time dependent, but faster convergence can be achieved by increasing the value of α as the simulation goes on, e.g. $\alpha(t) \propto t$.

Using (4) and (5) we can model the dynamics of the particles until the solution converges. We solve the set of $2N$ differential equations using a Runge-Kutta (2,3) solver [34] (implemented with Matlab), which provides high enough tolerance to solve the problem and is faster than higher order methods. Convergence is determined by looking at the average particle speeds over a short time span (the last few time steps of the ODE solver), and if this average speed is below some threshold, then we stop the simulation. Averaging over a time span is used because the clusters of particles can have vibrational or rotational modes [35] that take a long time to die out (perhaps 30% more computation time), and using the average speed allows these modes to be effectively ignored.

In **??**, we make connections between our optimization method and simulated annealing and gradient descent that could lead to faster convergence.

3 Locating Nuclei Centers

3.1 Method

The process we developed for applying our method to locating nuclei centers is shown in Figure 2. We go through each step below, and recommend viewing Supplemental Video 1 for a visual description.

3.1.1 Filter & Threshold

After reading in the data, the (overlapping) foreground objects should be separated from the background with the goal of creating a binary mask $BW(\mathbf{r})$ such that $BW(\mathbf{r}) = 1$ for all $\mathbf{r}$ in an object and $BW(\mathbf{r}) = 0$ for all $\mathbf{r}$ not in an object. For the nuclei images used in this work, we filtered the images with a Gaussian blur ($\sigma = 1$) and then used an adaptive log-weighted Otsu threshold [36].

3.1.2 Construct confining potential

We use the binary mask to create a confining potential well, $V(\mathbf{r})$. Let the set of coordinates belonging to the object be denoted by $\Omega = \{\mathbf{r} \mid BW(\mathbf{r}) = 1\}$. The confining potential is given by

$$V(\mathbf{r}) = \begin{cases} \text{dt}(\neg BW)^{-1}, & \forall\, \mathbf{r} \in \Omega \\ \text{dt}(BW)^2 + 1, & \forall\, \mathbf{r} \notin \Omega \end{cases} \quad (6)$$

where $\text{dt}(\cdot)$ is the distance transform, and $\neg$ is the negation operator ($0 \leftrightarrow 1$). The distance transform takes a binary image and assigns to each pixel the Euclidean distance to the nearest non-zero pixel in the binary image. The first line of (6) states that the potential inside the object is given by 1 over the distance to the nearest object boundary pixel, and the second line states that the confining potential increases quadratically outside of the object. Note that the potential is < 1 inside the object and > 1 outside. After forming $V(\mathbf{r})$, it is smoothed with a Gaussian with $\sigma = 1$ pixel.

This confining potential is not scale invariant: as a object becomes larger, the potential will become more *flat-bottomed*, which will cause the particles to push each other closer to the object boundary, as is shown in **??**a. Scale invariance can be added by implicitly scaling all objects so that their maximum distance transform values, $\lambda = \max(\text{dt}(\neg BW))$, are equal to the same fixed value, $\lambda_{\max}$ (see **??**b). This changes the confining potential on the interior of the object to be

$$V_{\mathbf{r}\in\Omega} = \left[1 + \frac{\lambda_{\max} - 1}{\lambda - 1}\Big(\text{dt}(\neg BW) - 1\Big)\right]^{-1}. \quad (7)$$

Note that for a single ellipsoidal object, λ represents the semi-minor axis of the object; scaling the distance transform effectively means we are resizing each object to have the same semi-minor axis, $\lambda_{\max}$.

3.1.3 Particle initialization

The next step is initializing the particles: determine the density/number of particles to use, set their mass and charge, and set their initial velocities and positions.

NUMBER: The number of particles is determined by setting the particle density. The parameter we use to set the particle density is the Wigner-Seitz radius r_s, which

describes the amount of space taken up by each particle. The number of particles N, in 2D, is then approximately

$$N = \frac{A}{\pi r_s^2}, \tag{8}$$

where A is the area of the object (the number of elements in set Ω) and πr_s^2 is the amount of space per particle.

MASS & CHARGE: We set the mass of the particles to one and the particle charge to decrease with N as $q = N^{-\beta}$. Larger values of β result in particle clusters that interact with each other less and can move around more, smaller values of β result in particle clusters that strongly interact and push each other closer to the object boundary.

VELOCITY: The particles are given random initial velocities according to $\mathbf{v}_0 = s_0\hat{\mathbf{u}}$, were s_0 is the initial speed and $\hat{\mathbf{u}}$ is a random unit vector.

POSITION: There are several possible methods of selecting the particles' initial positions; we describe three methods which would each have there own benefit depending on the data.

i) Random: The most simple method, choose N random points out of the set Ω.

ii) Uniform random: To ensure we can find all of the seed-points, the particles' initial locations should uniformly cover the domain of interest (Ω). This can be accomplished by overlaying a lattice across the domain and placing one particle in each lattice unit cell. In order to have approximately N particles, the lattice constant is chosen such that the lattice unit cell's area is approximately the size of particle, πr_s^2 in 2D. In 2D, we use a hexagonal lattice with lattice constant $2r_s$, and in higher dimensions we use a simple (hyper)cubic lattice. The location of the particle within each lattice cell is selected by choosing one random point from the points in Ω that are also in each lattice cell, $\Omega \cap \mathrm{U}_i$, where U_i is the set of all coordinates in the i'th lattice cell.

iii) Convex hull transformed center of curvature (CvxHll CoC): Better results can be obtained if we can choose points that are approximately where the true seed-points are expected. For this, we compute a set of possible positions, R_0, using the centers of curvature of each boundary vertex of the region Ω, see **??**. For each lattice cell, we then choose one random point from the set, $\mathrm{R}_0 \cap \mathrm{U}_i$. Note that the centers of curvature can be far from the geometric center of elliptical regions since the long side of an ellipse has a smaller curvature (larger radius). This is addressed by modifying the boundary curvature to be the convex hull of each negative curvature region (See **??** for an example of this method.)

As a final note on position initialization, the confining potential can be used to tighten the constraint on where particles can be located. At the least, we require all initial positions to have a confining potential value less than 1 (the particles must start inside the object); however, we can further require that the confining potential value at all initial positions be in some range $[V_{\min}, V_{\max}]$, which can be useful for both improving control over initial positions and reducing the number of particles used in the simulation.

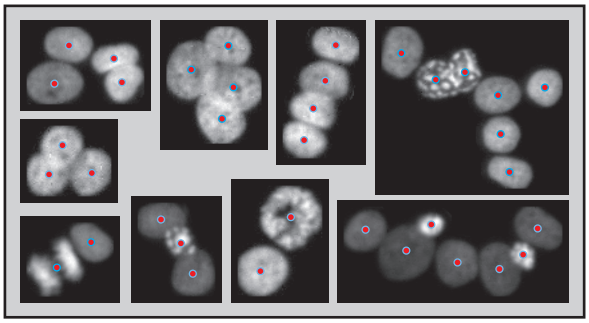


Fig. 3. Example nuclei clumps. Examples of the nuclei clumps used for validation; the contrast for each object has been enhanced. The red dots represent the labeled nuclei centers, and the edge of the markers represent our expected confidence in their location (~ 3 pixel radius).

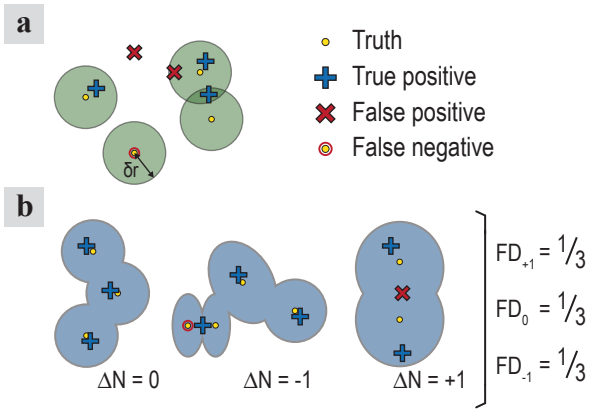


Fig. 4. Comparison metrics. **a**, Definition of true positive and false positive calculated seed-points and false negative seed-points. **b**, Example showing the calculation of the fractional distribution $\mathrm{FD}_{\Delta \mathrm{N}}$ measure for three objects: one with the correct number (left), one with one too few (middle), and one with one too many (right) computed seed-points.

3.1.4 *Model particle dynamics*

The particles are modeled using (4) and (5) until the solution converges. Box 4 in Figure 2a shows the particle positions after convergence; you can see clusters of particles are located at each of the nuclei centers. After the model has converged, we extract the center of each cluster by grouping together all particles within some distance of each other $(0.7\, r_0 + 0.3\, r_a)$ and then finding the mean location of each group. (This distance is empirically set and could be any distance $< r_a$ and $> r_0$.) These cluster centers are the final seed-points returned by our method.

3.1.5 *Reduction of work*

It is not necessary to model the particle dynamics on all objects; if an object is convex, or it is smaller than a particle (πr_s^2), or only one particle would be modeled, then we do not run the simulation, but simply return the centroid of the object as the seed-point. We determine if an object is convex by testing that the all boundary curvature is less than some threshold, $0.25/\lambda$.

3.2 Results

The data we use for validating our method is a set of 2,420 images of nuclei clumps with a total of 7,789 nuclei. The

cells are epithelial cancer cells from the tongue (squamous cell carcinoma, SCC25) that we cultured, stained with 4',6-diamidino-2-phenylindole (DAPI) to label their DNA, and then had imaged using a whole slide fluorescence reader, see **??** for more details. The center of each of these 7,789 nuclei were manually labeled to create the truth data. Figure 3 shows an example of nine nuclei clumps with the labeled nuclei centers (more examples can be seen in **??**).

We use two metrics to determine performance of nuclei center location. The first is the F_1 score, which ranges between 0 and 1 and is the harmonic mean of the precision and recall; a F_1 value of 1 means that all seed-points (nuclei centers) were calculated perfectly, and a F_1 value of 0 means not a single seed-point was calculated correctly. It is computed as

$$F_1 = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{9}$$

where TP, FN, and FP are the number of true positives, false negatives, and false positives, respectively. A graphical definition of these terms is shown in Figure 4a: a calculated seed-point is TP if it is within a distance of δr to a truth point and it is the closest calculated seed-point to that truth point, otherwise it is FP; a FN point is a truth point that does not have a calculated seed-point assigned to it. For example, the rightmost red X in Figure 4a is FP because there is another computed seed-point closer to the truth location than it.

The second metric we use measures the fractional distribution, $\mathrm{FD}_{\Delta \mathrm{N}}$, of objects (nuclei clumps) with the correct, too many, or too few calculated seed-points. For example, if $\mathrm{FD}_{-1} = 0.1$, then 10% of all the objects have one less calculated seed-point than the true number. This metric is important as it directly measures the ability of a method to calculate the correct number of seed-points, even if the seed-points are not in the correct location. The fractional distribution $\mathrm{FD}_{\Delta \mathrm{N}}$ of ΔN is given by

$$\mathrm{FD}_{\Delta \mathrm{N}} = \frac{1}{M} \sum_{i=1}^{M} \delta_{\Delta \mathrm{N},\, (\mathrm{FP}_i - \mathrm{FN}_i)} \tag{10}$$

where M is the number of objects (2,420), $\delta_{\cdot,\cdot}$ is the Kronecker delta, and FP_i and FN_i are the number of false positives and the number of false negatives for the i'th object. An example calculating $\mathrm{FD}_{\Delta \mathrm{N}}$ is shown in Figure 4b. Depending on how the seed-points will be used, either F_1 or $\mathrm{FD}_{\Delta \mathrm{N}}$ could be more important.

Using these two metrics, we compare the results of our method to seven other standard and leading methods for locating nuclei centers: multi-pass voting (MPV) by [16], two versions of single-pass voting by [10] ($\mathrm{SPV}_{\mathrm{Qi}}$) and by [18] ($\mathrm{SPV}_{\mathrm{Xu}}$), sliding-band filter (SBF) by [19], generalized Laplacian of Gaussian (gLoG) by [24], maximally stable extremal regions (MSER) by [37], and directly using the local maximum of the distance transform (Dist. Trans.). We optimized the parameters of each method to maximize the sum of $F_{1,\,\delta r=3}$ and FD_0; descriptions of the methods along with details on the parameters we optimized over and the final parameters used for each method can be seen in **??**. The parameters used for our method are shown in Table 1. Many of these parameters are set empirically, and several were set by explicit optimization and are discussed in **??**.

TABLE 1
Model parameter values used for our method in computing the results of Figure 5.

Parameter name	Symbol	Default value
Particle initialization		CvxHll CoC
Wigner-Seitz radius	r_s	5
Confining potential depth	$\lambda_{\max}$	18
V_{int} attractive extent	r_a	13
V_{int} minimum location	r_0	2
V_{int} depth	d_0	-1
Max init. particle potential	$V_{\max}$	1/5
Charge normalization	β	1/3
Damping rate	$\alpha(t)$	$5 \cdot 10^{-4}\, t$
Initial speed	s_0	0.01
Mass	m	1
Coupling constant	k	1

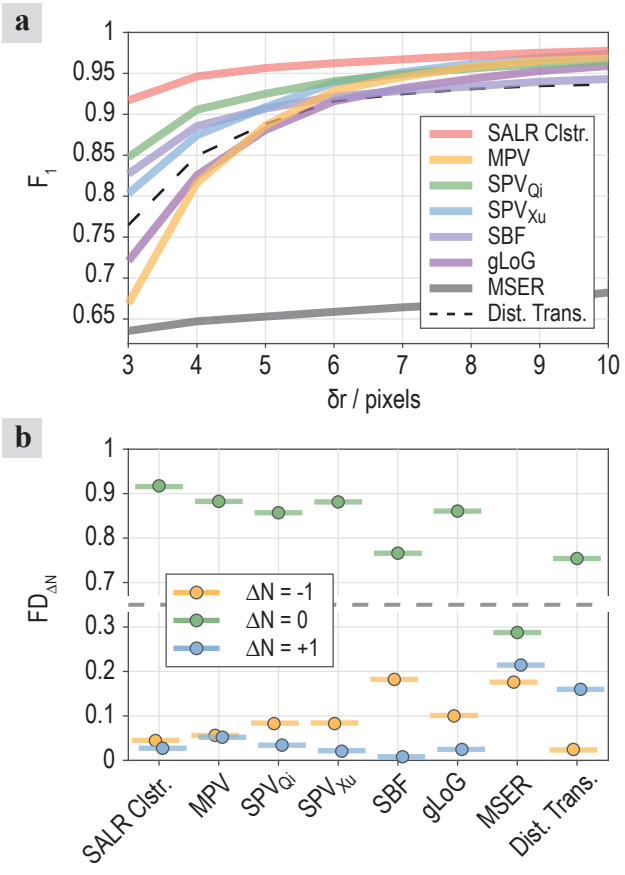


Fig. 5. Methods comparison. **a**, F_1 versus δr for the seven methods. **b**, $\mathrm{FD}_{\Delta \mathrm{N}}$ for $\Delta N = \{-1, 0, +1\}$ for the seven methods.

We show the results of the comparison in Figure 5. Our method (SALR Clstr.) has both the best F_1 score (0.069 higher, 8.2%, than the next best method at $F_{1,\,\delta r=3}$ and 0.028 higher, 3.0%, than the next best average F_1 score) and the best FD_0 value (0.033 higher, 3.8%, than the next best method).

In particular, note that our method does much better than the distance transform; this is important because our confining potential is proportional to one over the distance transform. This means our method is not simply locating the local maxima of the distance transform, see Discussion.

Additional validation of our method's scale invariance

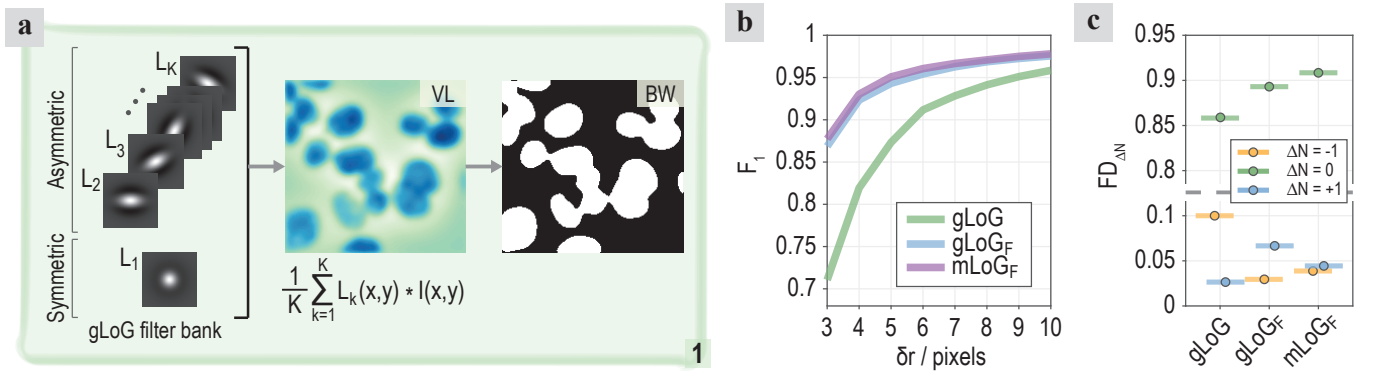


Fig. 6. Advanced filtering. **a**, *Filter & threshold* step of method: convolve a gLoG filter bank with an image to create a voting landscape, then threshold to create the binary mask. **b**, F_1 score using the previous gLoG method (shown in Fig. 5), using the entire gLoG filter bank (gLoG_F), and using only the symmetric gLoG filter (gLoG_SF). **c**, $\text{FD}_{\Delta\text{N}}$ results for the same three methods.

as well as the performance and computation time dependence on the initial particles starting locations and density are discussed/shown in **??**/**??** and **??**/**??**.

3.3 Advanced filtering

The previous methods we compared to all use a voting landscape to determine the nuclei centers. We surmised that thresholding their voting landscape to create a binary mask and then using our method, instead of using the local maxima of the voting landscape, could improve the performance of the methods. We tried this using LoG filtering as it is likely the most easy to implement as well as the fastest.

In Figure 6a we show the *Filter & Threshold* step using gLoG. A gLoG filter bank is created that consists of a symmetric multiscale LoG filter (mLoG) and several asymmetric multiscale LoG filters. The voting landscape is formed by summing the convolution of each of these filters with the input image, and the binary mask is created with a simple threshold. In Figure 6b and 6c we show the F_1 and $\text{FD}_{\Delta\text{N}}$ results of the original gLoG method (the same results as shown in Figure 5) and the results of using gLoG as the filter for our method (gLoG_F). You can see that using gLoG with our method results in a significant improvement in $F_{1,\,\delta r=3}$, by $\sim$ 0.15, and a large improvement in FD_0, by $\sim$ 4%. Additionally, using only the symmetric mLoG as the filter for our method (mLoG_F) results in even better performance than the gLoG_F, with about a 2% increase in FD_0.

These results are important because much research does not use images where the nuclei can be easily segmented by thresholding (like the cultured cells in this work), but use colored histopathological (tissue) images, which are not easily thresholded. Our results show that the nuclei regions can be detected using gLoG filters and then accurate nuclei center locations and accurate nuclei count can be obtained by using our method.

4 Application to Data Clustering

In unsupervised machine learning and data mining, data often comes in the form of scatter point data where each point is an observation and each dimension is a different measured quantity. An example of 2D scatter point data is shown in the top of Figure 7a. The easiest way to apply our seed-point detection method to this type of data is by *binning* the data, where a grid is overlaid on the data and the number of points in each bin is counted. The result, shown in the bottom of Figure 7a, can be thought of as an image, and we can threshold it to create a binary mask and then proceed as before.

4.1 Simple 3D data

We first validate the use of our method for detecting the cluster centers of scatter point data by using a simple 3D distribution where we can use the well known k-means clustering. The point-count isosurfaces of the 3D distribution can be seen along with the projection of the isosurfaces to the 2D axis planes in Figure 7b. We first scale the z-axis so that the objects are approximately the same size along any direction (see **??**), we then threshold the point-count and create the confining potential, which is shown in Figure 7c.

We initialize particles for the simulation using a uniform random distribution with $r_s \approx 9$ (resulting in 79 particles); all parameters other than particle density are the *same* as we used in the 2D images above (values shown in Table 1). We simulated the particles 20 times (each time with new initial positions) and show the computed seed-points from each iteration as the black points in the 3D view of Figure 7c. The seed-points are well located in seven primary locations; the few points not at the primary locations can be reduced by tuning the particle damping rate (**??**); however, it is computationally faster to simply cluster the results of these 20 iterations and take all clusters with more than a few points as the final seed-points, which are shown as the large red dots in the 3D view. This process of clustering the results of several iterations is very robust; the 2D view in Figure 7c shows the results of repeating this process 20 times, and you can see the computed seed-points are all in the same location.

We check the validity of our seven seed-points by running k-means clustering with two replicates (as implemented by Matlab's k-means++ routine) using 6, 7, and 8 clusters. We ran the clustering 20 times in each case, and show the results in Figure 7d. The stability of the results using 7 clusters (all 20 iterations produce the same seven positions), as compared with 6 or 8, imply that this 3D distribution does have seven clusters; and, the strong agreement in seed-point

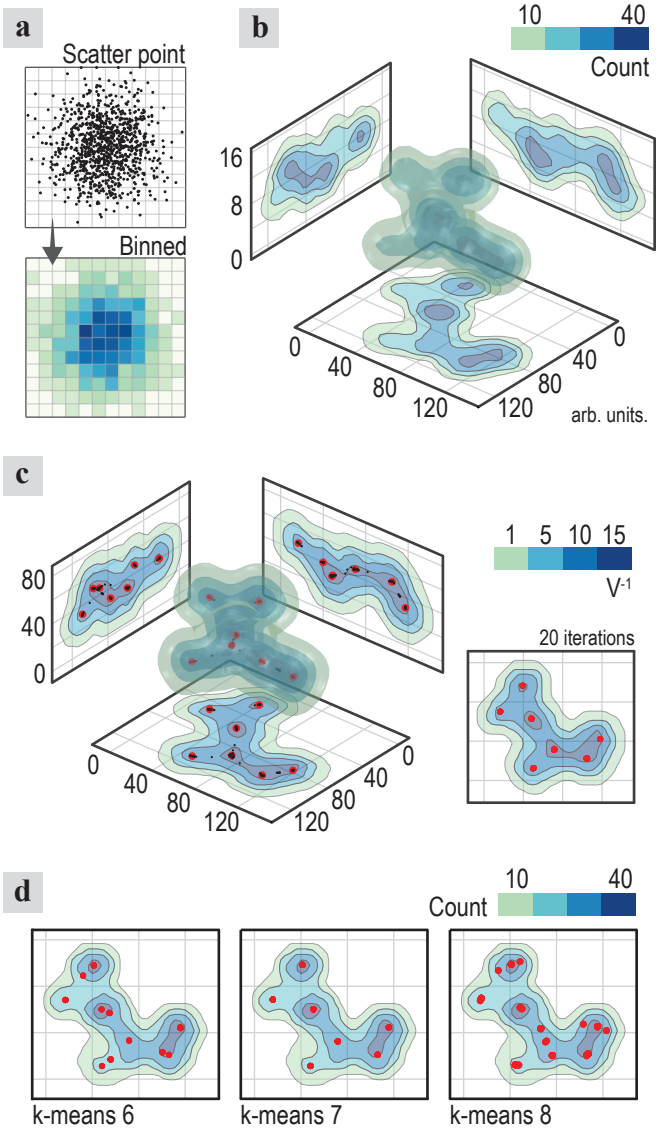


Fig. 7. SALR clustering for scatter point data. **a**, Example showing how scatter point data is binned. **b**, Isosurface plot of the point count in our 3D example; the axis planes show the projections of the isosurfaces to 2D. **c**, The confining potential created by thresholding the point count at 5 and then using (7). The black points are the results of running our seed-point calculation method 20 times, and the red points are the clusters formed from these results. The 2D view of the x-y plane shows the results of repeating this process 20 times. **d**, The red points show the result of running k-means clustering 20 times using 6, 7, and 8 clusters.

location between the k-means results and the results of our method imply that these are likely the correct locations.

These results indicate that our method can determine the correct number of seed-points and their correct location, and, in particular, could be a good choice for locating the centers of nuclei in 3D images, which this 3D distribution resembles.

4.2 More complex 5D data

We next turn to a real data set with five dimensions and 2.7 million points; each point represents one cell' nucleus (the same SCC25 cells as above) and the five features give a measure of the damage done to the nuclei. The data is scaled in each dimension by its standard deviation and then binned (about 33 bins in each dimension). Figure 8a shows point-count isosurfaces for two (of the ten possible) 3D projections of the 5D data; the data has one very dense region in the center (which are undamaged nuclei) and maybe three long/thin low-density regions extending outwards from it. The long/thin regions mean that the distance transform cannot be used to create the confining potential (**??**). Instead, we use one over the point density for the confining potential; however, we must scale the magnitude of the potential gradient $|\nabla V|$ so that it is approximately the same order of magnitude as the particle repulsion (see Discussion below). In addition to this change, we introduce two generalizations to our method.

- *Distance metric*: In higher dimensions, we found that we can get more stable results using the Minkowski distance metric where the distance between two points, $\mathbf{x}$ and $\mathbf{y}$, is defined as

$$d = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{11}$$

  where n is the dimension of the space and $p \geq 1$. If we set $p = 2$ then we have standard Euclidean distance; and if we set $p \to \infty$, then $d = \max_i |x_i - y_i|$, which, in our case, would mean that two particles are only attracted to each other if they are within r_a in all dimensions. (Note that, in general, any distance metric can be used as long as the interaction potential is correctly defined.)
- *Isotropic scaling*: We make a distinction between the data space (the N-D space the data is defined in) and the solver space (the N-D space we solve the equations of motion). We map between the two spaces with a simple scale factor defined to be ℓ/r_a, where r_a is the attractive extent in data space and ℓ is the *characteristic distance* of the solver space, which, for this paper, can be directly interpreted as the attractive extent in the solver space.
  Introducing the solver space has two primary benefits: 1) If the attractive extent in data space should be $r_a \lesssim 2$, then it is not possible to accurately solve for the interaction potential parameters A, μ, and σ (assuming $r_0 \approx 0.2 r_a$ and $d_0 = -1$). The solver space lets us scale up the problem to use an attractive extent of ℓ at which we can accurately solve for the parameters. 2) Changing the value of ℓ can precisely control the magnitude of the particle interaction (this could also be achieved using the coupling constant, k, if one wanted).
  In **??**, we show how the interaction potential and force in data space change as ℓ goes from 10 to 50. The interaction force is strongest at $\ell = 10$ and decreases as ℓ increases.

We model particles five times (small black points in Figure 8a) and cluster the results (red points in Figure 8a). You can see that our method located the three low density extrusions (labeled by black arrows) very well, and our method even finds the very small, but distinct, cluster marked by the circle in right plot of Figure 8a. (Supplemental Video 2 shows an animation of the particle dynamics along with the comparison to k-means with 5 clusters.) The parameter values used in these simulations were as follows; uniform random distribution, $r_s \approx 1.5$, $[V_{\min}, V_{\max}] = [1/4, 1/6]$,

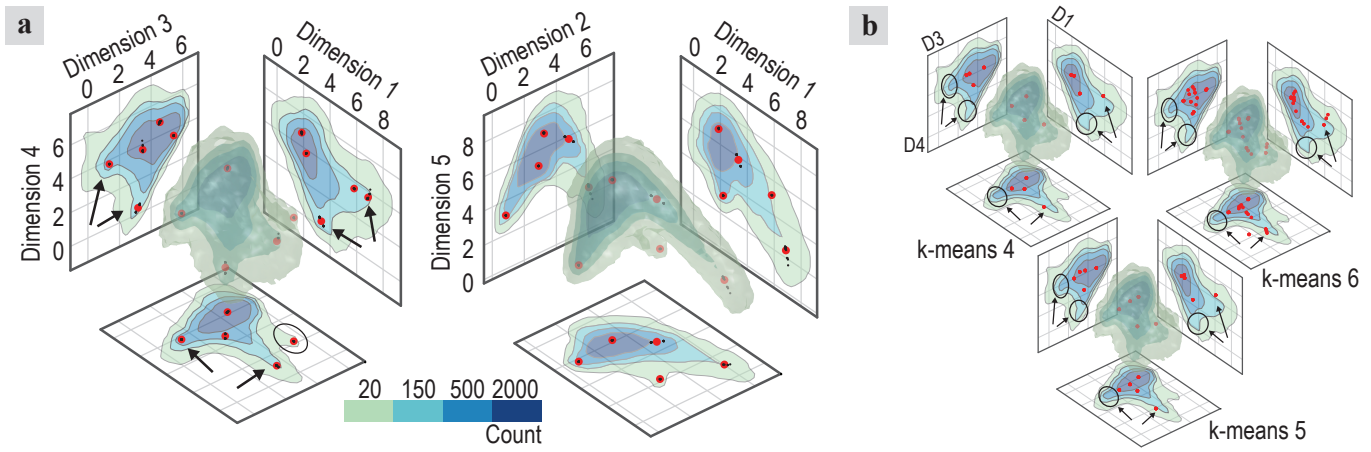


Fig. 8. SALR clustering for real data. **a**, projections of the 5D point count down to dimensions 1, 3, and 4 and dimensions 1, 2, and 5. Black points are the results from each of the 5 iterations, red points are the final seed-points. Arrows label the three low density regions. **b**, results using k-means with 4, 5, and 6 clusters. The black arrow labels the one seed-point that approximately locates a low-density region; the circles show the two missing seed-points.

$r_a = 2$, $r_0 = 0.15\,r_a$, $d_0 = -1$, $\ell = 12$, $p = 4$, and the potential force was scaled so that the 99% value is 0.4. (The potential bound requirements helps to reduce the number of particles, which is 147, and to position the particles near the outskirts of the region, which is better, in general.)

As means of comparison, we used k-means clustering with two repetitions for 4, 5, and 6 clusters; we repeated this 20 times and show the results in Figure 8b. The results using 4 or 5 clusters are stable (the clusters are in the same place in all 20 iterations), but are only able to approximately locate one of the three low density regions, while missing two of them (labeled by the circles). The reason k-means does not locate the low density regions well is that the cluster centers it finds are the locations which minimize the variance in each cluster, and the variance can be minimized by moving the cluster centers closer to the high density regions. (The same/similar reasoning also applies for fuzzy c-means or mixture of Gaussians.) This demonstrates an important feature of our method: our method can locate the center of low density regions (also called rare classes) even when the low density regions do not have local maximum.

Looking at the computation time, the k-means with 5 clusters, which was the fastest, takes 15 ± 5 sec per repetition; our method takes 1.15 ± 0.08 sec per iteration with ~ 5.5 sec overhead for binning, smoothing, and computing the confining potential gradient. (Note each repetition of k-means and each iteration of our method can run in parallel. Computer specifications are shown below.) Thus, our method can lead to an important speed increase for some data sets.

5 Discussion

It is interesting to consider how, and under what conditions, SALR clustering leads to these improvements. The answer comes by analyzing the force on the particles in the simulation: the confining potential exerts a force on the particles with a magnitude of $|\nabla V|$ (which is the slope of the confining potential in Figure 1f), and the particles repel each other with a force of about r_a^{-2}. If the force from the confining potential is much larger than the repulsive force, $|\nabla V| \ggg r_a^{-2}$, then the particles will only find the local minimum of the confining potential—this would lead to the results labeled Dist. Trans. in Figure 5, and is the same thing that the previous methods do in finding the local extrema of the voting landscape or fining the local maxima of the point density (mean-shift). Alternatively, if the repulsive force is much larger than the confining force, $r_a^{-2} \ggg |\nabla V|$, then the particles will ignore the confining potential minima and spread out—this would result in particle clusters separated by at least a distance r_a that uniformly cover the region. In between these two regimes, when the two forces are approximately the same order of magnitude, $|\nabla V| \sim r_a^{-2}$, there is an interaction between the confining potential and the particle repulsion, and this is the regime we operate in and that leads to SALR clustering's improved performance. (When using the distance transform to create the confining potential in our images, our confining force is $\propto r^{-2}$, where r is the distance to the nearest region boundary, and the repulsive force between the particles is also $\propto r^{-2}$, where r is the distance between particles.) This need for the forces to be approximately equal is why it is necessary to scale the magnitude of the potential gradient when the inverse point density is used as the confining potential.

Our method can be applied to even higher dimensional data sets, but it likely will not be possible to bin the data as we did above. Binning the data requires a large amount of space (the 5D data set, 37x30x28x34x36, takes ~ 150 MB as single precision), and is not necessary when the density is used as the confining potential. The benefit of binning the data is computational speed, as it allows us to precompute the density and its gradient everywhere. If the data cannot be binned, then the gradient of the density will need to be calculated (using a nearest neighbors range search) while solving the differential equations; this could be quite slow. Additionally, the magnitude of the gradient should be scaled; thus, before modeling the particle dynamics, a random sampling of the gradient magnitude will need to be performed so that the scale factor can be determined.

There are still many aspects of SALR clustering that are

missing or can be improved and expanded upon, such as using an asymmetric particle interaction, developing a performance metric, implementing a clustering method (where each scatter point is placed into a specific cluster), and using multiple confining potentials and types of particles. These are each briefly discussed in **??**.

6 Conclusion

SALR clustering can represent a significant improvement in locating the centers of overlapping convex objects: it locates the correct number of nuclei more often and the nuclei centers more accurately than standard and leading methods; it can significantly improve the performance of previous methods; and it is able to determine, not only the number of clusters, but the correct position of the cluster centers in data clustering while not requiring a cluster to have a local density maximum.

Acknowledgments

The authors would like to acknowledge Guillermina Garcia from Sanford-Burnham Medical Research Institute for providing the whole slide fluorescence imaging. J.K. acknowledges Prof. Sylwia Ptasinska for the opportunity and resources to work on this project. J.K. and X.H. acknowledge the support from the US Department of Energy Office of Science, Office of Basic Energy Sciences, under Award Number DE-FC02-04ER15533. This is contribution number NDRL 5175 from the Notre Dame Radiation Laboratory.

Computer specifications

Computation times related to locating the nuclei centers are based on using a 64-bit i7-4790 CPU 3.60 GHz with 32.0 GB ram. Computations times reported for the 5D data set are based on using a 64-bit i7-3770 CPU 3.40 GHz with 12.0 GB ram.

Code & data availability

The SALR clustering code (written in Matlab) and documentation, as well as all images, truth data, and scatter point data used in this work, are in Supplementary Software and Data and available on GitHub (https://github.com/jkpld/SALR_Clustering). In particular, we include example scripts that directly reproduce the results of this work.

Author contributions

J.K. conceived the project, implemented the method, created validation truth data, analyzed the results, and wrote the manuscript. X.H. performed all tasks relating to the cell work (cell culture, cell experiments, staining the cells, having them imaged) and provided helpful discussion throughout the course of the project. D.M. provided valuable support and feedback in elevating the project and improving the manuscript.

References

[1] F. Xing and L. Yang, "Robust Nucleus/Cell Detection and Segmentation in Digital Pathology and Microscopy Images: A Comprehensive Review," *IEEE Reviews in Biomedical Engineering*, vol. 3333, no. c, pp. 1–1, 2016.

[2] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, "Methods for Nuclei Detection, Segmentation and Classification in Digital Histopathology: A Review. Current Status and Future Potential," *IEEE Reviews in Biomedical Engineering*, vol. PP, no. 99, pp. 1–1, 2013.

[3] T. J. Fuchs and J. M. Buhmann, "Computational pathology: Challenges and promises for tissue analysis," *Computerized Medical Imaging and Graphics*, vol. 35, no. 7-8, pp. 515–530, 2011.

[4] S. Kothari, J. H. Phan, T. H. Stokes, and M. D. Wang, "Pathology imaging informatics for quantitative analysis of whole-slide images." *Journal of the American Medical Informatics Association : JAMIA*, vol. 20, no. 6, pp. 1099–108, 2013.

[5] C. Sommer and D. W. Gerlich, "Machine learning in cell biology teaching computers to recognize phenotypes," *Journal of Cell Science*, vol. 126, no. Pt 24, pp. 5529–5539, 2013.

[6] P. J. Navarro, F. Pérez, J. Weiss, and M. Egea-Cortines, "Machine learning and computer vision system for phenotype data acquisition and analysis in plants," *Sensors (Switzerland)*, vol. 16, no. 5, 2016.

[7] L. D. Cohen, "Contour Models," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 2, pp. 211–218, 1991.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[9] C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén, and J. C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *IEEE Transactions on Medical Imaging*, vol. 21, no. 10, pp. 1212–1221, 2002.

[10] X. Qi, F. Xing, D. J. Foran, and L. Yang, "Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 3, pp. 754–765, 2012.

[11] B. T. Grys, D. S. Lo, N. Sahin, O. Z. Kraus, Q. Morris, C. Boone, and B. J. Andrews, "Machine learning and computer vision approaches for phenotypic profiling," *Journal of Cell Biology*, vol. 216, no. 1, pp. 65–71, 2017.

[12] T. Blasi, H. Hennig, H. D. Summers, F. J. Theis, J. Cerveira, J. O. Patterson, D. Davies, A. Filby, A. E. Carpenter, and P. Rees, "Label-free cell cycle analysis for high-throughput imaging flow cytometry," *Nature Communications*, vol. 7, p. 10256, 2016.

[13] D. Chen, S. Sarkar, J. Candia, S. J. Florczyk, S. Bodhak, M. K. Driscoll, C. G. Simon, J. P. Dunkers, and W. Losert, "Machine learning based methodology to identify cell shape phenotypes associated with microenvironmental cues," *Biomaterials*, vol. 104, pp. 104–118, 2016.

[14] Q. Zhong, A. G. Busetto, J. P. Fededa, J. M. Buhmann, and D. W. Gerlich, "Unsupervised modeling of cell morphology dynamics for time-lapse microscopy," *Nature Methods*, vol. 9, no. 7, pp. 711–713, 2012.

[15] J. Wang, X. Zhou, P. L. Bradley, S.-F. Chang, N. Perrimon, and S. T. Wong, "Cellular Phenotype Recognition for High-Content RNA Interference Genome-Wide Screening," *Journal of Biomolecular Screening*, vol. 13, no. 1, pp. 29–39, 2007.

[16] B. Parvin, Q. Yang, J. Han, H. Chang, B. Rydberg, and M. H. Barcellos-Hoff, "Iterative voting for inference of structural saliency and characterization of subcellular events," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 615–623, 2007.

[17] X. Zhang, H. Su, L. Yang, and S. Zhang, "Fine-grained histopathological image analysis via robust segmentation and large-scale retrieval," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 5361–5368, 2015.

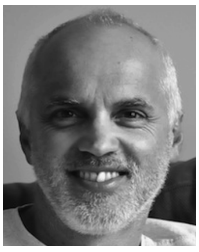
[18] H. Xu, C. Lu, and M. Mandal, "An Efficient Technique for Nuclei Segmentation Based on Ellipse Descriptor Analysis and Improved Seed Detection Algorithm," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1729–1741, 2014.

[19] P. Quelhas, M. Marcuzzo, A. M. Mendonça, and A. Campilho, "Cell nuclei and cytoplasm joint segmentation using the sliding band filter," *IEEE Transactions on Medical Imaging*, vol. 29, no. 8, pp. 1463–1473, 2010.

[20] T. Esteves, P. Quelhas, A. M. Mendonça, and A. Campilho, "Gradient convergence filters and a phase congruency approach for in vivo cell nuclei detection," *Machine Vision and Applications*, vol. 23, no. 4, pp. 623–638, 2012.
[21] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79 – 116, 1998.
[22] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.
[23] H. Kong, H. C. Akakin, and S. E. Sarma, "A generalized laplacian of gaussian filter for blob detection and its applications," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1719–1733, 2013.
[24] H. Xu, C. Lu, R. Berendt, N. Jha, M. Mandal, and S. Member, "Automatic Nuclei Detection based on Generalized Laplacian of Gaussian Filters," *Journal of Biomedical and Health Informatics*, vol. XX, no. XX, pp. 1–12, 2016.
[25] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 233, pp. 281–297, 1967.
[26] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.
[27] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, 1981.
[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, 1977, vol. 39, no. 1.
[29] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
[30] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
[31] F. Bolton and U. Rössler, "Classical model of a Wigner crystal in a quantum dot," *Superlattices and Microstructures*, vol. 13, no. 2, p. 139, 1993.
[32] S. Mossa, F. Sciortino, P. Tartaglia, and E. Zaccarelli, "Ground-state clusters for short-range attractive and long-range repulsive potentials," *Langmuir*, vol. 20, no. 24, pp. 10 756–10 763, 2004.
[33] F. Sciortino, S. Mossa, E. Zaccarelli, and P. Tartaglia, "Equilibrium cluster phases and low-density arrested disordered states: The role of short-range attraction and long-range repulsion," *Physical Review Letters*, vol. 93, no. 5, pp. 5–8, 2004.
[34] P. Bogacki and L. Shampine, "A 3(2) Pair of Runge-Kutta Formulas," *Appl Math Lett*, vol. 2, no. 4, pp. 321–325, 1989.
[35] S. A. Blundell and S. Chacko, "Excited states of incipient Wigner molecules," *Physical Review B*, vol. 83, no. 19, p. 195444, 2011.
[36] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[37] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal," *In British Machine Vision Conference*, pp. 384–393, 2002.

James Kapaldo James Kapaldo is a Ph.D candidate in Physics at the University of Notre Dame. His current research interests include machine learning and its use for analyzing scientific experiments, experimental design and automation, the study of cold plasma jets and their effect on oral cancer cells. Kapaldo received a M.Sc. degree in physics from the University of Notre Dame studying Wigner Molecules and Ga-In intermixing in InP/GaInP quantum dots. During the summers '09 (on a DAAD fellowship) and '10 he was a research intern at Max Planck for Quantum Optics near Garching, Germany where he worked on designing and building a ultra-high vacuum beamline for attosecond laser experiments. Contact him at jkapaldo@nd.edu.

Xu Han Xu Han received a Ph.D ('17) in Physics at the University of Notre Dame. Her research interests include studying the effect of cold plasma jets on oral cancer cells, using dosimetry methods for studying the chemical changes induced by cold plasma jets, and cold plasma jet diagnostics. Contact her at xhan1@nd.edu.



Domingo Mery Domingo Mery (M01) received the M.Sc. degree in electrical engineering from the Technical University of Karlsruhe in 1992, and the Ph.D. degree (Hons.) from the Technical University of Berlin in 2000. He was a Research Scientist with the Institute for Measurement and Automation Technology, Technical University of Berlin, in collaboration with YXLON X-Ray International. In 2001, he served as an Associate Researcher with the Department of Computer Engineering, Universidad de Santiago, Chile. In 2014, he was a Visiting Professor with the University of Notre Dame. He is currently a Full Professor with the Department of Computer Science, Pontificia Universidad Catlica de Chile, where he served as the Chair from 2005 to 2009. He has authored or coauthored 60 technical SCI publications and over 70 conference papers. His research interests include image processing for fault detection in aluminum castings, X-ray imaging, real-time programming, and computer vision. He has received scholarships from the Konrad Adenauer Foundation and the German Academic Exchange Service. He received the Ron Halmshaw Award in 2005 and 2012, the John Green Award from the British Institute of Nondestructive Testing in 2013, which was established to recognize the best papers published in the Insight Journal on Industrial Radiography, and the Best Paper Award at the International Workshop on Biometrics in conjunction with the European Conference on Computer Vision in 2014. He is a Local Co-Chair of ICCV2015 (to be held in Santiago de Chile). He served as the General Program Chair of PSIVT2007, the Program Chair of PSIVT2009, and the General Co-Chair of PSIVT2011 (Pacific-Rim Symposium on Image and Video Technology) and the 2007 Ibero-American Congress on Pattern Recognition.