Informed Component Label Algorithm for Identifying Connected Regions of Eulerian Data

Kelli Hendrickson[a,*], Gabriel D. Weymouth[b], Dick K.-P. Yue[a]

[a]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b]*Southampton Marine and Maritime Institute, University of Southampton, SO16 7QF, UK*

Abstract

We present two modifications to the standard connected component labeling techniques, or painter's algorithm, that remove the algorithm's bias when identifying components with spacing on the order of a grid size. Unlike the painter's algorithm, this Informed Component Labeling (ICL) algorithm incorporates additional knowledge of the field variable by using multilevel thresholding and surface normal information to establish and refine connectivity decisions. Using Volume-Of-Fluid (VOF) data as an example, we show that the ICL algorithm accurately predicts the connectivity of components with spacing on the order of the grid. Although our immediate objective is to identify entrained bubbles and spray droplets in VOF schemes, ICL applies to any Eulerian field variables where accurate Lagrangian information is desired.

Keywords:

1. Introduction

A critical research area in the study of air entrainment due to breaking waves is measuring and understanding the bubble-size distribution and bubble statistics during and after the breaking event. Obtaining this type of Lagrangian information accurately from simulations that rely on Eulerian interface capturing methods (such as Volume-of-Fluid and level set) is challenging. Recent

[*] Corresponding author
Email address: khendrk@mit.edu (Kelli Hendrickson)

application of the so-called ‘painter’s algorithm’ to identify connected regions of air has enabled Eulerian-based methods to identify and quantify individual entrained air bubbles [1, 2, 3] . The algorithm, which shares a pedigree with the connected component labeling (CCL) in pattern recognition and computer vision techniques [4, 5], translates the interface variable into a binary field (0 in air and 1 in water) and then sweeps through the domain point-by-point, determining paths of connectivity. The technique identifies components not connected to the bulk air field as bubbles and records information about their location, volume, etc.

This binary-based painter’s algorithm suffers a distinct weakness when bubbles are within a grid point of each other in that it can misidentify them as a single larger bubble. A extreme but not infrequent example would be a chain of individual bubbles that the algorithm identifies as long ligaments. While the measurement of entrained air is technically accurate, the painter’s algorithm biases distribution data towards larger bubbles.

This paper describes modifications to the original CCL algorithm [4] to correctly identify bubbles with spacing on the order of the grid size, removing this bias. This Informed Component Labeling algorithm (ICL) incorporates knowledge of the field variable and modifies the CCL algorithm using two main strategies. The first employs multilevel thresholding of the original field variable within the sweeps of the CCL algorithm. Retaining the field variable information (instead of binarizing it as in the painter’s algorithm) allows us to establish and refine the connectivity of the components with greater accuracy. The second strategy incorporates the gradient of the field variable to further establish connectivity. The ICL algorithm applies for any Eulerian field variable where Lagrangian information is desired. For illustration in this paper, we assume the field variable is the Volume-of-Fluid (VOF) function f that represents the normalized volume of water in a given cell. The surface normal information originates from the gradient of f or its linear reconstruction.

This paper is organized as follows. §2 defines the original CCL/painter’s algorithm and details the modifications that then form the ICL algorithm. §3 provides two detailed examples of how ICL resolves connectivity of components with spacing on the order of the grid. §4 provides verification tests and §5 concludes the paper.

2. Methodology

For simplicity of discussion, our field variable is the VOF function $f(i, j, k)$ that represents the volume of water in a given cell: 1, 0.5, or 0 if respectively full, half full, or empty (for example in [6]). The field resides within a three-dimensional "IJK" domain sized $N_i \times N_j \times N_k$. The surface normal vector for the fluid within the cell $\vec{n}(i, j, k)$ results from the linear reconstruction within VOF. We desire to know the set *blob* that contains a listing of every connected component within the domain. For each connected component, we require its volume V, centroid location (i_C, j_C, k_C), and minimum and maximum extent (x_i, x_j, x_k) in each direction. We note that including additional mean velocity or other quantities is straightforward. Without loss of generality, we assume that the grid is Cartesian and has the same number of points in each direction.

For the CCL (and ICL) algorithms, let $v(i, j, k) \geq 0$ be the field information to assess (within the same IJK domain). Any point i, j, k is connected to another point i', j', k' providing that a path exists between the two points that satisfies a given mask m. All of the points along a given path are thus part of the same component and share the same label l. The size of the set *blob* is the number of unique labels L. For our version of the CCL, an additional array $\ell(i, j, k)$ stores the label for each point in the domain. For simplicity of notation, we present the following algorithms in two dimensions on a Cartesian grid. We also note that ICL can apply to data with domain decomposition [7].

2.1. Standard CCL Algorithm

The basis for our algorithm is a pixel-based label-equivalent-resolving two-scan method, or the standard connected-component label (CCL) algorithm. The first scan assigns and resolves an equivalent label for each component and the second scan replaces the equivalent label and records the *blob* information. Section 3.3 of [5] provides an in depth overview of the state-of-the-art algorithms that is summarized here in our simplified form. The method is similar to other methods that arise from the need to (re-)assemble narrow bands around level sets [8, 9].

Scan 1: Equivalent label assignment:. Initialize the unique label l and specify a threshold t. For each point i, j, identify the current field value $v(i, j)$. The standard CCL algorithm proceeds as follows using a backward-looking mask m=$\{(i-1, j), (i, j-1)\}$:

Algorithm 1. *Standard CCL(v(i,j),t)*

```
1. if $v(i,j) > t$
   (a) for each point $p$ in mask $m$ that has a label $\ell(p) \neq 0$ assign
       $\ell(i,j) \leftarrow \begin{cases} Union(\ell(p), \ell(i,j)) & \ell(i,j) \neq 0 \\ \ell(p) & \text{otherwise} \end{cases}$
   (b) if no label is yet to be assigned, $l \leftarrow l + 1$, $\ell(i,j) \leftarrow NewTree(l)$
```

Step 1.1a uses a union-find-tree strategy to assign a provisional label using the neighboring cells. Step 1.1b assigns a new label if there is no result from step 1.1a. At the end of this first scan, all equivalent labels of a connected component are nodes in the corresponding tree (resulting from the union-find-tree strategy). The CCL performs a sweep of the labels (or flattening of the tree) to ensure each node in a tree directly points at its root. For brevity, we do not include here the details for the union-find-tree strategy or the flattening step as they are standard and refer the readers to [5, 10] for details.

Scan 2: Assign the representative label and collect data. For each point i, j, if $\ell(i, j) \neq 0$, assign the representative label (or the root of the tree) and collect the information about the components. The act of adding the information to the set $blob(\ell)$ through the AddInfo routine is as follows:

Algorithm 2. *AddInfo(S,v,i,j)*

```
1. $S[V] \leftarrow S[V] + v$
2. $S[ic, jc] \leftarrow S[ic, jc] + v * [i, j]$
3. $S[imax, jmax] \leftarrow \max(S[imax, jmax], [i, j])$
4. $S[imin, jmin] \leftarrow \min(S[imin, jmin], [i, j])$
```

Note that for each component the extent $(x_i, x_j) = (S[imax]-S[imin], S[jmax]-S[jmin])$ from algorithm steps 2.3 and 2.4.

The painter's algorithm is equivalent to this CCL algorithm with a threshold value t=0. As we will show in §3, the reduction of information caused by the binarization of the field data (especially the VOF function f) is the primary reason for the bias in the painter's algorithm to connect components that are spaced near each other.

2.2. Informed Component Labeling Algorithm (ICL)

Unlike the image data that formed the basis of the CCL algorithm, the field variable being processed has information that can aide the algorithm in determining connectivity. The ICL algorithm retains this valuable data and modifies the standard CCL algorithm to incorporate it using two main concepts: multilevel thresholding and utilizing additional (gradient) information. These straightforward modifications create a robust and accurate algorithm for identifying connected regions.

2.2.1. Multilevel Thresholding

The ICL algorithm incorporates multilevel thresholding by sweeping over the CCL algorithm for a number of Θ thresholds $thr = (t_\Theta, t_{\Theta-1}, \ldots, t_1)$ in the first scan. The full algorithm is:

Algorithm 3. *ICL (v,thr)*

```
1. for each threshold $t \leftarrow thr[\theta]$, $\theta \leftarrow \Theta$ to 1
```

```
  (a) *for all* points $(i,j)$ in the domain, call *OCL(v(i,j),t)*
2. *for all* points in the domain where $\ell(i,j) \neq 0$
  (a) $\ell(i,j) \leftarrow Root(\ell(i,j))$
          ... relabel with sequential labeling, if desired.
  (b) collect the data in set *blob* by calling $AddInfo(blob[\ell(i,j)], v(i,j), i, j)$
```

Unlike the application of image processing, the inherent knowledge of what the field variable $v(i,j)$ represents guides the number of levels and our selection criteria. Returning to our VOF example, we define the first threshold level with half-full cells f=0.5. The first pass through the domain with t_Θ=0.5 determines the initial connectivity of the components using only cells that are at least half full. Successive passes through the domain with reduced threshold levels ($\Theta \leq 3$ is sufficient for VOF) either adds to the connectivity for existing components or identifies smaller components. To label all of the components (and conserve volume), the last threshold sweep must use t_1=0. We found that Θ=3 produces an algorithm insensitive to the value of the threshold levels providing t_Θ=0.5 and t_1=0. The effectiveness of this modification is detailed in section 3.

2.2.2. Incorporating Additional Field Information

We assume there exists an accurate normal vector $\vec{n}(i,j)$ at every point in the domain that represents the normal (or gradient) associated to the field function. In our VOF example, $\vec{n}$ represents the surface normal to the interface and is obtained from the VOF linear reconstruction. Incorporating the normal vector into the standard CCL algorithm (algorithm step 1.1.a), allows further discrimination of connectivity. The CCLN algorithm now reads as:

Algorithm 4. *CCLN(v(i,j),$\vec{n}$(i,j),t)*

```
1. if $v(i,j) > t$
  (a) *for each* point $p$ in mask $m$ that has a label $\ell(p) \neq 0$ assign
      $\ell(i,j) \leftarrow \begin{cases} Union(\ell(p), \ell(i,j)) & \ell(i,j) \neq 0 \textbf{ and } n_p(p) \cdot n_p(i,j) \geq 0 \\ \ell(p) & \ell(i,j) = 0 \textbf{ and } n_p(p) \cdot n_p(i,j) \geq 0 \end{cases}$
```

(b) if no label is yet to be assigned, $l \leftarrow l + 1$, $\ell(i,j) \leftarrow NewTree(l)$

Algorithm step 4.1.a essentially allows connectivity *only* if the normals at the point of interest align with normals of the mask point (in the direction of the mask). We note here that the inclusion of the normal information can remove the bias of the painter's algorithm (see section 4.1).

2.3. ICL and various field functions

The ICL algorithm can identify Lagrangian information from any Eulerian field data. A simple example is the level set method, used in simulating multiphase flows, crack growth in extended finite element methods [11] and shape detection in computer graphics. In typical level set methods, the function ϕ is a signed distance function with ϕ =0 the interface of interest and $\vec{n} \sim \nabla\phi$. Sample test 1 (see §4.1) shows that the CCL algorithm (a single threshold of t =0) easily establishes the connectivity and the normal information (CCLN) improves the estimate. However, the ICL/ICLN algorithm can identify and tracking the band region (as well as its edges) within the domain, which is necessary for narrow-banded level set methods. Within the context of vortex tracking in fluid dynamics with Eulerian methods, the ICL/ICLN method can provide Lagrangian information through connected components of the scalar λ_2, Q_c and Δ criteria, where a range of isovalues represent an estimate of the location and orientation of a vortex [12].

3. Effectiveness of ICL

We present two detailed examples of how the ICL performs compared to the painter's algorithm when identifying components that are near each other. For notation, we call the painter's algorithm CCL and consider two cases that originate in the context of multiphase flows: drops with spacing $h \sim \Delta$ (both large and small). For this section, we focus solely on the CCL and the multilevel threshold ICL algorithms. Further verification and validation of the algorithm (including CCLN and ICLN) is in §4.

First, consider two large drops spaced near each other . Figure 1a shows the raw field information $v = f$. The algorithm performance is equivalent to that of two bubbles if $v = 1 - f$. The CCL algorithm identifies the field as a single connected cavity (labeled 1 in figure 1b) using points **a-d**,**f-g**,**i-l**. In contrast to this, the ICL algorithm identifies two connected cavities (see figure 1d).

To understand the root cause of the differences, consider the first threshold level of the ICL algorithm $t_\Theta = 0.5$. It identifies two initial connected regions (labeled as 1 with points **j-l** and 2 with points **b-d**, as seen in figure 1c). With these two components identified, the remaining thresholds will incorporate into one of these two components. For example, consider **i** and **f** for t_1=0. For **i**, no point in the mask m is labeled and it is also not currently labeled, thus a new tree is created with label 4 via algorithm step 1.1.b. When the sweep reaches **j**, it will merge **i** into the same tree through the Union operation (algorithm step 1.1.a).

For **f** at t_1=0, **b** lies within the mask m and has an existing label and **f** is not currently labeled so it will receive the label of **b** (algoirthm step 1.1.b). We note here that the connection of **f** (and also **g**) into either of the components depends upon the direction of the overall scans and the mask.

The next example uses multiple smaller drops near (and offset) from each other. Figure 2a shows the raw field information (again, $v = f$) for multiple small drops with spacing $h \sim \Delta$, offset vertically from each other a distance $\sim \Delta$. The CCL algorithm identifies two connected components in figure 2b: (1) using **e**,**i-j**,**m-n**,**r** and (2) using **a**,**f-g**,**k**. Because **g** is not within the mask of **m**, these are not connected per our definition of connectivity.

The ICL after t_Θ=0.5 identifies 3 cavities in figure 2c: (1) using **f**; (2) using **j**; and (3) using **m-n**. As **j** does not lie within the mask of **n** and $v(\mathbf{i}) < t_\Theta$, these points are not within the same path and not connected at t_Θ=0.5. The additional t_i levels connects the remaining points to the established components (see figure 2d).

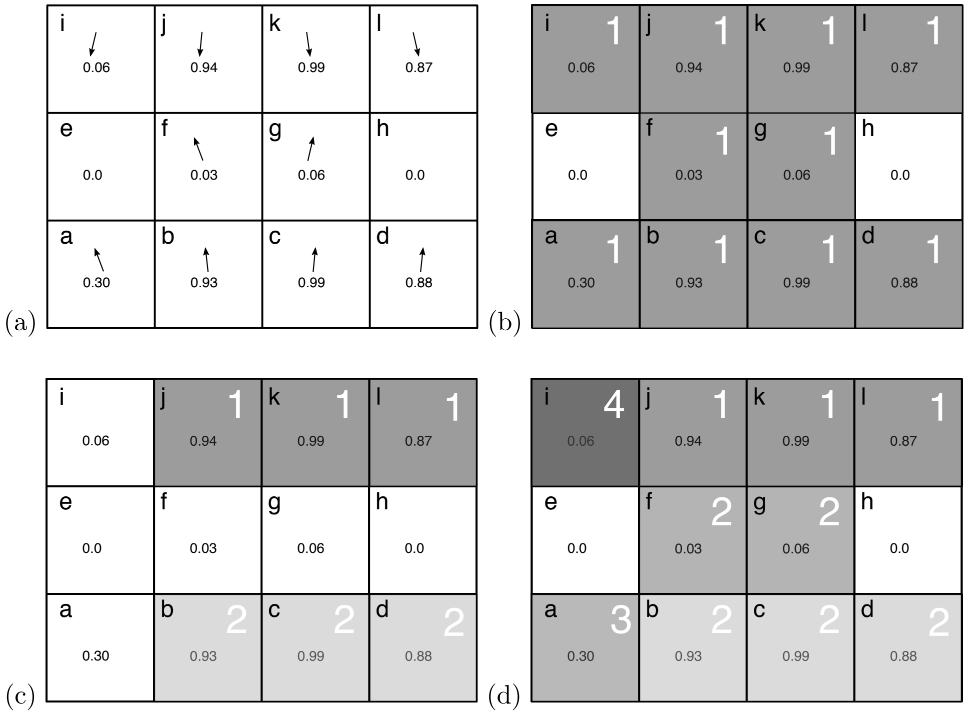


Figure 1: Example of CCL and ICL algorithms for a test with two drops with spacing $h \sim \Delta$. (a) raw field data, arrows represent $\vec{n}$; (b) CCL identified information; (c) ICL identified after t_{Θ}=0.5; (d) ICL identified after t_1=0.

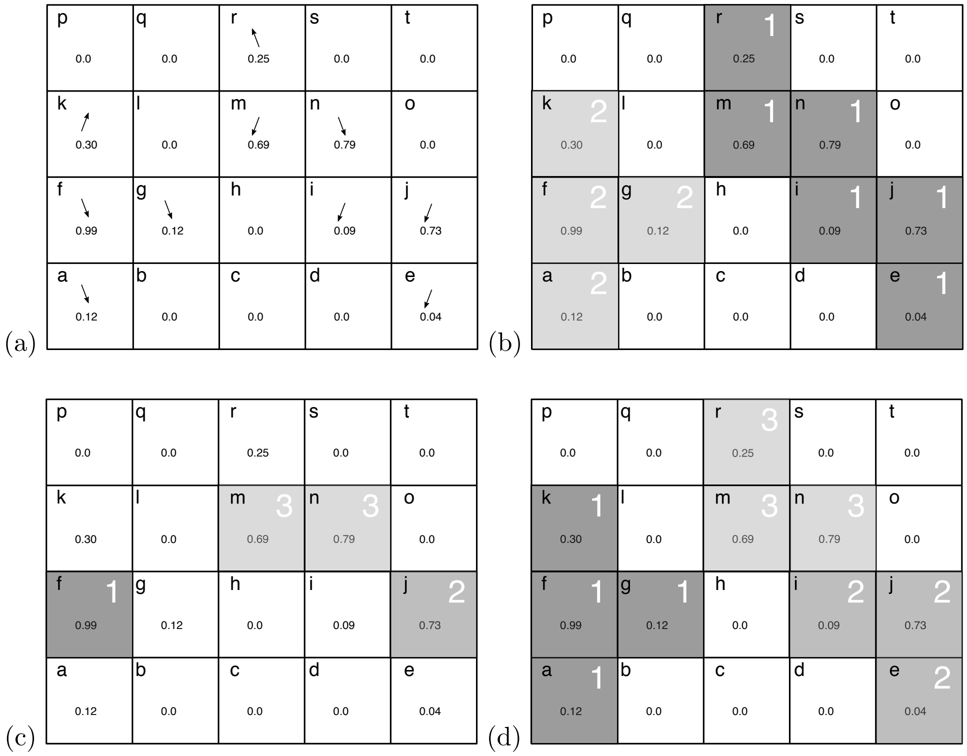


Figure 2: Example of CCL and ICL algorithms for a test with three small drops with spacing and offset $\sim \Delta$. (a) raw field data, arrows represent $\vec{n}$; (b) CCL identified information; (c) ICL identified after t_{Θ}=0.5; (d) ICL identified after t_1=0.

4. Verification & Validation

This section identifies three tests that establish the robustness and effectiveness of the modifications. We compare the modified algorithm (ICL) to the original painter's algorithm (CCL) as in §3 and include two additional variants: CCLN and ICLN. The 'N' designates the use of the normal information to inform decisions on connectivity on both the CCL and ICL algorithms.

For all of the tests below, we define a distance function ϕ based on the geometry of drops, calculate an initial volume fraction $\hat{f}$ from the distance function and pass it through the reconstruction scheme to create the field function f and its normal $\vec{n}$ to be consistent with simulated data. The field function to identify the drops is $v = f$. In all cases, the domain is a unit square and $\Delta = 1/32$ unless otherwise noted. In our sample tests, three-level $thr = (0.5, \alpha, 0)$ thresholds provide consistent results with ICL/ICLN, regardless of the choice of α. Thus, in these results ICL/ICLN use three threshold levels: $thr = (0.5, 0.25, 0)$.

4.1. Test 1: Two drops $r \gg \Delta$, spacing $h \sim \Delta$

Test 1 defines two drops placed such that their centers are vertically aligned and the minimum spacing between the two drops is Δ. Specifically: drop 1 r_1=0.13, $(x_0, y_0)_1$=(0.575, 0.33875) and drop 2 r_1=0.25, $(x_0, y_0)_2$=(0.575, 0.75). This test provides the raw field data in figure 1.

Table 4.1 shows location and radius of the drops identified by the four algorithms. As discussed in §3, CCL only identifies a single drop where ICL labels two drops. Of note, the CCLN algorithm also identifies two drops as the normals between points **j-k** and **f-g** do not align vertically (see figure 1a). Inspection of the resulting label field shows that CCLN and ICLN assign **f** to drop 1 and **g** to drop 2, which results in the small error in y_0 from the prescribed value for drop 2. Overall, the three modified algorithms (CCLN, ICL and ICLN) identify two drops with radius and location within an error $\ll \Delta$.

Also included in table 4.1 are two additional applications of the CCL/ICL algorithm. The first is CCL-H, which uses the single threshold algorithm with

Algorithm	Drop 1	Drop 2
CCL	0.2824; (0.5781, 0.5938)	–
CCLN	0.1313; (0.5781, 0.3438)	0.25; (0.5781; 0.7344)
ICL	0.1313; (0.5781, 0.3438)	0.25; (0.5781, 0.75)
ICLN	0.1313; (0.5781, 0.3438)	0.25; (0.5781, 0.7344)
CCL-H	0.1277; (0.5781, 0.3438)	0.2454; (0.5625; 0.75)
CCL-LS	0.1661; (0.5781, 0.3438)	0.3315; (0.5625, 0.75)
ICL-LS	0.1661; (0.5781, 0.3438)	0.3315; (0.5625, 0.75)

Table 1: Drop information r; (x, y) reported by algorithm for Test 1.

$t = 0.5$ to establish connectivity. The second is CCL-LS and ICL-LS, which use $v = |\phi|$ (the initial distance function) using $thr = 0$ and $thr = [0, \Delta/2, \Delta]$. All of these applications identify two drops and determines their locations and radii within $O(\Delta)$, as expected.

4.2. Test 2: Multiple subgrid drops, spacing $h \sim \Delta$

Test 2 defines a chain of subgrid drops $r = 0.6\Delta$ that have a horizontal spacing between their centers of 2.2Δ. The drop locations are $(x_i, y_i(n)), i = 1 \ldots N_b$, where n is the iteration step. The vertical drop location for bubble i is

$$y_i(n) = \begin{cases} 0.375 - n\Delta & i \text{ even} \\ 0.25 & i \text{ odd} \end{cases} .$$

The horizontal location of the first drop x_1=0.1+x', where $x' < 0.5\Delta$ to allow variability in the drop position within a cell. We prescribe the horizontal location of the remaining $i = 2 \ldots N_b$ drops from x_1. At step n=4 (see figure 4c), the drops are in line with each other with horizontal spacing between the interfaces Δ. For this test, we ran $n = 1 \ldots 10$ with 6 different realizations of x'.

Figure 3 shows contours of the volume fraction at select iterations for a single realization x'. For steps n =3 and 5, the drops are near each other but offset vertically (see figure 4b). Depending on the location of the drop with the cell,

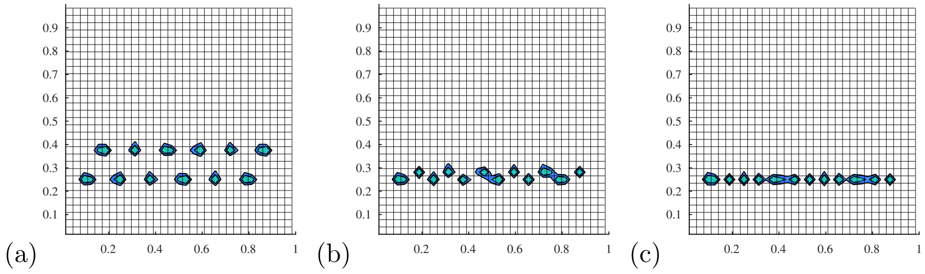


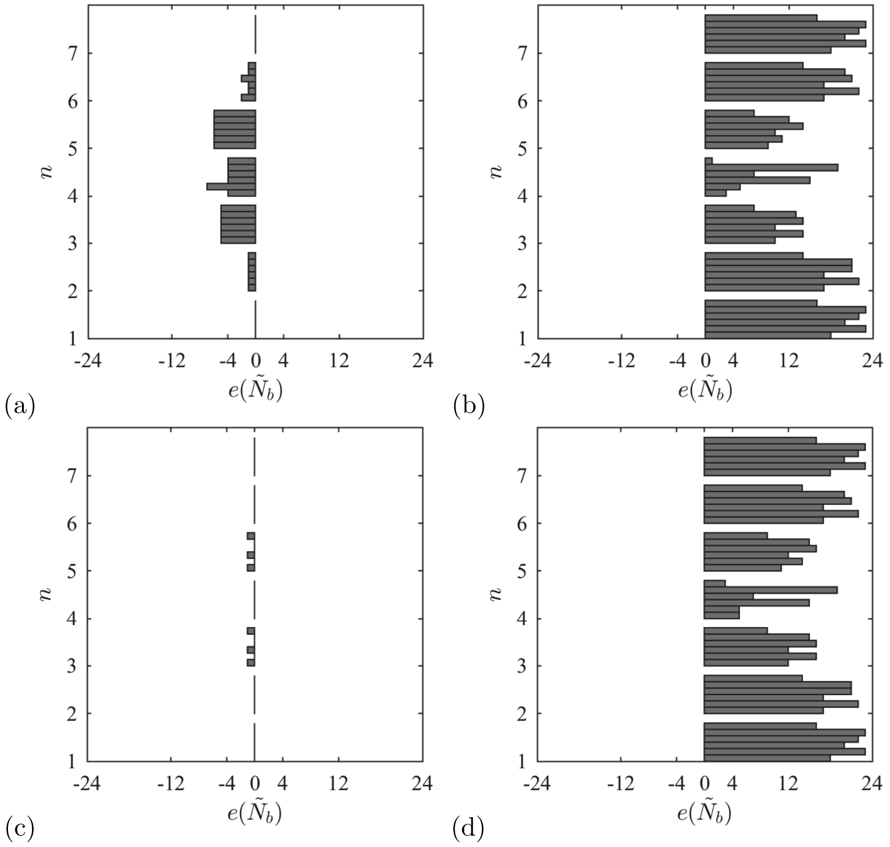
Figure 3: Contours of VOF function f at different steps (a) step 0; (b) step 3; and (c) step 4; for a single realization x'.

the resulting reconstructed f can indicate a single drop or a drop with radius larger than the cell (e.g. figure 3c). Thus, the solution N_b varies with each realization and is determined visually from f.

Figure 4 shows a bar chart of the error in the number of drops found $\tilde{N}_b$ at each step, $e(\tilde{N}_b) = \tilde{N}_b - N_b$. The different bars for each step represent the different realization of x'. As discussed in §3, CCL under predicts the number of drops, especially for steps 3-5 when the spacing between drops is the smallest (see figure 4a). ICL also incurs small error for steps 3 and 5, depending on the realization. However, the *N algorithms never under predict the number of drops, regardless if it is the single or multilevel threshold algorithm. In fact, it over predicts the number of subgrid-size drops because the algorithm depends on a normal that changes sign across a cell. This artifact fragments subgrid-size drops, shifting the error from $r > \Delta$ (due to merging of smaller drops into larger ones) to $r < \Delta$. As will be shown in Test 3, this fragmentation behavior only occurs for sub-grid components where the normals change sign across a cell (or drops with radius $\lesssim \Delta$). Tests with ligaments with thickness less than Δ do not get fragmented by CCLN/ICLN.

4.3. Test 3: Prescribed distribution, random placement

We designed the preceding tests to focus on the weakness of the painter's algorithm: identifying components placed near each other. Test 3 represents a "real world" problem with a known answer. To form the known solution, we



(a)

(b)

(c)

(d)

Figure 4: Error in drops $e(\tilde{N}_b) = \tilde{N}_b - N_b$ at each iteration n identified by the four algorithms for Test 2. (a) CCL; (b) CCLN; (c) ICL; (d) ICLN.

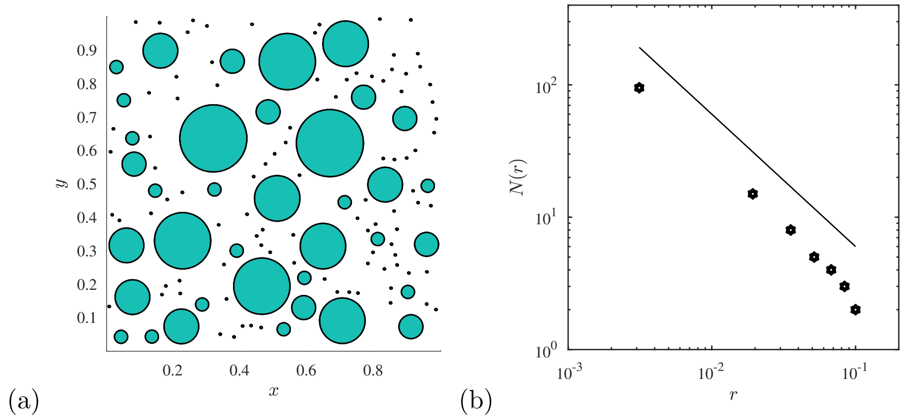


Figure 5: (a) $f = 0.5$ contour of the prescribed distribution of drops on the solution grid ($\Delta_s = 1/512$) and (b) resulting distribution of drops on the solution grid found by the four algorithms. ○ CCL; □ ICL; ▽ CCLN; △ ICLN; and —— r^{-1}.

prescribe a distribution of drops $N(r) = 0.3r^{-1}$, with 7 radii equally spaced between $r = [0.1, 0.003125]$ or specifically $r =$ (0.1, 8.385E-2, 6.771E-2, 5.156E-2, 3.542E-2, 1.927E-2, 3.125E-3) and place them randomly within the domain. Limits on the placement ensure: (1) every drop is fully inside the unit domain and (2) the distance between the interface of two drops must be greater than h_{min}. We use a highly resolved grid Δ_s=1/512 as the known solution and a test grid of Δ=1/64. We set the minimum gap h_{min}=1/64 such that the spacing of the drop on the test grid is resolvable. For this test, we constructed 10 different realizations of the spatial distribution for the solution, matching it on the test grid for each realization. As a reference, figure 5 shows an individual realization on the solution grid identified by all four algorithms.

Figure 6a shows the mean drop distribution (error bars are standard deviation $\sigma_N(r)$) identified for each of the four algorithms on the test grid. All of the algorithms predict the general trend of the number distribution. For clarity, figure 6b plots the standard deviation as a function of radius $\sigma_N(r)$. The original painters algorithm (CCL) has $\sigma_N(r)$ >0 above the test grid size, or in other words biases the distribution data for $r > \Delta$. The CCLN, ICL and ICLN algorithms, which represent all of the modifications developed here, have *zero*

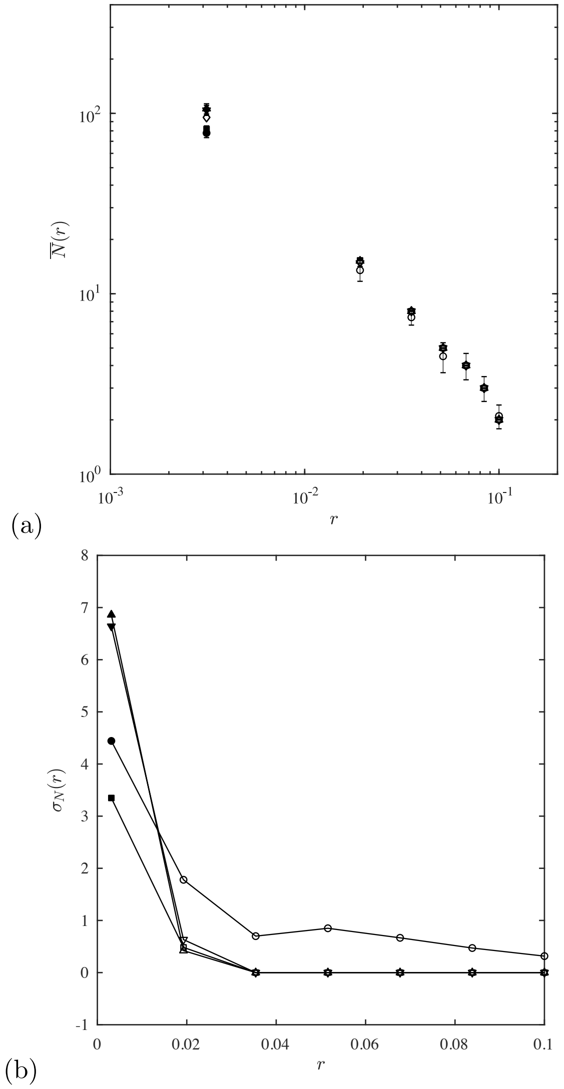


Figure 6: (a) Comparison of the mean distributions $\overline{N}(r)$ for Δ to the solution on Δ_s $\diamond$, error bars represent a standard deviation. (b) Standard deviation of the distribution $\sigma_N(r)$. $\circ$ CCL; $\square$ ICL; $\triangledown$ CCLN; and $\triangle$ ICLN. Filled symbols represent $r < \Delta$.

error for $r \gtrsim 1.927\text{E-}2 \approx 1.2\Delta$. The error is confined to distribution data for $r \leq \Delta$, which is the desired result.

5. Conclusions

We present a multilevel thresholding connected component labeling algorithm, Informed Component Labeling (ICL), that removes error and bias (towards larger components) encountered when using the standard connected component labeling (CCL) algorithms, or painter's algorithm. Rather than binarizing the field variable (as in the painter's algorithm), the ICL incorporates knowledge of the field variable and modifies the CCL algorithm using two main strategies.

The first uses multilevel thresholding to establish and refine the connectivity of the components. Unlike in computer vision and image processing applications where the number and levels of the thresholding are still an active area of research, we establish that the number of thresholds necessary is only three when the field information is the Volume-of-Fluid function f. We also note that, provided that the first level is 0.5 and the last level is that of the bulk fluid, ICL conserves volume and is insensitive to the value of the middle level.

The second modification incorporates the surface normal to further inform connectivity. Our tests show that this single modification applied to the original painter's algorithm improves accuracy in connecting nearby components, which is useful if the field function is the level set variable. This modification also inherently shifts any error in connectivity to sub-grid components due to the normal changing sign across the mask direction. This is critical in determining connectivity (or lack thereof) when many sub-grid components lie near each other.

The ICL/ICLN algorithm provides a robust and effective method of extracting Lagrangian information from an Eulerian field data. This applies to any application involving level sets (e.g. multiphase flows, crack growth and shape detection) and scalar fields (e.g. vortex identification and tracking and scalar variables). When combined with the Volume-of-Fluid function and the surface normal, the ICL/ICLN algorithm forms a powerful analytic technique for determining bubble- (and drop-) size distribution and statistics in multiphase flow simulations.

Acknowledgments

This work was funded by the Office of Naval Research N00014-10-1-0630 and N00014-17-1-2089 under the guidance of Dr. Thomas C. Fu, Dr. Ki-Han Kim and Dr. Joseph Gorski.

References

[1] L. Deike, W. K. Melville, S. Popinet, Air entrainment and bubble statistics in breaking waves, Journal of Fluid Mechanics 801 (2016) 91–129.

[2] Z. Wang, J. Yang, F. Stern, High-fidelity simulations of bubble, droplet and spray formation in breaking waves, Journal of Fluid Mechanics 792 (2016) 307–327.

[3] W. H. R. Chan, J. Urzay, P. Moin, Subgrid-scale modeling for microbubble generation amid colliding water surfaces, in: Proceedings of the 32nd Symposium on Naval Ship Hydrodynamics, 2018.

[4] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (4) (1988) 579–586.

[5] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, Y. Chao, The connected-component labeling problem: A review of state-of-the-art algorithms, Pattern Recognition 70 (2017) 25–43.

[6] G. Weymouth, D. K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on cartesian-grids, Journal of Computational Physics 229 (8) (2010) 2853–2865.

[7] C. Harrison, H. Childs, K. Gaither, Data-parallel mesh connected components labeling and analysis, in: T. Kuhlen, R. Pajarola, K. Zhou (Eds.), Eurographics Symposium on Parallel Graphics and Visualization, 2011.

[8] M. Herrmann, A parallel eulerian interface tracking/lagrangian point particle multi-scale coupling procedure, Journal of Computational Physics 229 (3) (2010) 745–759.

[9] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, Journal of Computational Physics 227 (4) (2008) 2674–2706.

[10] K. Wu, E. Otoo, K. Suzuki, Optimizing two-pass connected-component labeling algorithms, Pattern Analysis and Applications 12 (2) (2009) 117–135.

[11] M. Stolarska, D. L. Chopp, N. Moës, T. Belytschko, Modelling crack growth by level sets in the extended finite element method, International Journal for Numerical Methods in Engineering 51 (8) 943–960.

[12] J. Jeong, F. Hussain, On the identification of a vortex, Journal of Fluid Mechanics 285 (1995) 69–94.