Introduction to Level Sets for Image Analysis

James McClure
Overview

- Level set representation of surfaces
- Data representation for porous media
- Marching cubes (MC) algorithm
- Porous medium marching cubes algorithm (PMMC)
- Higher order PMMC (HOPMMC)
- Interfacial curvatures
- Examples: Lattice Boltzmann, X-ray tomography
What is a Level Set?

Suppose that you have some continuous function $G(x)$ defined $\forall x \in \Omega$

If we set this function to be equal to a constant value $\nu$, $G(x) = \nu$ defines an “level set” or isosurface.

If we are able to determine $G(x)$ and $\nu$ so that this isosurface corresponds to a surface which we are interested in, our job will be easy.
Data Sets
Data Sets
### Data Sets

```
<table>
<thead>
<tr>
<th>0.38</th>
<th>0.4</th>
<th>0.36</th>
<th>0.33</th>
<th>0.37</th>
<th>0.51</th>
<th>0.64</th>
<th>0.68</th>
<th>0.66</th>
<th>0.67</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.37</td>
<td>0.39</td>
<td>0.36</td>
<td>0.34</td>
<td>0.39</td>
<td>0.52</td>
<td>0.66</td>
<td>0.68</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>0.38</td>
<td>0.39</td>
<td>0.37</td>
<td>0.33</td>
<td>0.4</td>
<td>0.57</td>
<td>0.66</td>
<td>0.66</td>
<td>0.66</td>
<td>0.68</td>
</tr>
<tr>
<td>0.39</td>
<td>0.39</td>
<td>0.35</td>
<td>0.34</td>
<td>0.42</td>
<td>0.57</td>
<td>0.67</td>
<td>0.68</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4</td>
<td>0.37</td>
<td>0.34</td>
<td>0.42</td>
<td>0.57</td>
<td>0.66</td>
<td>0.67</td>
<td>0.68</td>
<td>0.69</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4</td>
<td>0.35</td>
<td>0.33</td>
<td>0.43</td>
<td>0.58</td>
<td>0.65</td>
<td>0.67</td>
<td>0.67</td>
<td>0.68</td>
</tr>
<tr>
<td>0.42</td>
<td>0.41</td>
<td>0.36</td>
<td>0.33</td>
<td>0.42</td>
<td>0.57</td>
<td>0.65</td>
<td>0.66</td>
<td>0.67</td>
<td>0.68</td>
</tr>
<tr>
<td>0.44</td>
<td>0.42</td>
<td>0.37</td>
<td>0.33</td>
<td>0.42</td>
<td>0.56</td>
<td>0.63</td>
<td>0.66</td>
<td>0.66</td>
<td>0.69</td>
</tr>
</tbody>
</table>
```
Goal

Use this information to determine the position of the phases, interface location, curvature of the interface, other types of information which are of interest
Marching Cubes (MC)

Compute isosurface $G(x) = \nu$

from continuous data

Here we look at a two
dimensional case:
  - values of $G(x)$ are known at
    the corners of squares that
    cover the domain
  - Move through the system one
    box at a time
Marching Cubes (MC)

An interface exists along a cube edge if (intermediate value theorem):

\[ [G(x^+) - \nu][G(x^-) - \nu] \leq 0 \]

The interface locations are approximated along the cube edges using linear interpolation:

\[ x_{\nu} = x^+ + \frac{G(x^+) - \nu}{G(x^+) - G(x^-)}(x^- - x^+) \]
Marching Cubes (MC)

The reason that the MC algorithm is not called the marching squares algorithm is because it is a 3-D algorithm.

Three dimensions makes things more complicated:
- surfaces are represented as a set of triangles
- you have to figure out a way to specify how to form these triangles
Data Sets with MC
Data Sets with MC
MC and Interfacial Areas

Suppose we want to measure interfacial area from a porous medium data set. We are interested in the component interfacial areas between one phase and another. For a three phase (two fluids) system there are three such interfaces: $\Omega_{ws}, \Omega_{ns}, \Omega_{wn}$

The MC allows us to extract one interface at a time, which usually means you have to compute the interfacial areas you want using interfaces you can compute easily:

$$a_{wn} = \frac{1}{2} \left( a_w + a_n - a_s \right)$$
Porous Medium Marching Cubes (PMMC) Objectives

- Use the ideas developed in the MC to compute the component interfaces explicitly to obtain a precise representation of the system including the common curve.

- Implement in a general framework so that it can be applied to different kinds of data.
Porous Media Marching Cubes (PMMC)

- Solid surface defined by $S(x) = \nu_s$
- Interface between wetting and non-wetting fluids defined by $F(x) = \nu_{wn}$
- $S(x)$ is defined $\forall x \in \Omega$, $F(x)$ may be defined $\forall x \in \Omega_w \cup \Omega_n$
- Using linear interpolation, construct approximations for the interfaces and common curve within each cube as in the MC algorithm
Porous Media Marching Cubes (PMMC)

Sets needed for the PMMC algorithm.

<table>
<thead>
<tr>
<th>Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{N}<em>{c,l}, \mathcal{N}</em>{li}, \mathcal{E}_{c,l}, \mathcal{X}_l$</td>
<td>nodes bounding cube, nodes forming all edges $i$, edges in cube $l$, and locations of nodes bounding cube</td>
</tr>
<tr>
<td>$\mathcal{V}<em>{wn}, \mathcal{V}</em>{ns}, \mathcal{V}_{ws}, \mathcal{V}_s$</td>
<td>vertices on the respective surface</td>
</tr>
<tr>
<td>$\mathcal{T}<em>{wn}, \mathcal{T}</em>{ns}, \mathcal{T}_{ws}, \mathcal{T}_s$</td>
<td>triangles comprised of vertices on the respective surface</td>
</tr>
<tr>
<td>$\mathcal{V}_{wns}$</td>
<td>vertices where the three phases meet</td>
</tr>
<tr>
<td>$\mathcal{L}_{wns}$</td>
<td>line segments consisting of pairs of vertices from $\mathcal{V}_{wns}$</td>
</tr>
</tbody>
</table>
PMMMC Algorithm - Five Primary Cases

1. All cube corners correspond to single phase; no interface exists.
2. Some corners have \( F(x) \leq \nu_{wn} \), the rest have \( F(x) > \nu_{wn} \); only the wn interface exists.
3. Some corners have \( S(x) \leq \nu_{s} \), the rest have \( F(x) \leq \nu_{wn} \); only the ws interface exists.
4. Some corners have \( S(x) \leq \nu_{s} \), the rest have \( F(x) > \nu_{wn} \); only the ns interface exists.
5. Some corners have \( S(x) \leq \nu_{s} \), some have \( F(x) \leq \nu_{wn} \), the rest have \( F(x) > \nu_{wn} \); all interfaces and common curve are present.
Algorithm 1 PMMC Algorithm

for \( l = 1, \ldots, n^3 \) do

Form \( N_{c,l}, N_{a,l}, E_{c,l} \) and \( X_l \)

if \( x_i \in \Omega_a \forall x_i \in X_l, \alpha \in \{w, n, s\} \) then

no surfaces or common curve segments exist in cube \( l \)

else if \( x_i \in \{\Omega_w, \Omega_n\} \forall x_i \in X_l \) then

apply the MC algorithm using \( F(x) = \nu_{wn} \) to find \( V_{wn,l} \) and \( T_{wn,l} \)

else if \( x_i \in \{\Omega_w, \Omega_s\} \forall x_i \in X_l \) then

apply the MC algorithm using \( S(x) = \nu_s \) to find \( V_{ws,l} \) and \( T_{ws,l} \)

else if \( x_i \in \{\Omega_n, \Omega_s\} \forall x_i \in X_l \) then

apply the MC algorithm using \( S(x) = \nu_s \) to find \( V_{ns,l} \) and \( T_{ns,l} \)

else

apply the MC algorithm using \( S(x) = \nu_s \) to find \( V_s \) and \( T_s \)

find \( F(x) \forall x \in V_s \)

if \( F(x) \) is defined \( \forall x_i \in X_l \) then

use linear interpolation to form \( V_{wns,l} \) and \( L_{wns,l} \)

else

use extrapolation to form \( V_{wns,l} \) and \( L_{wns,l} \)

end if

use \( V_s \) and \( V_{wns,l} \) to form \( V_{ws} \) and \( V_{ns} \)

form \( T_{ws,l} \) and \( T_{ns,l} \)

form \( V_{wn,l} \) from \( E_{c,l} \)

form \( T_{wn,l} \) from \( V_{wns,l} \) and \( V_{wn,l} \)

end if

update global sets \( V_{wn}, V_{ws}, V_{ns}, V_{wns}, T_{wn}, T_{ws}, T_{ns} \) and \( L_{wns} \) as needed

end for
PMMC - Finding the Common Curve

- Values of $F(x)$ are obtained at every point on the solid surface.
- These values may be used to compute the common line along triangle edges in exactly the same fashion vertices are computed along cube edges in the MC algorithm.
Higher Order Approach (HOPMMC)

Construct a higher order representation of \( S(x) \) and \( F(x) \)
use this to compute the common curve more accurately.

\[
\tilde{F}(x) = a_0 + a_1x + a_2y + a_3z + a_4xy + a_5xz + a_6yz + a_7xyz
\]

Based on our knowledge of the system, we require

\[
\begin{align*}
\tilde{F}(x) - \nu_{wn} &= 0 \\
\dot{S}(x) - \nu_s &= 0 \\
n \cdot (x - x_0) &= 0
\end{align*}
\]

\[
n = [\nabla \tilde{F}(x_0) \times \nabla \dot{S}(x_0)]/|\nabla \tilde{F}(x_0) \times \nabla \dot{S}(x_0)|
\]
- Subdivide each segment of the common curve to generate initial points

- Use points as an initial guess for a Newton iteration that moves them to the proper position.

- Note that this could be applied to improve the PMMC surfaces also.
PMMC in Application

We have methods to construct the objects we are interested in as long as we have functions \( S(x) \) and \( F(x) \) and isovalue \( \nu_s \) and \( \nu_{wn} \)

- Gives us information about the position of the phases and interfaces as well as other information

- How do we obtain these functions and isovalue for a particular data set?
Test Case 1: Lattice Boltzmann (LB)

The solid phase is composed of a sphere packing so that the geometry is known exactly. $S(x)$ is computed using the signed distance function:

$$S(x) = \min \left\{ ||x - c_s||_2 - r_s \right\}, \text{ for } s = 1, \ldots, n_S$$

Note that $S(x) = 0$ by construction at the solid surface.

$F(x)$ is chosen to be the fluid density output from the LB simulation. Because it is only defined in the porespace, an extrapolation is necessary to determine its value at the solid surface.
Test Case 2: X-Ray Tomography

Objective for experimental data: Determine $S(x)$ and $F(x)$, as well as isovalues $\nu_s$ and $\nu_{wn}$
Test Case 2: X-Ray Tomography

“dry” data

three-phase data
Isovalue determination - solid phase

Histogram for 2-phase system

mesh value

number of values

$\nu_s$

solid values
Isovalue determination - fluid phase

Histogram for 3-phase system

- Air values
- Water values

Number of values vs. mesh value
Things to Consider: Isovalue Determination

- If you have any information that you know for sure (ie. porosity, saturation), you should be able to use this information to determine isovalues that will match them.

- Interfacial area estimates should be relatively insensitive to the choice of isovalue.
Other Ideas to Consider

“Edge detection” or “steepest descent”
Edge Detection Algorithms

Take the gradient of the values and look for the regions where they are high.

If you are looking for a maximum in the gradient and wish to find points that do not live on the grid, you need to use a non-linear approximation (not too bad in 2-D).
Data Set Overlap