Recognition and Reconstruction of Tumor Boundaries

Vincent Angelucci

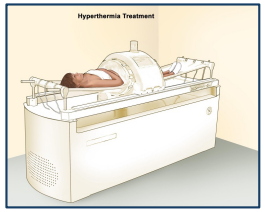
Department of Mathematics, West Chester University of Pennsylvania

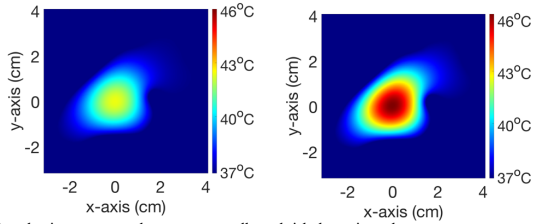
Abstract

Hyperthermia therapy is an emerging and relatively safe clinical treatment for human tumors. Mathematical models, such as Pennes' bioheat equation, are widely used to simulate heat dissipation during hyperthermia procedures, providing deeper insight into treatment efficacy in clinical trials. When solving Pennes' bioheat equation, a crucial step is accurately identifying the tumor boundary from CT scan images and reconstructing that boundary using an appropriate algebraic representation. In this project, we study a reliable computational framework that implements a sequence of numerical algorithms to detect and reconstruct tumor boundaries. Numerical experiments based on real CT scans of live tumors are also presented to demonstrate the effectiveness of the proposed method.

Introduction

- In Hyperthermia Therapy often a fluid containing paramagnetic nano particles is injected into a patient's tumor. Once exposed to an alternating magnetic field, the magnetic energy is converted into thermal energy. This heat is applied and it kills the tumor internally.





- This is beneficial because it prevents doctors from having to extract the tumor manually and risk damaging other areas.
- However, for this method it is crucial to have a precise model on the heat transfer of the biological tissue of the tumor to ensure accurate application of heat.
- The equation that models this specific type of heat transfer is known as Penne's Bioheat equation.

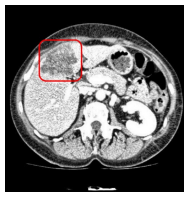
$$\rho^- c^- \frac{\partial T^-}{\partial t} = \nabla \cdot (k^- \nabla T^-) - \omega_b^- \rho_b c_b (T^- - T_b) + Q_m^- + Q_r,$$

T^-: tissue temperature, ρ^- and ρ_b: tissue density,
c^- and c_b : specific heat, k^-: thermal conductivity,
ω_b^-: blood perfusion rate, T_b: arterial blood temperature,
Q_m^-: metabolic heat, Q_r: external heat

- In order to use this partial differential equation on a particular tumor, we need an initial condition in the form of an algebraic equation that models the physical tumor boundary. (Note: Penne's equation can't read images - needs an algebraic domain to represent tumor)
- This brings us to the focus of this project which is to recognize and reconstruct this tumor boundary into an algebraic equation from a CT scan image to be able to use in solving Penne's Bioheat equation.

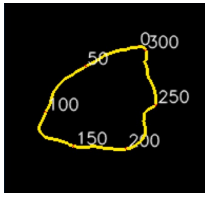
Recognizing Tumor

STEP 1) Original CT image scan is taken with enhanced contrast.

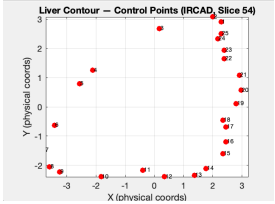


STEP 2) Tumor is isolated in picture using an AI binary masking technique.

STEP 3) Canny edge detection is used to mark all points around boundary and order them counter-clockwise (points are yellow).



STEP 4) Douglas-Peucker Algorithm is used to filter out only the most important points. We call these points the Control Points. (Not every equidistant point).



Reconstructing Tumor Boundary

- Now that we have condensed the tumor boundary into the control points, the next step is to interpolate these coordinates (Meaning fill the gaps between each point).
- Typically, interpolation methods take a set of points and fill the gaps between them using a single algebraic function y=f(x)
- However, in this case, we can't use a standard function since the shape causes some inputs to correspond to several outputs, so we must use 2 parametric equations $X(\varepsilon)$ and $Y(\varepsilon)$ to represent the boundary
- Furthermore, instead of just using one pair of equations, we can use a method that generates a pair of equations for every space between each point.
- The method we use is called parametric spline interpolation which generates a pair of piecewise defined equations to fit the points.

Parametric Spline Interpolation

For some set of control points: $\{CP_n = (x_n, y_n)\}_{n=0}^{N_{CP}}$,
$CP_0 = CP_{N_{CP}}$, (N_{CP} = number of control points)

The series of parametric equations that approximate the spaces between each point is given below:

$$X_n(\varepsilon) = a_n + b_n(\varepsilon - \varepsilon_n) + c_n(\varepsilon - \varepsilon_n)^2 + d_n(\varepsilon - \varepsilon_n)^3,$$
$$Y_n(\varepsilon) = p_n + q_n(\varepsilon - \varepsilon_n) + u_n(\varepsilon - \varepsilon_n)^2 + v_n(\varepsilon - \varepsilon_n)^3,$$

for $n = 0, \dots, N_{CP} - 1$,
provided the following conditions are satisified

Standard Conditions:

$$X_{n-1}(\varepsilon_n) = X_n(\varepsilon_n) = x_n, \quad Y_{n-1}(\varepsilon_n) = Y_n(\varepsilon_n) = y_n$$
(Continuity)
$$X'_{n-1}(\varepsilon_n) = X'_n(\varepsilon_n), \quad Y'_{n-1}(\varepsilon_n) = Y'_n(\varepsilon_n)$$
(Differentiability)
$$X''_{n-1}(\varepsilon_n) = X''_n(\varepsilon_n), \quad Y''_{n-1}(\varepsilon_n) = Y''_n(\varepsilon_n)$$
(Twice Differentiability)
$$X_0(\varepsilon_0) = X_{N_{CP}-1}(\varepsilon_{N_{CP}}), \quad Y_0(\varepsilon_0) = Y_{N_{CP}-1}(\varepsilon_{N_{CP}})$$
(Ensures first and last values in equations are equivalent to close loop)

Boundary Conditions:

$$X'_0(\varepsilon_0) = X'_{N_{CP}-1}(\varepsilon_{N_{CP}}), \quad Y'_0(\varepsilon_0) = Y'_{N_{CP}-1}(\varepsilon_{N_{CP}})$$
$$X''_0(\varepsilon_0) = X''_{N_{CP}-1}(\varepsilon_{N_{CP}}), \quad Y''_0(\varepsilon_0) = Y''_{N_{CP}-1}(\varepsilon_{N_{CP}})$$
(Ensures smoothness and prevents unnatural curvature between the first and last Control Points)

STEP 1) Start with the set of control points: $\{CP_n = (x_n, y_n)\}_{n=0}^{N_{CP}}$, $CP_0 = CP_{N_{CP}}$

STEP 2) Define each set of equations:

$$X_n(\varepsilon) = a_n + b_n(\varepsilon - \varepsilon_n) + c_n(\varepsilon - \varepsilon_n)^2 + d_n(\varepsilon - \varepsilon_n)^3,$$
$$Y_n(\varepsilon) = p_n + q_n(\varepsilon - \varepsilon_n) + u_n(\varepsilon - \varepsilon_n)^2 + v_n(\varepsilon - \varepsilon_n)^3,$$
for $n = 0, \dots, N_{CP} - 1$

(Need to solve for unknowns)

STEP 3) Solve for each ε using the formula below:

$$\varepsilon_n = \varepsilon_{n-1} + \frac{(D_n)^e}{\sum_{k=1}^{N_{CP}} (D_k)^e}, \text{for } n = 1, \dots, N_{CP} \text{ where e} = 0.5, \varepsilon_0 = 0$$

$$\text{and} \quad D_n = \left\| CP_n - CP_{n_-} \right\| = \sqrt{\left(x_n - x_{n_-}\right)^2 + \left(y_n - y_{n_-}\right)^2}$$
(The distance between each control point)

STEP 4) Can now solve for the constants a_n and p_n directly from the control points:

$$a_n = x_n, \quad p_n = y_n$$

STEP 5) Next solve for c_n and u_n using a linear system constructed from substituting conditions

The coefficients $\vec{c}$ are found by solving $A\vec{c} = \vec{g}$:

$$A = \begin{pmatrix} 2\left(\Delta\varepsilon_{N_{cp}-1} + \Delta\varepsilon_0\right) & \Delta\varepsilon_0 & \dots & \Delta\varepsilon_{N_{cp}-1} \\ \Delta\varepsilon_0 & 2(\Delta\varepsilon_0 + \Delta\varepsilon_1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Delta\varepsilon_{N_{cp}-2} \\ \Delta\varepsilon_{N_{cp}-1} & \dots & \Delta\varepsilon_{N_{cp}-2} & 2\left(\Delta\varepsilon_{N_{cp}-2} + \Delta\varepsilon_{N_{cp}-1}\right) \end{pmatrix},$$

$$\vec{c} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N_{cp}-1} \end{pmatrix}, \quad \vec{g} = \begin{pmatrix} \frac{3}{\Delta\varepsilon_0}(a_1 - a_0) - \frac{3}{\Delta\varepsilon_{N_{cp}-1}}\left(a_0 - a_{N_{cp}-1}\right) \\ \frac{3}{\Delta\varepsilon_1}(a_2 - a_1) - \frac{3}{\Delta\varepsilon_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{\Delta\varepsilon_{N_{cp}-1}}\left(a_0 - a_{N_{cp}-1}\right) - \frac{3}{\Delta\varepsilon_{N_{cp}-2}}\left(a_{N_{cp}-1} - a_{N_{cp}-2}\right) \end{pmatrix}$$

Where $\Delta\varepsilon_n = \varepsilon_{n+1} - \varepsilon_n$

And for u_n, we use a similar system (switching out the 'c' constants for 'u' constants and 'a' constants for 'p' constants to orientate for Y equations)

After each system is solved, we will have coefficients:
a_n, p_n, c_n, u_n, and only need to solve for b_n, d_n, q_n, v_n

STEP 6) Use the formulas Below (derived by substituting Conditions) to solve for each b and d:

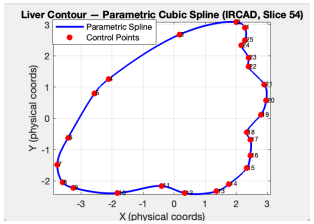
$$b_n = \frac{1}{\Delta\varepsilon_n}(a_{n+1} - a_n) - \frac{\Delta\varepsilon_n}{3}(2c_n + c_{n+1}), \quad \text{for } n = 0, \dots, N_{cp} - 2$$
$$b_{N_{cp}-1} = \frac{1}{\Delta\varepsilon_{N_{cp}-1}}\left(a_0 - a_{N_{cp}-1}\right) - \frac{\Delta\varepsilon_{N_{cp}-1}}{3}\left(c_0 + 2c_{N_{cp}-1}\right)$$
$$d_n = \frac{1}{3\Delta\varepsilon_n}(c_{n+1} - c_n), \quad \text{for } n = 0, \dots, N_{cp} - 2$$
$$d_{N_{cp}-1} = \frac{1}{3\Delta\varepsilon_{N_{cp}-1}}\left(c_0 - c_{N_{cp}-1}\right)$$

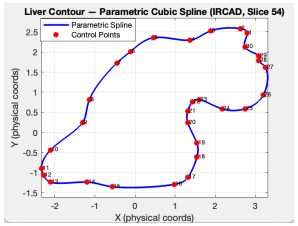
STEP 7) And for q and v, we use similar equations (switching out the 'b' constants for 'q' constants, the 'd' constants for 'v' constants, the 'a' constants for 'p' constants and the 'c' constants for 'u' constants to orientate for Y equations)

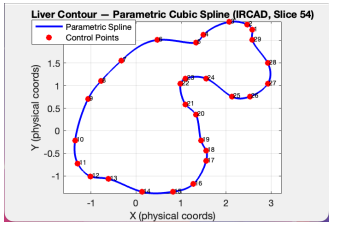
STEP 8) Lastly, plug all values into each pair equations and use them as the domain in the bioheat equation.

Numerical Experiments

3 Simulations were run in Matlab using real data from Tumor Scans.







Conclusion and Summary

In this project, we used a method to recognize and reconstruct tumor boundaries from CT scan data using a combination of image processing and parametric spline interpolation. By converting the image data into control points, and interpolating them, we were able to accurately capture the relative shape of the tumor into algebraic equations. Using these equations, we can solve Penne's Bioheat equation to model heat transfer of the tumor. This method can be applied to multiple slices of the same tumor to construct an accurate heat dissipation model, providing valuable insight for doctors to reference during hyperthermia therapy. In the future, we could extend this method to utilize another bioheat equation to model the surrounding tissue. By modeling both the tumor and its surroundings, we can better approximate tissue damage risks and provide more reliable guidance for treatment planning. Overall, this project demonstrates a useful technique for improving hyperthermia treatment through mathematical modeling and lays the foundation for development of more accurate models.

References

Xu, J., Guan, Z., & Li, C. (2025). A highly accurate augmented matched interface and boundary method for heat dissipation in tumor anatomies. Journal of Thermal Biology, 133, 104299. https://doi.org/10.1016/j.jtherbio.2025.104299

National Cancer Institute. (2025, May 15). *Hyperthermia to treat cancer* [Image]. Hyperthermia page